

LUDIFICACIÓN DE DOCKING MOLECULAR PARA ACELERAR EL DISEÑO DE FÁRMACOS

por

Camilo Ignacio Vega Hidalgo

Patrocinante: Gonzalo Rojas Durán

Memoria presentada
para la obtención del título de

INGENIERO CIVIL INFORMÁTICO

Departamento de Ingeniería Informática y Ciencias de la Computación

de la

UNIVERSIDAD DE CONCEPCIÓN



Concepción, Chile

Agosto, 2018

“Estas páginas están dedicadas a todos aquellos que me han acompañado en este largo proceso de formación, en especial a mi familia, quienes fueron un pilar de apoyo en los momentos mas difíciles de este camino recorrido.”



Resumen

Hoy en día se hace uso de herramientas Bioinformáticas para el diseño de fármacos mediante simulación. Dentro de los métodos usados se encuentra el de Docking Molecular, en el cual se predice la posición preferida de una molécula para unirse a otra, con el objetivo de que estas formen un complejo estable. Por lo general simular esto es muy costoso en cuanto a tiempo y no siempre se puede llegar a una solución óptima, debido a que hay que explorar un rango muy amplio de posiciones. El trabajo de esta memoria fue adaptar el método de Docking Molecular a un videojuego, usando representaciones bidimensionales de las moléculas del complejo y calculando la estabilidad entre estas, para así, mediante la interacción de una gran cantidad de jugadores se pueda obtener una mejor aproximación inicial que mejore los tiempos de la simulación.



Índice general

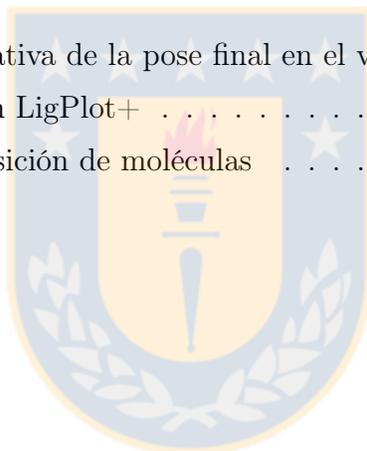
Resumen	III
Índice de figuras	VI
Capítulo 1. Introducción	1
1.1. Objetivos	3
1.1.1. Objetivo General	3
1.1.2. Objetivos Específicos	3
Capítulo 2. Marco Teórico	5
2.1. Enfoques del Docking Molecular	5
2.1.1. Complementariedad de forma	6
2.1.2. Simulación	6
2.2. Ludificación en ciencias	7
2.2.1. Foldit	7
2.2.2. Phylo	8
2.2.3. UDock	8
2.3. Representación 2D	9
2.3.1. PoseView	10
2.3.2. LigPlot+	10
2.3.3. Conclusiones preliminares	11
Capítulo 3. Sistema Desarrollado	12
3.1. Desafío a abordar	12
3.1.1. Concepto de Docking Molecular	12
3.1.2. Proceso	13
3.2. Solución propuesta	14
3.3. Alcance de la memoria	16
3.4. Descripción del Sistema	17

3.4.1. Plataforma de desarrollo	17
3.4.2. Formato de entrada	17
3.4.3. Sistema	17
3.4.4. Arquitectura	18
3.4.5. El juego	22
3.4.6. Metodología utilizada	24
Capítulo 4. Resultados	25
4.1. Pose final en el juego y en LigPlot+	25
4.2. Rendimiento	26
4.3. Pruebas con usuarios	27
4.4. Aspectos a mejorar	27
4.5. Comparativa con otros enfoques	29
Capítulo 5. Conclusiones	30
Bibliografía	32



Índice de figuras

2.1.	Diagrama de acoplamiento entre una proteína y un ligando. . .	5
2.2.	Sistema de puntos de anclaje de UDock.	9
3.1.	Comparativa entre orientación y la energía de interacción . . .	13
3.2.	Diagrama de proceso de Docking Molecular	14
3.3.	Diagrama de proceso del proyecto	15
3.4.	Arquitectura del juego	18
3.5.	Captura de pantalla del videojuego	22
3.6.	Pantalla de resultados al final termino del juego	23
4.1.	Comparativa de la pose final en el videojuego contra represen- tación en LigPlot+	25
4.2.	Superposición de moléculas	28



Capítulo 1

Introducción

Históricamente para contrarrestar las enfermedades se han hecho uso de sustancias. Mediante el estudio y la refinación de estas se dio paso a lo que conocemos como fármacos. En un principio estos se obtenían mediante la experimentación. Actualmente, una de las formas para el diseño y creación de fármacos es a través de la Bioinformática, la cual, gestiona y analiza datos biológicos mediante la aplicación de estadísticas y tecnologías computacionales [11]. Uno de los métodos usados por la Bioinformática es del de Docking Molecular. Este método consiste en encontrar la orientación de acoplación entre una molécula pequeña, llamada **ligando** y que será candidata a fármaco, con una **proteína** sobre la cual se ejercerá la acción [18].

La mejor orientación¹ entre ambas moléculas es aquella que logre minimizar la energía generada por la interacción entre estas, lo que llevará a cabo la generación de un complejo estable.

Uno de los enfoques usados en el método del Docking Molecular es la simulación. Este enfoque va calculando la energía de interacción entre la proteína y ligando mediante ecuaciones que dependen de la distancia física que hay entre estas moléculas. Si bien los resultados de este enfoque son cercanos a la realidad, para obtenerlos el costo en tiempo de computación es demasiado alto. Esto se debe a que hay que explorar un amplio rango de configuraciones para el complejo proteína-ligando. Otro inconveniente es que no siempre se puede llegar a la mejor configuración cayendo en óptimos locales [18], esto se debe a que la solución encontrada depende de lo buena que sea la aproximación inicial. Una forma de aminorar estos problemas es mediante la búsqueda de aproximaciones iniciales que tengan mayor probabilidad de encontrar una mejor solución final.

El rango de búsqueda para encontrar estas aproximaciones iniciales es bastante

¹Corresponde a la posición y orientación de un ligando con respecto a una proteína.

amplio. Se puede usar la capacidad de cómputo de múltiples dispositivos para ayudar a esta búsqueda, pero es requerido encontrar una forma de poder acceder a la capacidad de cómputo de estos dispositivos. A partir de este escenario surge la idea del Doctor en Ciencias Biológicas, Alexis Salas Burgos, que propone ludificar el método del Docking Molecular. Ludificar es hacer uso de mecánicas de juego en entornos no lúdicos [19]. De acuerdo a esto, como trabajo de esta memoria se realizó un videojuego que es capaz de adaptar complejos proteína-ligando a escenarios de juego. La interacción permitida del usuario con el juego está basada en movimientos de **traslación** y **rotación**. En base a estos movimientos se obtendrá una calificación mediante una función de puntaje, la cual tiene relación con la energía libre del complejo. A partir de los mejores puntajes y la configuración utilizada para obtenerlo, se busca obtener una mejor aproximación inicial para los métodos de simulación, con tal de evitar los mínimos locales. La ventaja de adaptar el problema a un videojuego es que un número potencialmente alto de jugadores, que no sepan necesariamente de farmacología, puede aportar a la búsqueda de estas aproximaciones aprovechando la capacidad de cómputo de cada dispositivo.

Como prueba de concepto, en esta propuesta se implementó un juego bidimensional. Dado que los modelos de los complejos proteína-ligando son tridimensionales, tienen que ser procesados previamente por un programa que realiza la conversión del modelo de 3D a 2D. Una vez obtenido esta representación, se procede a extraer los datos necesarios, los que son almacenados en un archivo que pueda ser leído por el videojuego, para generar un escenario de juego. Así mismo, el archivo de salida generado por la solución obtenida en el videojuego es uno que puede ser leído por el programa de visualización de diagramas 2D. Complementándolo con la descripción original del complejo, se espera que este archivo pueda ser convertido para que la solución obtenida en el videojuego pueda ser analizado por un software de simulación de Docking Molecular.

1.1. Objetivos

1.1.1. Objetivo General

Crear la base de un videojuego de Docking Molecular el cual a partir de datos reales, pueda generar aproximaciones iniciales que mejoren los resultados obtenidos por métodos de simulación.

1.1.2. Objetivos Específicos

- Obtener una representación bidimensional de modelos tridimensional de complejos proteína-ligando.
- Extraer datos de representaciones bidimensionales de complejos proteína-ligando con tal de generar escenarios de juego para el videojuego.
- Crear una base escalable para el videojuego, tal que sea fácil agregar mayor cantidad de escenarios de juego a partir de diversos complejos proteína ligando.
- Implementar funciones de puntaje propias del Docking Molecular en un entorno bidimensional.
- Almacenar en un archivo las posiciones de las moléculas que entreguen los puntajes mas altos del jugador.

