



**UNIVERSIDAD DE TALCA
FACULTAD DE INGENIERÍA
ESCUELA DE INGENIERÍA CIVIL EN COMPUTACIÓN**

**DISEÑO DE UNA INTERFAZ GRÁFICA PARA
BÚSQUEDA DE PATRONES
ESTRUCTURALES EN EL PROTEIN DATA
BANK**

DIEGO ALEJANDRO ANDRÉS CISTERNA VÁSQUEZ

Profesor Guía: RENZO ANGLES ROJAS

Memoria para optar al título de
Ingeniero Civil en Computación

Curicó – Chile
Julio, 2017



**UNIVERSIDAD DE TALCA
FACULTAD DE INGENIERÍA
ESCUELA DE INGENIERÍA CIVIL EN COMPUTACIÓN**

**DISEÑO DE UNA INTERFAZ GRÁFICA PARA
BÚSQUEDA DE PATRONES
ESTRUCTURALES EN EL PROTEIN DATA
BANK**

DIEGO ALEJANDRO ANDRÉS CISTERNA VÁSQUEZ

Profesor Guía: RENZO ANGLES ROJAS

Profesor Informante: RODRIGO PAREDES

Profesor Informante: CÉSAR ASTUDILLO

Memoria para optar al título de
Ingeniero Civil en Computación

El presente documento fue calificado con nota: _____

Curicó – Chile

Julio, 2017

Dedicado a mis padres.

AGRADECIMIENTOS

En primer lugar, me gustaría agradecer sinceramente a mi profesor guía, Dr. Renzo Angles, por la orientación que me entregó sobre todos los aspectos relacionados a este proyecto de memoria, también por sus conocimientos, su forma de trabajar y de enseñar, su paciencia, su dedicación y su motivación frente al tema desarrollado.

En segundo lugar, quiero agradecer al profesor de bioinformática, Dr. Mauricio Arenas, por las explicaciones de conceptos propios su área, por desempeñar de cierta forma el rol de cliente de la herramienta desarrollada y por ayudarme en el proceso de evaluación. También agradezco a sus estudiantes que colaboraron en la evaluación.

En tercer lugar, me gustaría agradecer a todos los profesores que me hicieron clases, ya que ellos me guiaron y me brindaron importantes conocimientos a través de los diferentes módulos que constituyen la carrera de Ingeniería Civil en Computación.

Finalmente, quiero agradecer a mis compañeros ya que algunos de ellos me dieron muy buenas recomendaciones que hicieron más fácil mi progreso, y en especial a todos aquellos con los que hice equipo en los diferentes módulos.

TABLA DE CONTENIDOS

	página
Dedicatoria	I
Agradecimientos	II
Tabla de Contenidos	III
Índice de Figuras	VI
Índice de Tablas	VIII
Resumen	IX
Abstract	x
1. Introducción	1
1.1. Definición del problema	1
1.2. Propuesta	2
1.3. Hipótesis	3
1.4. Metodología	3
2. Contexto	5
2.1. El Protein Data Bank	5
2.2. Proteínas, aminoácidos y ligandos	6
2.3. Herramientas de consulta para PDB	8
2.4. Búsqueda de patrones estructurales en PDB	10
2.5. Herramientas basadas en grafos para búsqueda de patrones en PDB .	13
3. Diseño y Construcción	16
3.1. Requisitos de la aplicación	16
3.1.1. Requisitos funcionales	16
3.1.2. Requisitos no funcionales	17
3.2. Arquitectura del sistema	18
3.3. Componentes y asociaciones en un patrón	20