Previo a desarrollar el software, se realizó un estudio del método de Docking Molecular, para poder entender los conceptos que lo envuelven y en base a ello evaluar cómo se llevaría a cabo el desarrollo. Posteriormente se continuó con el desarrollo del software, partiendo con un prototipo que serviría de base del videojuego. Se realizaron demostraciones al cliente cada vez que se lograban avances significativos o cuando se requería realizar cambios con respecto a la idea original.

Este informe, por medio de sus capítulos, busca justificar las decisiones de diseño, mostrar el desarrollo y los resultados obtenidos de este trabajo. En el capítulo 2, Marco Teórico, muestra la forma actual de abordar el problema existente. El capítulo 3, Sistema Desarrollado, va desde el problema que se quiere a abordar hasta llegar a

explicar el uso del sistema. En el capítulo 4, Resultados, se realizaron pruebas de rendimiento y con usuarios, de las cuales se pudo determinar las fortalezas y debilidades del proyecto. En el capítulo 5, Conclusiones, se realizó un análisis resumido del trabajo realizado, describiendo las dificultades encontradas como también estableciendo aquello que todavía queda por realizar en un futuro.



Capítulo 2

Marco Teórico

Es necesario realizar una investigación previa con tal de poder tener una mejor visión del trabajo a realizar. Dentro de los tópicos analizados, se encuentran los enfoques usados en el Docking molecular y las herramientas bioinformáticas usadas en el enfoque de simulación. Se procedió con la ludificación en ciencias, incluyendo algunos ejemplos que muestran cómo es aplicada a distintas áreas de la bioinformática. Finalmente, se procede a comparar entre alternativas existentes para la representación bidimensional de complejos proteína ligando en 2 dimensiones, y así poder decidir cuál será usada para desarrollar los escenarios de juego.

2.1. Enfoques del Docking Molecular

Existen diversos métodos para el diseño de fármacos, uno de ellos corresponde al Docking Molecular. El objetivo de este método es buscar la posición y orientación preferida de 2 moléculas con tal de que estas formen un complejo estable. Estas moléculas por lo general corresponden a una **proteína** y a un **ligando**, donde el ligando es bastante mas pequeño que la proteína (ver Figura 2.1).

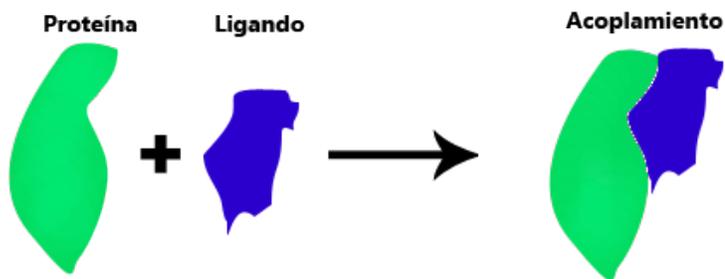


Figura 2.1: Diagrama de acoplamiento entre una proteína y un ligando.

Un complejo proteína-ligando se considera estable cuando la energía total del complejo es baja. Para esto, se evalúan las interacciones favorables entre los elementos

que componen ambas moléculas, y a estas interacciones por medio de una función se asigna un puntaje.

El Docking Molecular puede ser aplicado por 2 enfoques distintos. El primero de ellos describe la proteína y el ligando como superficies complementarias. El segundo intenta recrear las energías de interacción entre ambas moléculas en su proceso de acoplamiento. Ambos enfoques tienen ventajas y limitantes.

2.1.1. Complementariedad de forma

Este enfoque describe a la proteína y ligando como conjuntos de rasgos que los hacen acoplables. Dentro de estos rasgos, generalmente se incluyen la superficie molecular o descriptores de complementariedad de la superficie [20].

Si bien este enfoque por lo general entrega soluciones robustas y rápidas, es incapaz de modelar los movimientos dinámicos y cambios conformacionales del complejo de forma precisa. Los métodos basados en complementariedad de forma pueden escanear muchos ligandos en pocos segundos y lograr deducir si se pueden unir al sitio activo de la proteína.

2.1.2. Simulación

En este enfoque, la proteína y ligando están separados por una distancia física, en la cual el ligando encuentra su posición de acoplamiento dentro del sitio activo de la proteína después de realizar cierta cantidad de movimientos en su espacio conformacional. Estos movimientos corresponden a traslados y rotaciones e inducen un costo energético al sistema, por lo cual después de cada movimiento se calcula la energía total del sistema [18].

Las ventajas de este método es que se puede incorporar de manera más fácil la flexibilidad del ligando en el modelado y además es mucho más cercano físicamente a lo que ocurre en la realidad. Lamentablemente, el costo computacional con respecto al tiempo de ejecución de este enfoque es demasiado alto, ya que el espectro de movimientos que hay que analizar es demasiado amplio. Para aminorar este inconveniente se suelen usar técnicas basadas en rejillas o métodos de optimización rápida.

Actualmente existen diversos programas que simulan Docking Molecular. Dos de

los más usados son DOCK6 [5] y AutoDock [1]. Ambos usan distintos algoritmos para encontrar la orientación de acoplación del complejo proteína-ligando, y a través de sus distintas versiones han ido refinando estos algoritmos, para entregar soluciones más precisas y de manera mas veloz.

2.2. Ludificación en ciencias

La ludificación consiste en usar mecánicas y pensamientos de juegos a contextos no lúdicos. Dentro de los usos que se aplica la ludificación se encuentra el aprendizaje de ciencias, motivar a personas para obtener objetivos específicos y también se puede usar para ayudar a resolver complejos problemas de investigación [19]. Para lo último, se tiene que lograr que la interacción del jugador con el juego otorgue resultados que puedan ser usados como base para la solución del problema en el que está basado el juego. Aprovechando las capacidades de conectividad de hoy en día, estos juegos se pueden distribuir por internet y también ser desarrollados para diversas plataformas. Esto provoca que se alcance una mayor base de jugadores, de los cuales se pueden extraer gran cantidad de datos para ser analizados y a través de estos, obtener resultados que ayuden a la investigación.

Una de las grandes ventajas de la ludificación es permitir que personas puedan aportar en investigaciones sin tener conocimientos de aquello que se está investigando, sino simplemente por el hecho de jugar [12].

Se analizaron 3 videojuegos que sirven de apoyo a la investigación bioinformática, Foldit [8], Phylo [6] y UDock [9]. Este último está basado en el diseño de fármacos mediante Docking Molecular.

2.2.1. Foldit

Dentro del estudio de predicción de proteínas uno de los programas usados es Rosetta@home. Este programa otorga una interacción con el usuario mediante la línea de comandos. Si bien la precisión de este programa es alta, se requiere de gran cantidad de conocimientos técnicos para poder realizar un aporte a estos estudios. En base a esto nació Foldit, el cual es un videojuego basado en Rosetta@home que provee una interfaz mucho mas amigable e interactiva con la cual el jugador, mediante

su intuición y suerte, puede encontrar formas naturales de las proteínas que forman parte de los seres vivos.

Usando este videojuego como base, mas tarde se creó Foldit Standalone, el cual remueve los elementos competitivos de videojuego que incluye Foldit y agrega más características avanzadas, convirtiéndolo en una interfaz de manipulación de proteínas interactiva fácil de usar [13]. Ambos, Foldit y Foldit Stanalone no requieren conocimientos previos de Rosseta para empezar a usarlos.

2.2.2. Phylo

Phylo es un juego de puzzle desarrollado en Unity. Se desarrollaron versiones para dispositivos móviles con Android o iOS y además de una en WebGL. El juego traduce problemas de alineamiento de secuencia(múltiple sequence alignment MSA) a un escenario de puzzle, cuyas soluciones propuestas por jugadores en línea son usadas para mejorar la precisión de los MSA obtenidos con algoritmos de vanguardia. Al representar fragmentos de las secuencias de ADN mediante cuadros de colores permite que el juego pueda ser jugado por cualquier persona que esté ajena al contexto biológico del problema. Por tal motivo permite cubrir un rango de edad de usuarios amplio.

En un principio los datos que generaban los escenarios de juego de Phylo sólo eran formulados y seleccionados por un pequeño grupo de desarrolladores e investigadores. Tiempo después se lanzó Open-Phylo [14], la cual es una plataforma web de acceso libre que permite a los científicos ingresar sus propias secuencias al sistema y gestionar los esfuerzos de la gente que está tratando de alinearlas.

2.2.3. UDock

Es un videojuego de Docking Molecular que representa un complejo proteína-ligando de forma tridimensional. Este permite rotar en cualquier eje ambas moléculas mediante gestos de deslizamiento con el mouse, lo cual permite manipularlos de forma intuitiva. Para lograr esto utiliza Bullet, un motor de físicas de videojuego de código abierto [3]. Al ajustar una pose en específico el usuario puede definir puntos de anclaje entre estas dos moléculas, para luego ser atraídas entre sí por estos puntos (ver Figura

2.2). Al momento de atraerse se empieza a calcular la fuerza de interacción entre estas por una función de puntaje que está basada en la ley de Coulomb y una ecuación simplificada de van der Waals. El usuario en cualquier momento puede mejorar la solución llamando al método de optimización de Monte Carlo [17] por 5 segundos.

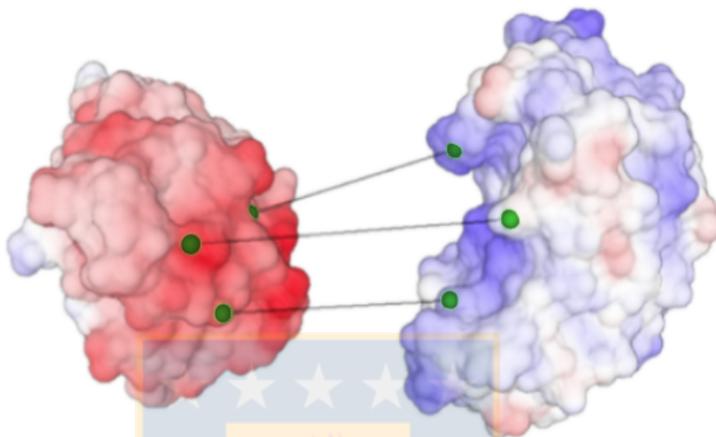


Figura 2.2: Sistema de puntos de anclaje de UDock.

El color de la superficie de los modelos moleculares de UDock se genera a partir de un suavizado del potencial eléctrico de la superficie, los cuales son datos otorgados por un archivo en formato mol2 [16].

El juego implementó un sistema de ranking en línea, donde uno puede comparar su puntaje con el del resto de los jugadores. Los mejores puntajes y sus configuraciones son analizados para obtener mejores soluciones.

La principal desventaja de este juego es que sólo sirve para complejos proteína-ligando donde éstas moléculas tengan similares tamaño, cuando la proteína es bastante mas grande que el ligando, el proceso de Docking por lo general se realiza en una cavidad de la proteína, por lo cuál mediante este juego sería difícil manipular y visualizar ambas moléculas.

2.3. Representación 2D

Para el desarrollo del videojuego se decidió usar una representación bidimensional del complejo proteína-ligando, esto se debe a que se quiere facilitar la visualización e interacción que tiene el usuario con el complejo. Además esta visualización permite

manipular ciertos complejos imposibles de manipular en UDock (descrito en la sección 2.2.3) tales como aquellos donde la proteína es de mucho mayor tamaño que el ligando.

Los archivos que contienen a los complejos, los representan en su forma tridimensional. Para poder obtener representaciones bidimensionales de estos, por lo general se usan algoritmos de aplanamiento. Ya que implementar uno de estos algoritmos requiere de bastante tiempo, se optó por utilizar un programa que generara estas representaciones a partir de un archivo de entrada. Los programas analizados fueron los siguientes:

2.3.1. PoseView

PoseView es un programa que genera de forma automática diagramas 2D de alta calidad de complejos ligando-proteína provistos como una entrada 3D [2]. Esa entrada debe venir directamente de estructuras de cristal o ser computarizadas por ejemplo por un programa de docking. La entrada puede ser un archivo PDB o puede estar en FlexX-specific RDF-notation. Los ligandos se deben proveer como archivos mol2.

La salida puede ser visualizada en la GUI del programa o exportada a archivos PNG, PDF o XFIG. Esta última permite modificar el diagrama con programas de gráficos compatibles con archivos XFIG.

Se pueden seleccionar varias opciones de visualización, como mostrar/ocultar interacción de energías, contactos hidrofóbicos como también múltiples opciones de etiquetado.

Puede ser usado en la línea de comandos para procesamiento de alto rendimiento de múltiples datos o por medio de una interfaz gráfica intuitiva para el usuario.

2.3.2. LigPlot+

Al igual que Poseview, LigPlot+ genera de forma automática diagramas 2D de complejos ligando-proteína que son provistos mediante un archivo PDB [4]. Lo que busca el programa es a partir de las coordenadas del archivo, calcular los enlaces de hidrógeno entre el ligando y la proteína para después proceder a aplanar el ligando y los residuos de la proteína que interactúa con éste. El diagrama generado busca ser lo más limpio posible, con un mínimo choque de átomos y líneas de unión sobrepuestas.

Los enlaces de hidrógeno son calculados a través de un programa llamado HBPLUS2.

Una vez creado el diagrama, estos pueden ser editados mediante una interfaz interactiva del programa. Las interacciones permiten mover los componentes del diagrama, rotarlos sobre átomos seleccionados, voltearlos sobre enlaces seleccionados, cambiar sus colores y editar sus etiquetas de texto.

El programa permite imprimir los diagramas, exportar a archivos PostScript o a un archivo DRW. Este último archivo guarda toda la información del diagrama, tales como coordenadas, componentes, nombre de las moléculas y enlaces. Además, estos archivos pueden ser leídos de vuelta en LigPlot para que el programa no tenga que realizar nuevamente la conversión de un archivo DRW [15].

2.3.3. Conclusiones preliminares

Al comparar ambos enfoques de Docking Molecular, el enfoque de simulación es aquél que nos da resultados mas cercanos a la realidad a costa de rendimiento. Pese a que los programas de simulación de Docking han podido disminuir los tiempos de cómputo, estos siguen siendo altos, debido al amplio espectro de movimientos de las moléculas que analizan.

Mediante la ludificación varios videojuegos apoyan problemas de investigación complejos, gracias a que con ellos, una cantidad masiva de jugadores pueden ayudar a buscar mejores soluciones. Mediante interfaces intuitivas permiten que personas puedan aportar sin tener conocimientos del tema. La implementación de estas interfaces a veces limita la variedad de problemas que pueden ser analizados a través de estos videojuegos. Tal es el caso de UDock, que sólo permite la manipulación de complejos moleculares en los cuales las moléculas tienen tamaños similares.

Para poder analizar una mayor variedad de complejos, se optó por realizar un videojuego que use representaciones bidimensionales de los complejos. Entre los dos programas de conversión a representación bidimensional analizados, LigPlot+ es aquel que entrega una representación mas fiel al modelo original, ya que para esto usa las coordenadas de las moléculas a aplanar. Además, el archivo de salida contiene las coordenadas de los átomos que contiene el diagrama bidimensional del complejo molecular, haciendo mas fácil la adaptación de este diagrama a un escenario de juego.

Capítulo 3

Sistema Desarrollado

Este capítulo describe las decisiones de desarrollo adoptadas para construir el sistema descrito. Comienza con un análisis del problema a tratar, y lo que propone esta memoria para resolverlo. Luego, se presenta la descripción del desarrollo del sistema creado, detallando la arquitectura usada y el funcionamiento de este.

3.1. Desafío a abordar

3.1.1. Concepto de Docking Molecular

El método del Docking Molecular, como ya se ha dicho previamente, predice la orientación preferida de una molécula al acoplarse a otra, con el fin de que estas moléculas formen un complejo estable. La molécula que se acopla por lo general es mas pequeña y es llamada **ligando**, mientras que la mayor es una **proteína**. Para determinar la estabilidad del complejo se utilizan **funciones de puntuación**, las cuales dependen de la posición y rotación del ligando con respecto a la proteína. Tales funciones son una composición de ecuaciones o leyes físicas aplicadas entre los átomos de ambas moléculas y en conjunto calculan la energía de interacción del complejo. Entre estas ecuaciones se puede encontrar las fuerzas de Van der Waals, las fuerzas por enlace de Hidrógeno o las fuerzas electromagnéticas descritas por la Ley de Coulomb. El complejo se considera estable cuando la energía de interacción es baja, mientras menor sea, más estable es el complejo. Tal como muestra la figura 3.1, mientras el ligando esté más acoplado a la proteína, la energía de interacción es menor.