3.4. Mockups y elementos de la interfaz	22
3.5. Interacción con la interfaz	25
3.6. Estructura de la base de datos	27
3.7. Confección de consultas desde el patrón gráfico	30
3.7.1. Asociaciones de componentes a consultas SQL	30
3.7.2. Definición inductiva de patrones y ejemplo de transformación .	37
3.7.3. Resultados repetidos para un patrón diseñado correctamente .	40
3.8. Prototipo final	42
3.9. Optimización de la base de datos	45
4. Verificación, Validación y Evaluación	48
4.1. Verificación de funcionalidades	48
4.2. Validación de funcionalidades	51
4.3. Evaluación de usabilidad	58
4.3.1. Metodología del proceso	58
4.3.2. Aspectos de la evaluación	58
4.3.3. Análisis de resultados	59
5. Conclusiones	64
5.1. Conclusiones generales	64
5.2. Aportes del proyecto	65
5.3. Trabajo futuro	66
Glosario	68
Bibliografía	69
Anexos	
A: Extracto archivo de la Hemoglobina (4HHB.pdb)	73
B: Lista de herramientas para PDB revisadas	74
C: Requisitos funcionales de la aplicación	78
D: Códigos SQL para creación de vistas materializadas	85

E:	Capturas de la aplicación web	87
F:	Códigos SQL asociados a los patrones de prueba	90
G:	Documentos para evaluación de usabilidad	93
H:	Resumen evaluaciones de criterios de usabilidad	99

ÍNDICE DE FIGURAS

	página
2.1. Niveles de estructura en las proteínas	7
2.2. Página oficial de PDBj Mine.	8
2.3. Página oficial de Ligand Expo.	9
2.4. Página oficial de PPI.	10
2.5. Distancias entre un ligando y un aminoácido	11
2.6. Distancias entre dos aminoácidos	12
2.7. Ejemplo de consulta en PatternQuery	13
2.8. Marvin mostrando la estructura química de la Atorvastatina.	14
2.9. STRING mostrando la red de interacciones para Hemoglobina Beta.	15
3.1. Arquitectura y proceso de datos.	19
3.2. Panel donde se diseña el patrón estructural.	23
3.3. Panel donde se muestran los resultados de la búsqueda.	24
3.4. Modal donde se muestra la representación 3D de cada resultado.	25
3.5. Patrón estructural resultante siguiendo los pasos del ejemplo.	27
3.6. Esquema relacional para PDB.	28
3.7. Vistas materializadas.	29
3.8. Ligando asociado a un aminoácido fijo mediante distancia.	30
3.9. Ligando asociado a un aminoácido comodín mediante distancia.	31
3.10. Dos aminoácidos fijos asociados mediante distancia.	32
3.11. Aminoácido fijo asociado a un aminoácido comodín mediante distancia.	33
3.12. Dos aminoácidos comodín asociados mediante distancia.	33
3.13. Dos aminoácidos fijos asociados mediante secuencia.	34
3.14. Aminoácido fijo asociado a un aminoácido comodín mediante secuencia.	35
3.15. Aminoácido comodín asociado a un aminoácido fijo mediante secuencia.	35
3.16. Dos aminoácidos comodín asociados mediante secuencia.	36
3.17. Patrón de ejemplo para transformación.	38
3.18. Patrón estructural con resultados repetidos.	40
3.19. Ejemplos de casos donde se repiten los aminoácidos en el resultado.	41
3.20. Panel donde se diseña el patrón estructural.	42
3.21. Panel donde se muestran los resultados de la búsqueda.	43

3.22. Modal donde se muestra la representación 3D de cada resultado.	44
3.23. Patrón de prueba número uno.	46
3.24. Patrón de prueba número dos.	46
3.25. Patrón de prueba número tres.	47
4.1. Patrón objetivo para ejecutar el método de prueba manual.	52
4.2. Resultados obtenidos al ejecutar el método de prueba manual.	55
4.3. Resultados obtenidos de la búsqueda del patrón en la herramienta.	56
4.4. Vista tridimensional del resultado #1.	57
4.5. Vista tridimensional del resultado #4.	57
4.6. Ambiente de evaluación a nivel de software.	60