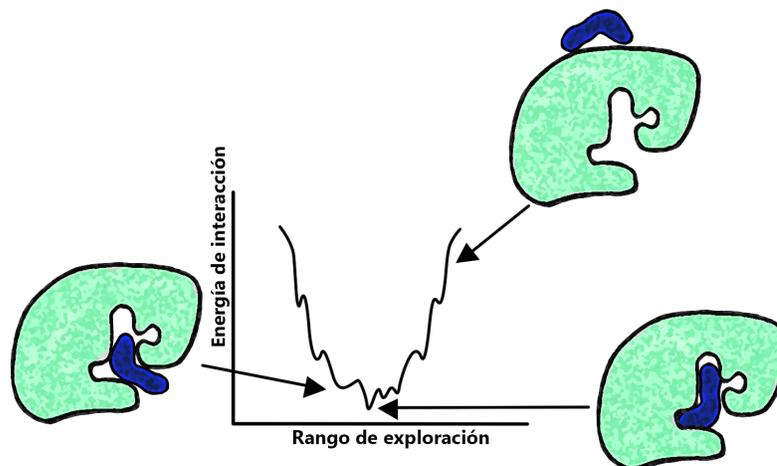


Figura 3.1: Comparativa entre orientación y la energía de interacción

En la Figura 3.1 se puede observar que para encontrar una energía de interacción baja, se debe explorar entre un amplio rango de orientaciones. Esta exploración se realiza computacionalmente cuando se usa el enfoque de simulación (explicado en 2.1.2). La velocidad con la que los programas de simulación entregan la solución y el resultado de esta, depende de la orientación inicial del ligando, la que es entregada por el archivo de entrada. El mayor problema es que en un número considerable de veces, encontrar la mejor solución no es posible, ya que mucha de estas veces se puede caer en un óptimo local, es decir, una solución que es la mejor dentro de las soluciones cercanas, pero que no lo es dentro del rango de exploración.

3.1.2. Proceso

Tal como muestra la Figura 3.2, el proceso comienza con la obtención del complejo proteína-ligando a analizar. Este complejo se obtienen desde el Protein Data Bank¹ [7]. El complejo puede ser obtenido en un solo archivo o en varios.

Este complejo es llevado a algún programa de Docking Molecular, donde se realizan las configuraciones de parámetros previo a la ejecución del algoritmo de Docking. Una vez realizada la configuración de parámetros, se procede a ejecutar el algoritmo. En cada iteración de este algoritmo se evalúa la orientación del ligando mediante una

¹Sitio en el cual la comunidad científica aporta subiendo archivos con estructuras moleculares obtenidas mediante investigación

función de puntaje. El algoritmo itera hasta encontrar una solución aparentemente buena, la cual puede corresponder a un óptimo local en vez del global. Una vez encontrada la solución, el programa genera un informe con los resultados de la simulación, entre esos, un archivo en formato PDB que contiene la orientación del ligando con respecto a la proteína.

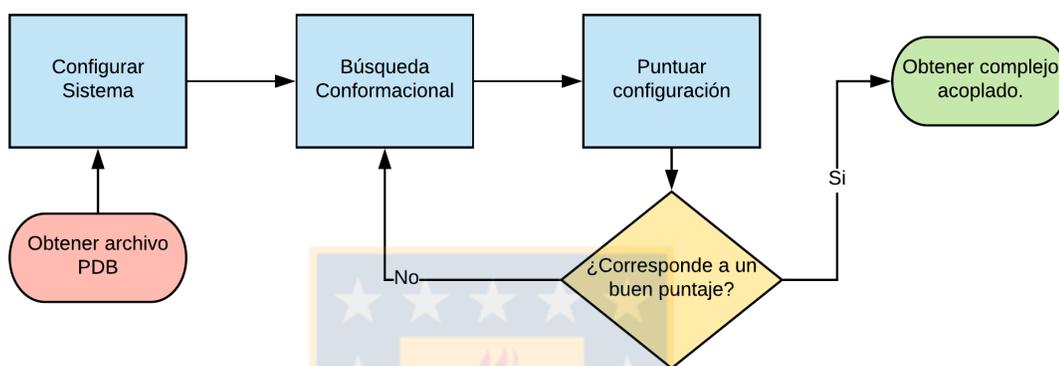


Figura 3.2: Diagrama de proceso de Docking Molecular

3.2. Solución propuesta

Ya que en el proceso de Docking Molecular mediante simulación, la precisión del resultado de este y la velocidad con la que se ejecute dependen de la orientación del ligando que contenga el archivo de entrada, se propuso crear un software que pueda calcular una mejor aproximación inicial de la solución, para que a partir de esta, se pueda mejorar el rendimiento del Docking Molecular para los complejos. Este software será un videojuego que ludificará el método del Docking Molecular.

¿Por qué un videojuego? Con la masificación de los computadores, dispositivos móviles y el acceso a internet, nació un estilo de videojuegos en línea llamados **Online Citizen Games** [12]. Estos juegos tienen la finalidad de que a través de la interacción de un grupo masivo de personas con el juego, se puedan obtener datos que ayuden a resolver problemas del área de la ciencia. La ventaja de estos videojuegos, es que en muchos casos pueden ser jugados por personas que no tengan conocimientos en el

área del problema a resolver. Incluso, estudios realizados de algunos Online Citizen Games muestran que, ante dificultades altas, las personas que juegan videojuegos frecuentemente obtiene mejores puntuaciones que las personas con conocimientos del trasfondo del problema [21].

Analizando las capacidades de otro videojuego de Docking Molecular llamado **UDock** (sección 2.2.3), el videojuego propuesto permite manipular una representación bidimensional del complejo proteína-ligando. Esto permitirá poder trabajar con complejos donde la proteína se encuentre dentro del sitio activo de la proteína, lo cual es complejo de lograr en UDock.

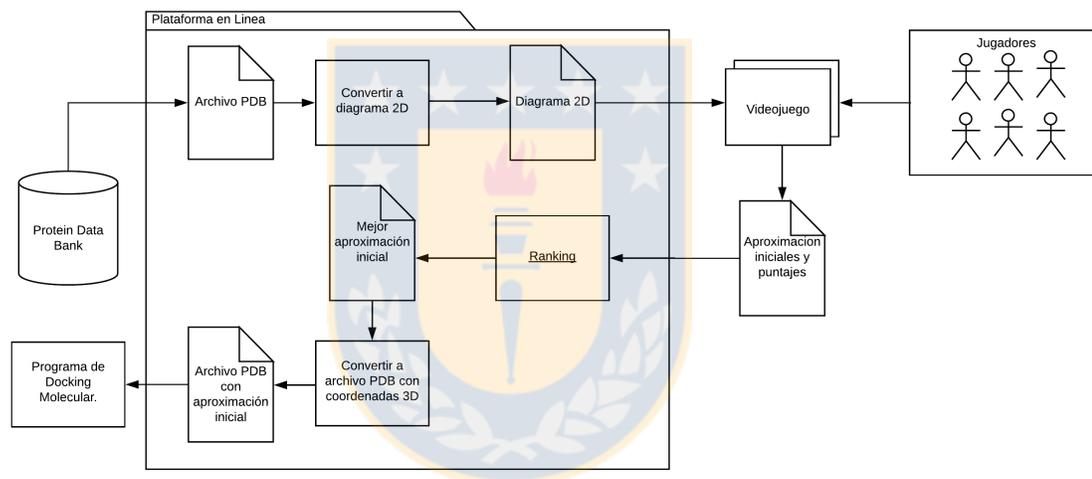


Figura 3.3: Diagrama de proceso del proyecto

Como muestra la Figura 3.3, la completitud de este proyecto requiere que se desarrollen 2 sistemas: una plataforma en línea y un videojuego que se comuniquen con esta plataforma. El objetivo de la plataforma en línea es que, a partir de datos reales, genere y provea de niveles al videojuego, además de procesar y almacenar las soluciones entregadas por todos los usuarios. Se creará un ranking con los mejores puntajes, ya que en teoría, estos entregarán las mejores aproximaciones finales. Otro objetivo del ranking es generar competitividad entre los jugadores, los cuales reintentarán un nivel con tal de superar puntajes que sean mejores que los de ellos. Las aproximaciones iniciales generadas por este sistema de ranking deberán ser convertidas a

representaciones 3D que se almacenen en archivos PDB para que puedan ser usadas por algún programa de simulación. Por otro lado, el videojuego tendrá que proveer de una adaptación del método de Docking Molecular, en la cual, por medio de la interacción del jugador y una función de puntaje, se obtenga una aproximación inicial que será entregada a la plataforma en línea.

3.3. Alcance de la memoria

El trabajo de esta memoria se enfocó en la creación del videojuego, dejando de lado la comunicación en línea, haciendo que este funcione de forma local. Se utilizó un software de terceros para la generación de niveles a partir de un archivo PDB. El videojuego genera las aproximaciones iniciales a partir de las configuraciones finales que obtengan un mayor puntaje, pero quedan almacenadas de forma local. Por último, la función de puntaje implementada (función 3.1) está sólo basada en la Ley de Coulomb, ya que se estimó que para esta primera fase del proyecto, era suficiente implementar una función que dependiese de la distancia entre los átomos de las moléculas y la carga entre ellas.

$$F_{ij} = k \frac{q_i q_j}{r_{ij}^2} \hat{r}$$

$$F_{total} = \sum_{i=1}^n \sum_{j=i+1}^n F_{ij} \quad (3.1)$$

La Función 3.1 corresponde a la suma de todas las fuerzas eléctricas generadas por las cargas de cada par de átomos i, j . Esta fuerza es inversamente proporcional a la distancia que los separa (representada por r). Esto quiere decir que mientras mas alejados estén los átomos entre si, menor será la fuerza total y por tanto el complejo será mas estable.

3.4. Descripción del Sistema

3.4.1. Plataforma de desarrollo

Ya que para cada pose obtenida es necesario calcular el puntaje para compararlo con otros puntajes obtenidos por distintas poses, es necesario crear el videojuego en un motor de videojuegos que permita realizar estos cálculos. Ese fue el mayor motivo por el cual el motor elegido fue Unity [10], específicamente su versión 2017.2 que era la versión estable más reciente al momento de empezar a trabajar en el prototipo. Unity es un motor que permite la realización de videojuegos tanto en 2D como 3D. El motor permite programar en C#, por lo que es posible implementar funciones de puntaje de una complejidad similar a las usadas en los software de simulación.

3.4.2. Formato de entrada

Los complejos proteína ligando se obtienen del sitio del Protein Data Bank, el cual los entrega en un archivo en formato PDB [7]. Estos archivos corresponden a representaciones tridimensionales de estos complejos. Para esta memoria se decidió usar un software de terceros para generar una representación bidimensional de los complejos a partir de los archivos PDB, El programa elegido fue LigPlot+, ya que este ofrece una licencia para uso académico y a través de los datos de su archivo de salida de formato DRW se pueden obtener las coordenadas espaciales de los átomos de la proyección generada.

3.4.3. Sistema

Cada sección de un videojuego creado en Unity se denomina escena. Los objetos que componen estas escenas son denominados **GameObject**. A estos objetos se le pueden otorgar componentes que determinarán el comportamiento que tendrán estos en el videojuego. Por defecto estos sólo incluyen el componente **Transform**, el cual les da la posición, rotación y escala dentro de la escena, además de su relación de parentesco con algún otro objeto.

Tomando esto como base, se decidió realizar 2 escenas para que fueran la base del videojuego. La primera corresponde al menú principal donde se podrán seleccionar el

complejo con el que se desea jugar, entre otras cosas, y la segunda corresponde a la escena donde se podrá jugar.

Si bien, uno puede programar componentes que determinen el comportamiento de los GameObjects que decida implementar en las escenas del videojuego, Unity ya tiene varios incluidos para agregarse de forma directa a los GameObjects.

Para poder representar estructuras moleculares dentro de la escena de juego, es necesario crear objetos que puedan ser instanciados varias veces, ya que las moléculas son conjuntos de átomos. Para esto se hizo uso de un tipo de objeto en Unity llamado **prefab**, el cual tiene la particularidad de poder ser instanciado en la escena por medio de un script y también de que un prefab puede estar compuesto de otros prefabs.

3.4.4. Arquitectura

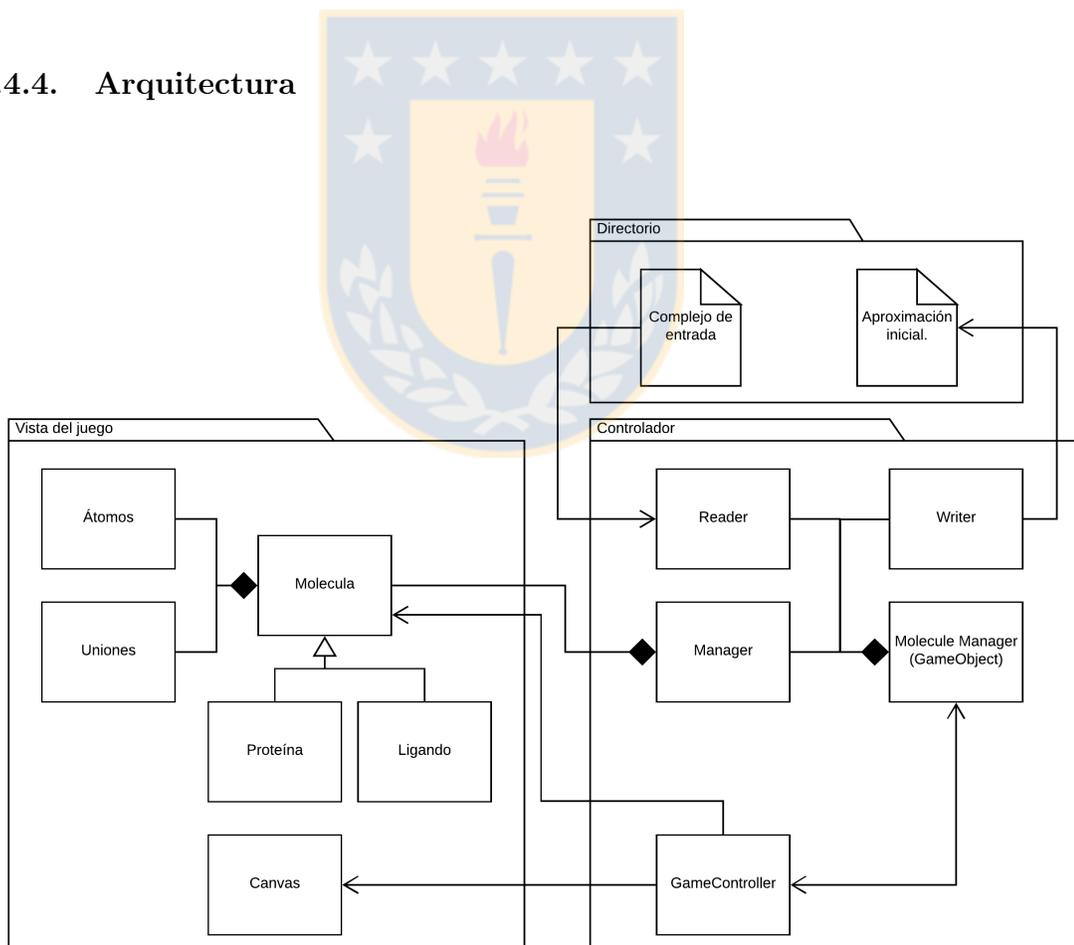


Figura 3.4: Arquitectura del juego

La arquitectura de la escena del juego corresponde a un patrón Modelo Vista Controlador. La Figura 3.4 no muestra el modelo directamente, ya que como los elementos son `GameObjects`, algunos de estos además de contener componentes que los colocan en la categoría de Vista o Controlador, también tienen componentes que contienen los datos del modelo.

La vista está compuesta por todos aquellos elementos del videojuego que se representan visualmente en el juego. Dentro de estos se incluyen las distintas moléculas y sus componentes que están generados como prefabs, o el **Canvas**, que contiene y despliega información relevante para el usuario, tal como el tiempo restante o el puntaje obtenido. El canvas es el único objeto de la vista que está inserto en la escena desde el inicio de esta.

A continuación se procederá a detallar los elementos que componen la vista

a) **Moléculas** Representación gráfica de las moléculas entregadas como entrada a partir de la proyección bidimensional creada por `LigPlot+`. Contienen su nombre, un `RigidBody2D` y conocen todos los átomos y uniones que la componen. Su comportamiento dentro del juego va a estar dado por el tipo de molécula que sea, entre las cuales se encuentran:

- a.1) **Ligandos** Es la única molécula que es controlada por el jugador. El jugador puede desplazarla, hacer que rote en torno a un átomo seleccionado por él o realizar un volteo con respecto al eje vertical. Su `RigidBody2D`² es de tipo `Dynamic`, por tanto las colisiones afectan a la posición y rotación de la molécula.
- a.2) **Proteínas** Estas moléculas parten alejadas del punto que les corresponde según las coordenadas de la proyección bidimensional. Mientras dura una partida, las proteínas se acercan a su posición de la proyección para estar en ella cuando el tiempo termine. Para lograr esto, el `RigidBody2D` es de tipo `Kinematic`, o sea afecta a los objetos que colisionan con el y que no son `Kinematic`, pero su movimiento no se ve afectado por esto.

²Componente de Unity que controla las físicas bidimensionales de un objeto

- b) **Átomos** Prefabs que forman parte de las moléculas. Son visibles para el jugador mediante un sprite de una esfera. El color y el tamaño de esta esfera será determinado por el elemento del átomo. La carga del átomo también depende del elemento. Estos datos se pueden observar en la Tabla 3.1. Ya que la representación del átomo es una esfera, estos contienen como componente un `CircleCollider2D`³.
- c) **Uniones** Son la representación visual de los átomos que están unidos dentro de una molécula. Ello lo logran mediante un `LineRender` entre ambos átomos. Para definir las áreas de colisión de estos se usó un `PolygonCollider2D`⁴ de 4 vértices que dependen de la posición de los átomos unidos por este objeto.
- d) **Canvas** La función de este elemento es de proveer información útil para el jugador, ya sea en formato de texto o de imágenes. Entre esta información se encuentra una cuenta regresiva para dar inicio al juego, el tiempo de juego restante, un aviso de que se acabó el tiempo, el puntaje final y el puntaje más alto actual.