ÍNDICE DE TABLAS

	página
4.1. Resumen de cumplimiento para lista de criterios.	61
C.1. Requisito funcional: Agregar ligando.	78
C.2. Requisito funcional: Agregar aminoácidos.	79
C.3. Requisito funcional: Agregar aminoácidos comodín.	79
C.4. Requisito funcional: Identificar ligando y aminoácidos.	80
C.5. Requisito funcional: Agregar conector distancia.	80
C.6. Requisito funcional: Agregar conector siguiente.	81
C.7. Requisito funcional: Incorporar propiedades.	81
C.8. Requisito funcional: Eliminar componentes.	82
C.9. Requisito funcional: Buscar patrón en PDB.	82
C.10. Requisito funcional: Mostrar resultados de búsqueda.	83
C.11. Requisito funcional: Enlazar resultados a PDB.	83
C.12. Requisito funcional: Visualizar resultados en JSmol.	84

RESUMEN

La estructura que posee la materia define sus propiedades y funciones. En las últimas décadas, gracias a los avances de la ciencia y la tecnología, investigadores han utilizado variadas técnicas para determinar la estructura de macromoléculas tales como proteínas, ADN y ARN. La información conseguida es útil para describir las interacciones que se dan entre moléculas, y así poder entender las estructuras y funciones que permiten muchos procesos biológicos importantes.

El volumen de información de macromoléculas que se maneja hoy en día es muy grande. Por esto se han creado bancos de datos que intentan unificar y mantener accesible la información a nivel global. Este es el caso del Protein Data Bank (PDB). Sin embargo, poder analizar los datos almacenados es una tarea difícil, que sólo se lleva a cabo mediante herramientas de software especializadas.

La mayoría de las herramientas disponibles actualmente poseen interfaces estándar y se limitan a realizar búsquedas a través de formularios, con los cuales no se pueden elaborar consultas avanzadas que entreguen datos realmente trabajados. Para efectuar una consulta compleja, las herramientas ofrecen confeccionarlas y ejecutarlas directamente, por ejemplo, en código SQL o a través de lenguajes de programación.

Este proyecto se enfoca en entregar una solución práctica para un problema específico, la búsqueda de patrones estructurales a nivel de interacción entre un ligando y los aminoácidos que lo rodean, en su sitio de unión en las macromoléculas. La solución consiste en el diseño de una interfaz gráfica que permita dar forma a un patrón estructural basado en elementos de grafo, y que luego tal patrón sea transformado en una consulta SQL para buscar coincidencias dentro de una base de datos relacional que contiene la información de PDB.

Respecto a las contribuciones, en este proyecto se implementó un prototipo funcional que satisface las necesidades originadas por el problema abordado, el prototipo fue validado, y evaluado positivamente por un grupo de usuarios cualificados. Además, en el proceso se desarrolló un método que permite transformar asociaciones de pares de componentes gráficos en consultas SQL, y que al unirlos pueden representar patrones estructurales de alta complejidad. Por último, el aporte a la comunidad científica es facilitar el estudio de estructuras por medio de una interfaz comprensible y fácil de usar, que revoluciona el modo de interacción habitual.

Palabras clave Patrones estructurales · Interacciones proteína–ligando · Interfaz gráfica de usuario · Sitio de unión · PDB

ABSTRACT

The structure of the matter defines its properties and functions. In the last decades, thanks to the advances of the science and the technology, researchers have used various techniques to determine the structure of macromolecules such as proteins, DNA, and RNA. The information obtained is useful to describe the interactions occurring between molecules, and thus be able to understand the structures and functions that allow many important biological processes.

The volume of information of macromolecules that is handled today is very big. Therefore, data banks have been created to try unifying and maintaining accessible the information at a global level. This is the case of the Protein Data Bank (PDB). However, being able to analyze the stored data is a difficult task, that is only carried through specialized software tools.

The majority of the tools available currently have standard interfaces and are limited to execute searches through forms, with which is not possible to elaborate advanced queries that return data actually worked. To make a complex query, the tools offer to compose them and execute them directly, for example, in SQL code or through programming languages.