Elemento	Símbolo	Carga	Radio	Color
Carbono	C	-1	0.65	Negro
Hidrógeno	H	1	0.85	Blanco
Oxígeno	O	-2	1	Rojo
Nitrógeno	N	2	1.15	Azul
Fósforo	P	-2	1.3	Amarillo
Azufre	S	3	1.45	Naranja

Tabla 3.1: Datos de cada átomo

Los elementos del controlador se encuentran insertos en la escena desde un inicio, su función es controlar la lógica del juego, formar las estructuras moleculares a partir de archivos y en caso de ser necesario, almacenar los datos de una buena aproximación inicial encontrada.

1. **Molecule Manager** La función de este `GameObject` es administrar todo lo referente a moléculas dentro del juego, para lograrlo lo hace a través de 3 scripts distintos

³Define un área de colisión circular para un `RigidBody2D`

⁴Define un área de colisión poligonal para un `RigidBody2D`

- a) **MoleculeReader** Su función principal es cuando se inicia el juego. Lee el archivo de entrada que fue indicado en el menú principal y a partir de él crea las moléculas con sus respectivos átomos y uniones. Además, después de creada una molécula la agrega a una lista del MoleculeManager.
 - b) **MoleculeManager** Almacena las moléculas del sistema, ya sean proteínas o ligando, al ser el único script que conoce todas las moléculas es el encargado de dirigir a estas para que calculen la energía total del sistema.
 - c) **MoleculeWriter** El rol de este script es guardar en un archivo las coordenadas finales de las moléculas y sus átomos en caso de obtener una puntuación que supere a la que había almacenada. Además modifica un archivo DRW para que ahora represente la configuración con la cuál se obtuvo ese puntaje.
2. **GameController** Este objeto se encarga de controlar el estado en el cuál se encuentra el juego y de cómo mostrar éste al jugador mediante la UI o por medio de sonidos. Además, este objeto detecta cuando se obtuvo un nuevo récord y actualiza la puntuación más alta obtenida para el complejo con el cuál se acaba de jugar.

Por otro lado, la escena del menú principal tiene la finalidad de seleccionar el complejo a jugar, enseñar los controles y mostrar las mejores puntuaciones para cada complejo.

Los datos de los complejos, tales como el nombre del complejo, del archivo que contiene los datos de este y las puntuaciones más altas se encuentran en un archivo JSON que se lee cuando la escena es cargada. Al momento de cargarse los datos, estos crean botones para mostrar los datos de cada complejo en la pantalla de selección de nivel.

Ya que era necesario traspasar los datos extraídos del JSON desde la escena del menú a la escena de juego se creó un GameObject que al momento de instanciarse se referencia a si mismo en una variable estática. Este GameObject debe estar presente en cada escena donde se vaya a utilizar. Para evitar duplicaciones que por lo general

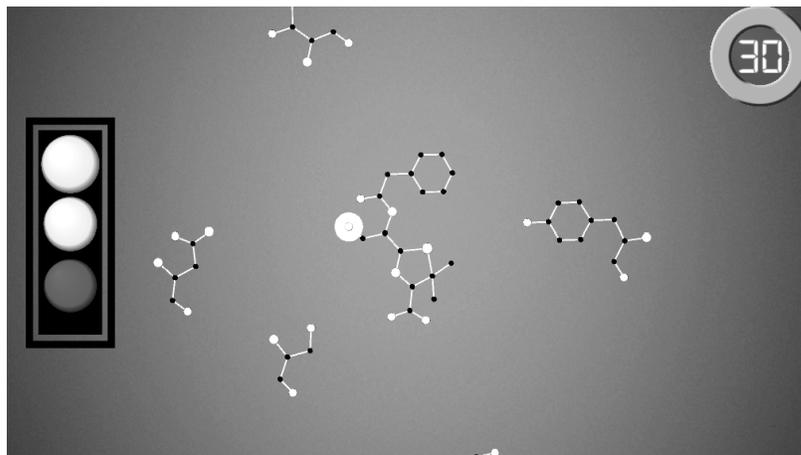


Figura 3.5: Captura de pantalla del videojuego

ocurren cuando se abre una escena que contenga este GameObject, el duplicado revisará si existe la variable estática con la referencia no es nula, en ese caso se destruye y sólo queda la instancia mas antigua.

3.4.5. El juego

En la escena de juego, al jugador se le muestra la representación bidimensional del complejo proteína ligando, tal como muestra la Figura 3.5. El juego parte cuando el semáforo de la izquierda ponga sus luces en verde, luego de esto, el semáforo desaparece desplazándose hacia la izquierda.

- El jugador controla el ligando, será la molécula central del juego.
- Las moléculas correspondiente a la proteína aparecerán alejadas de su posición del modelo bidimensional, a medida que transcurra el tiempo se irán acercando a esa posición de manera lineal.
- El jugador podrá mover el ligando durante 30 segundos. Estos segundos se muestran en un contador en la esquina superior derecha del juego.
- Los movimientos permitidos para el jugador son desplazamiento horizontal y vertical, rotación en sentido horario y antihorario, además de voltear en 180° con respecto a un eje de simetría vertical.

- La rotación en sentido horario y antihorario se realizan en torno a un átomo que se toma como eje de rotación. Ese átomo es resaltado del resto con un indicador. Se le permite al jugador cambiar libremente el átomo que utilizará como eje de rotación.
- El juego prioriza el movimiento de las moléculas de las proteínas por sobre el del ligando. Si estos colisionan, el ligando será empujado por la otra molécula, la cual continuará normalmente su movimiento.
- Cuando se realiza un volteo con respecto al eje de simetría vertical, el tiempo se detiene mientras se realiza el volteo, por tanto, el movimiento de las moléculas de la proteína también. Si durante el volteo, el ligando colisiona con otra molécula, el volteo se cancela y el ligando vuelve a su posición anterior.
- El puntaje se calcula cuando se acaba el tiempo. Para ello se remueve el control del ligando al jugador. Luego se le informa el puntaje obtenido al jugador como muestra la Figura 3.6.

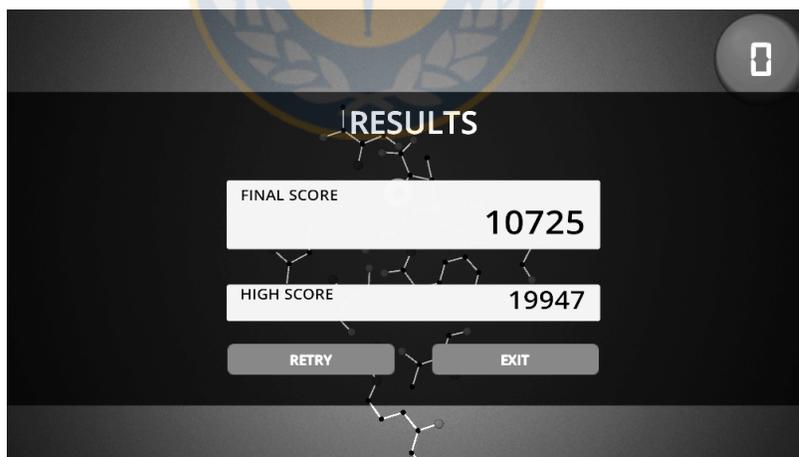


Figura 3.6: Pantalla de resultados al final termino del juego

Para calcular el puntaje, el juego utiliza una función compuesta por la ecuación 3.1. El puntaje entregado por esta función es inversamente proporcional a la fuerza electromagnética que generan los átomos del complejo: mientras más baja sea esta fuerza, el complejo es más estable y el puntaje es mayor. Si el puntaje obtenido es

superior al puntaje más alto actual, queda registrado como un nuevo récord y se le informa al jugador de que batió el récord anterior. A su vez, el juego genera 2 archivos que contienen la pose final que obtuvo tal puntaje. Estos archivos están en 2 formatos, el primero corresponde a una representación que puede ser leída por el videojuego y que puede generar un nuevo nivel con éste, mientras que el segundo es una representación en formato de LigPlot+.

3.4.6. Metodología utilizada

Al ser este un proyecto de desarrollo unipersonal, se adaptaron prácticas de métodos ágiles.

Antes de comenzar el desarrollo del videojuego, se realizaron múltiples reuniones con el cliente Alexis Salas Burgos. Estas reuniones sirvieron para conocer los requerimientos del cliente, discutir ideas para la implementación y definir los alcances de la memoria. En estas reuniones se definieron los temas a estudiar y su correspondiente bibliografía.

Una vez obtenido un conocimiento más completo del problema, se procedió a desarrollar un prototipo. El objetivo de este prototipo era que, a partir de coordenadas de moléculas escritas en un archivo de texto pudiese representarlas de forma visual en una escena de Unity. Tal prototipo fue mostrado al cliente y al profesor patrocinante, y posteriormente fue utilizado como la base para desarrollar el videojuego.

Durante el desarrollo, se realizaron reuniones con el cliente o el profesor patrocinante cada vez que se obtenían progresos importantes. Cada vez que se presentaron dudas con respecto a Docking Molecular o surgieron ideas con respecto al diseño del videojuego, se contactó con el cliente para conversar sobre esto.

Capítulo 4

Resultados

Para la evaluación de este producto, se comparó la solución obtenida en el juego con su representación en un programa de conversión y visualización de diagramas bidimensionales. Además midió el rendimiento del videojuego con respecto a su estabilidad de fotogramas por segundo. Se realizaron diversas pruebas con usuarios que jugaron el videojuego en su estado de desarrollo y final, pudiendo obtener retroalimentación de ellos. A partir de esta retroalimentación y de lo concluido a través del desarrollo, se pudo determinar cuales son los aspectos a mejorar cuando se continúe con el desarrollo del videojuego. Por último, fue posible comparar lo que ofrece este videojuego en relación a otros enfoques para diseñar fármacos, incluyendo una comparativa con su rival mas cercano, UDock.

4.1. Pose final en el juego y en LigPlot+

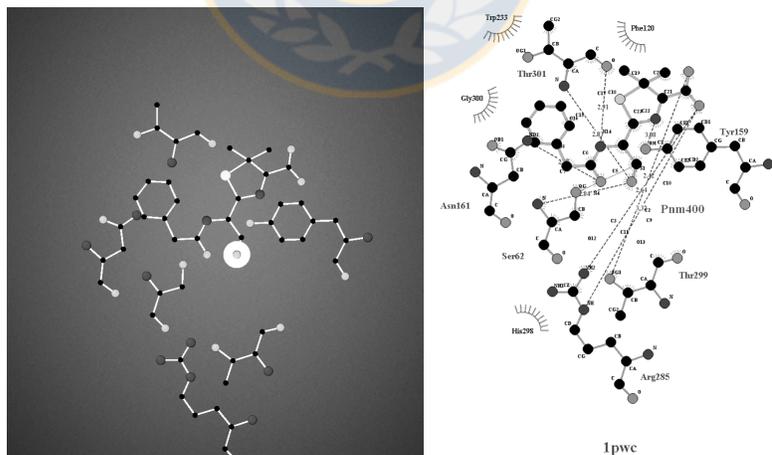


Figura 4.1: Comparativa de la pose final en el videojuego contra representación en LigPlot+

Como fue descrito previamente, el videojuego al obtener un nuevo puntaje alto

para un complejo en particular, genera un archivo que puede ser leído en el programa de generación y visualización de diagramas bidimensionales LigPlot+. Para las pruebas, se eligió tomar una captura de pantalla en el momento justo cuando se registra el puntaje alto y se genera el archivo en formato DRW. Estas acciones fueron realizadas dentro de la función Update, la cual es una función que se puede definir para cada objeto de la escena y se ejecuta en cada cuadro del juego. Cabe destacar que el siguiente cuadro no se ejecuta hasta que todas las acciones de la función Update del cuadro actual se hayan ejecutado. En este caso estas acciones sólo se ejecutaron en un solo cuadro, por lo que la captura realizada corresponde exactamente a las coordenadas que se están escribiendo en el archivo DRW en tal momento. En la figura 4.1 se puede apreciar una comparación entre el cuadro renderizado al momento de generar el DRW y la representación de ese archivo en LigPlot+. Se espera que a este archivo DRW se le pueda realizar el procedimiento inverso que realiza LigPlot+ para así poder obtener un PDB tridimensional con la rotación y traslación del ligando aplicadas, para que pueda ser utilizado en ayudar a los métodos de simulación.

4.2. Rendimiento

El videojuego fue capaz de ser ejecutado sin mayor complicaciones dentro del editor de Unity en 2 computadores distintos, cuyas especificaciones se muestran en la tabla 4.1. En ambos la tasa de cuadros por segundo no disminuyó de los 60, la cuál es ya considerada la tasa de refresco estándar de la gran mayoría de pantallas. Esta tasa se mantuvo estable incluso cuando se ejecutó los cálculos de la función de puntaje en tiempo real.

Característica	Computador 1	Computador 2
Procesador	Intel Core i7-6700	Intel Core i5-3470S
Frecuencia	2.60GHz	2.90GHz
RAM	12GB	4GB
T. de Video	NVidia GTX960M	Intel HD Graphics 2500
S.O.	Windows 10	Windows 10

Tabla 4.1: Especificaciones de los computadores usados

Posteriormente se realizaron pruebas de rendimiento con una versión ejecutable

del juego. Esta versión fue compilada para ser ejecutada en una edición de Windows de 64bits. Al igual que al ser ejecutada en el editor, la tasa de cuadros por segundo no disminuyó de los 60 cuadros. Los únicos momentos en los cuales esta tasa de cuadro disminuye, ya sea en el editor como en el ejecutable, es cuando se cambia de una escena a otra, lo cual no afecta a la jugabilidad.

4.3. Pruebas con usuarios

Se realizaron pruebas con los usuarios durante el desarrollo y también posterior a este. Estas pruebas incluyeron 3 usuarios con conocimiento del tema y 6 sin conocimientos de este. Estos también se distribuían en videojugadores frecuentes y ocasionales.

La mayoría usó el teclado para controlar el videojuego, aunque algunos tuvieron la oportunidad de usar un control análogo para controlar el videojuego. Si bien algunos de los jugadores ocasionales tuvieron problemas para acostumbrarse a controlar el juego mediante teclado, a la tercera partida ya podían realizar todos los movimientos de forma natural. Aquellos que tuvieron la oportunidad de usar control análogo, lo prefirieron por sobre el teclado y destacaron que podían realizar movimientos mas precisos con este.

Al realizar ciertos movimientos, algunos jugadores superponían el ligando con algunas moléculas, tal como muestra la figura 4.2. La mayor crítica al videojuego fue no tener una forma para saber si los movimientos realizados servían para mejorar el puntaje final, ya que este sólo podía ser mostrado al finalizar el tiempo.

4.4. Aspectos a mejorar

A partir de las pruebas de usuario y durante el desarrollo del videojuego se encontraron varios detalles a mejorar para futuras iteraciones, algunos de estos son referentes en cuanto a la jugabilidad propuesta. El objetivo de estas mejoras es que se informe al jugador en tiempo real el puntaje que está obteniendo, así como también evitar resultados incorrectos.

Las moléculas pertenecientes a la proteína se van acercando gradualmente en el

tiempo hacia el centro del escenario. Como estas no se detienen hasta llegar a sus coordenadas originales, si 2 de ellas impactan al ligando en extremos opuestos (como muestra la figura 4.2), el ligando queda sobrepuesto a estas moléculas y no hay forma de zafarlo hasta que se agote el tiempo. La solución propuesta en este caso es que el ligando pueda detectar cuando ocurra esta superposición y así se pueda anular la partida avisando al jugador de lo ocurrido.

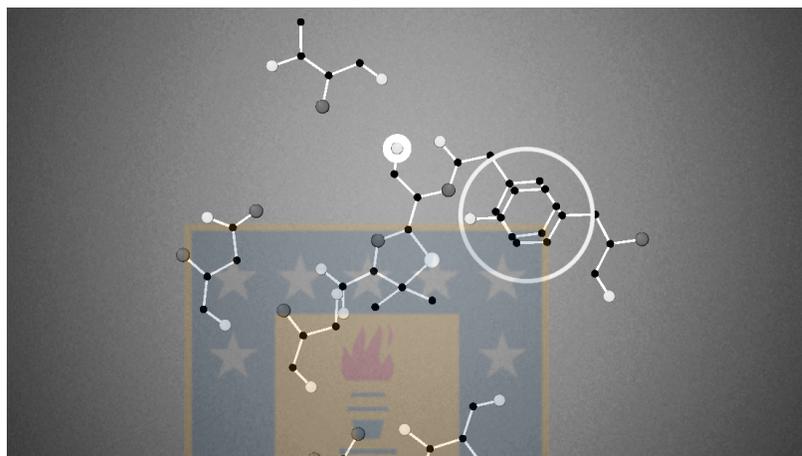


Figura 4.2: Superposición de moléculas

Otro problema detectado con la jugabilidad, es que el jugador no puede saber su puntaje hasta que el tiempo de juego se haya agotado. Esto se debe a que el puntaje calculado es inversamente proporcional a la distancia que hay entre los átomos de las moléculas, y cómo estos parten alejados unos de otros, el puntaje comienza con un valor alto que va disminuyendo con el tiempo. Para solucionar esa problemática se propone cambiar la jugabilidad por una donde las moléculas de proteínas siempre se encuentren en una posición fija desde el inicio y que el jugador pueda mover el ligando libremente por la zona de interacción. Para poder realizar esto, los colisionadores del ligando deben pasar a ser triggers, es decir, áreas que detectan cuando un objeto ingrese a ellas, para que detecten cuando un colisionador de otro RigidBody entre en el área del trigger. El juego tiene que informar al jugador los colisionadores que están sobrepuestos y no permitir al jugador que entregue un resultado a menos que el ligando no esté sobrepuesto a la molécula. De esta forma, ya que las moléculas de proteína se mantienen siempre en las mismas coordenadas, la función de puntaje

entregará siempre un puntaje valido a menos que las moléculas estén sobrepuestas.

Se puede evaluar la opción de implementar flexibilidad en los ligandos, ya que por un lado entrega mayor exactitud a los resultados, pero por otro esto significa tener que realizar modificaciones en cuanto a la jugabilidad.

4.5. Comparativa con otros enfoques

Este videojuego está diseñado para apoyar al enfoque de la simulación. La principal ventaja es que permite que gente ajena al problema pueda aportar al diseño de fármacos. Cabe destacar que los cálculos para llegar a estas soluciones iniciales estarán dados por los dispositivos de los jugadores y la posibilidad de encontrar una solución inicial que sirva dependerá de la intuición de algunos jugadores entre miles de ellos.

Si bien el objetivo es idéntico es idéntico al de UDock, el videojuego desarrollado sacrifica precisión en los modelos de las moléculas y resultados con tal de obtener mayor simpleza en la visualización de estas. Esto también permite abarcar otros tipos de complejos que con UDock serían muy difíciles de manipular, ya que a través de la representación de LigPlot+ sólo se representa el sitio activo de la proteína permitiendo al jugador jugar sólo dentro de este.

Capítulo 5

Conclusiones

Concluyendo y resumiendo lo realizado, podemos decir que después de una investigación sobre el método de Docking Molecular y la de ludificación aplicada a problemas de bioinformática, se pudo desarrollar un videojuego para ludificar el Docking Molecular. El proceso de proyectar un modelo molecular tridimensional a un diagrama bidimensional mediante un programa externo permitió trasladar este diagrama a un escenario de juego completamente funcional e interactivo. La forma modular en que se diseñó permite la inclusión de distintos diagramas para generar distintos escenarios de juego. A la vez permite la extracción de las mejores soluciones para que sean visualizadas en programas como LigPlot+.

El trabajo realizado permitió cumplir los objetivos planteados en un principio. Ya que el trabajo de esta memoria corresponde a sólo una parte del proyecto total, no se puede comprobar todavía si las aproximaciones iniciales generadas por el videojuego mejoran los resultados obtenidos mediante simulación. Pese a aquello, la base creada es lo suficientemente completa para calificar el objetivo general como logrado.

Fue desafiante tener que investigar un área ajena a lo estudiado durante la carrera como lo es la bioinformática. Si bien la bioinformática contiene ciertas similitudes con la informática, el problema desarrollado involucra muchos conceptos de las ciencias biológicas que no son cubiertos en la carrera. Se requirió tener claros estos conceptos para poder definir los alcances del trabajo de la memoria de título.

Gran parte de las dificultades encontradas al momento de desarrollar el videojuego tienen su origen al tratar de implementar las colisiones de este. Ya que las moléculas son de estructura variable, sus áreas de colisión también lo es. Y como el área de colisión es diferente para cada elemento que conforma la molécula, hacer que estas áreas se ajustaran a la forma de estos requirió bastante trabajo.

Como fue visto en la sección 3.2 y basándonos en los analizados en la sección 2.2,

los resultados del videojuego deben ser almacenados una plataforma en línea, lo que sera prioritario desarrollar más adelante. Ya que el objetivo principal de este y los otros videojuegos es que mediante la actividad de miles de jugadores se pueda llegar a aquellas soluciones que resuelven el problema objetivo.

Es necesario mejorar la función de puntaje, ya que la que está implementada sólo depende de la Ley de Coulomb por motivos de desarrollo y experimentales. Gracias a la modularidad con la que se realizó el programa se podría modificar fácilmente la función y agregar los parámetros que fueran requeridos por esta.

La plataforma destino para este trabajo fue Windows de 64 bits, ya que fue el sistema operativo en el que se desarrolló. Si se quiere llegar a la mayor cantidad de jugadores, se debe realizar una adaptación de este videojuego para otras plataformas como WebGL, Mac, Linux, Android o iOS. Unity permite desarrollar para estas plataformas, y la exportación del proyecto no requiere de tanto trabajo a nivel de código. Para los dispositivos móviles, la interfaz debe ser rediseñada para adaptarse a las capacidades táctiles de estos dispositivos.

Por último, se debe crear un método para traspasar de un archivo DRW a uno PDB, es decir, de un diagrama bidimensional a un modelo tridimensional. Esto es necesario, ya que si se desea comprobar que las soluciones entregadas por el videojuego sirven como aproximación inicial para el problema real de diseño de fármacos, esta solución al menos debe estar en un formato que pueda ser ejecutado en algún programa de simulación.

Bibliografía

- [1] AutoDock Vina - molecular docking and virtual screening program. <http://vina.scripps.edu/>.
- [2] BioSolveIT GmbH - PoseView. <https://www.biosolveit.de/poseview/>.
- [3] Bullet | Real-Time Physics Simulation. <http://bulletphysics.org/wordpress/>.
- [4] LigPlot+ home page. <https://www.ebi.ac.uk/thornton-srv/software/LigPlus/>.
- [5] Overview of DOCK. http://dock.compbio.ucsf.edu/Overview_of_DOCK/index.htm.
- [6] Phylo | DNA puzzle. <https://phylo.cs.mcgill.ca/>.
- [7] RCSB PDB: Homepage. <https://www.rcsb.org/>.
- [8] Solve Puzzles for Science | Foldit. <https://fold.it>.
- [9] Udock. <https://udock.fr/>.
- [10] Unity. <http://unity3d.com/>.
- [11] Óscar Coltell, María Arregui, Antonio Fabregat, and Olga Portolés. La bioinformática en la práctica médica: Integración de datos biológicos y clínicos. *Revista médica de Chile*, 136(5):645–652, May 2008.
- [12] Vickie Curtis. Online citizen science games: Opportunities for the biological sciences. *Applied & Translational Genomics*, 3(4):90–94, December 2014.
- [13] Robert Kleffner, Jeff Flatten, Andrew Leaver-Fay, David Baker, Justin B. Siegel, Firas Khatib, and Seth Cooper. Foldit Standalone: a video game-derived protein structure manipulation interface using Rosetta. *Bioinformatics (Oxford, England)*, May 2017.
- [14] Daniel Kwak, Alfred Kam, David Becerra, Qikuan Zhou, Adam Hops, Eleyine Zarrour, Arthur Kam, Luis Sarmenta, Mathieu Blanchette, and Jérôme Waldispühl. Open-Phylo: a customizable crowd-computing platform for multiple sequence alignment. *Genome Biology*, 14(10):R116, 2013.
- [15] Roman A. Laskowski and Mark B. Swindells. LigPlot+: multiple ligand-protein interaction diagrams for drug discovery. *Journal of Chemical Information and Modeling*, 51(10):2778–2786, October 2011.

- [16] Guillaume Levieux, Guillaume Tiger, Stéphanie Mader, Jean-François Zagury, Stéphane Natkin, and Matthieu Montes. Udock, the interactive docking entertainment system. *Faraday Discussions*, 169:425–441, 2014.
- [17] Stephan Lorenzen and Yang Zhang. Monte Carlo refinement of rigid-body protein docking structures with backbone displacement and side-chain optimization. *Protein Science : A Publication of the Protein Society*, 16(12):2716–2725, December 2007.
- [18] Xuan-Yu Meng, Hong-Xing Zhang, Mihály Mezei, and Meng Cui. Molecular Docking: A powerful approach for structure-based drug discovery. *Current computer-aided drug design*, 7(2):146–157, June 2011.
- [19] Lic Hairol Romero Sandí. La Gamificación como participante en el desarrollo del B-learning: Su percepción en la Universidad Nacional, Sede Regional Brunca. page 10, 2013.
- [20] Brian K. Shoichet, Irwin D. Kuntz, and Dale L. Bodian. Molecular docking using shape descriptors. *Journal of Computational Chemistry*, 13(3):380–397.
- [21] Akash Singh, Faizy Ahsan, and Mathieu Blanchette. Lessons from an online massive genomics computer game. page 10.