This project focuses on delivering a practical solution to a specific problem, the search for structural patterns at an interaction level between a ligand and the amino acids surround it, in their binding site in the macromolecules. The solution consists of the design of a graphic interface that allows giving form to a structural pattern based on graph elements, and that then such pattern be transformed into a SQL query to look for coincidences inside a relational database that contains the information of PDB.

Respect to the contributions, in this project was implemented a functional prototype that satisfies the necessities caused by the problem addressed, the prototype was validated, and evaluated positively by a group of qualified users. Moreover, in the process was developed a method that allows transforming associations of pairs of graphic components into SQL queries, and that by uniting them can represent structural patterns of high complexity. Finally, the contribution to the scientific community is to facilitate the study of structures through an understandable and easy to use interface, what revolutionizes the habitual way of interaction.

Keywords Structural patterns · Protein–ligand interactions · Graphical user interface · Binding site · PDB

1. Introducción

En este primer capítulo se presenta la justificación del tema seleccionado y el problema abordado en esta memoria, seguido de la propuesta de solución que se desarrolla a través del documento. Además, se expone la hipótesis y la metodología utilizada para la consecución de los objetivos del proyecto.

1.1. Definición del problema

Las propiedades y funciones de moléculas orgánicas como las proteínas y sus complejos ensamblajes, son definidas por su estructura molecular [21].

Con los avances de la ciencia y la tecnología, variadas técnicas se han desarrollado para determinar la estructura de macromoléculas, consiguiendo modelos precisos a un nivel de resolución atómica en forma rápida y confiable. Clarificar los detalles de moléculas que interaccionan es particularmente importante, ya que con esto se pueden describir de mejor manera los mecanismos de enzimas y sus reacciones, y además es esencial para el diseño de fármacos guiado por estructuras.

En adición a lo anterior, en los últimos años ha crecido el interés por identificar varios tipos de inhibidores para su uso terapéutico en el tratamiento de muchas enfermedades. Esto ha iniciado una etapa donde se busca caracterizar la mayor cantidad de estructuras que sean posibles, y en este proceso las herramientas computacionales han sido y son un elemento clave [16].

Muchas funciones llevadas a cabo por proteínas no se podrían cumplir sin la presencia de pequeñas moléculas denominadas ligandos. La interacción entre proteínas y ligandos es altamente diversa, y depende del ambiente coordinador existente en su sitio de unión. Entender la estructura y los aspectos dinámicos de estas uniones facilitan la comprensión de la estructura y funciones de las proteínas en general [2].

A causa de lo explicado anteriormente, es que el estudio de interacciones específicas de una macromolécula con sus ligandos es un campo de investigación activo.

La necesidad de analizar grandes conjuntos de datos para comprender funciones biológicas que exhiben los organismos requiere del desarrollo de modernos sistemas de gestión de información que faciliten la manipulación y consulta de datos, y así crear recursos disponibles para la investigación biológica.

El problema abordado radica en que la mayoría de herramientas existentes, para explorar una base de datos biológicos y extraer la información deseada, se limitan a realizar búsquedas a través de formularios con los cuales no se puede elaborar una consulta que entregue datos realmente trabajados, sino que más bien se obtienen datos estadísticos simples o subconjuntos de información. Cuando uno está interesado en consultas avanzadas, las herramientas actuales ofrecen llevarlas a cabo directamente, teniendo que aprender para esto complejos esquemas de base de datos o funciones que son parte de lenguajes de programación, dificultando en gran medida el trabajo de la comunidad científica.

1.2. Propuesta

Debido a la relevancia intrínseca del tema, es que en esta memoria se pretende aportar con una solución práctica. El diseño de una herramienta computacional que facilite el análisis de macromoléculas que yacen almacenadas en el banco de datos de proteínas más importante en la actualidad.

En concreto, es el diseño de una interfaz gráfica para la búsqueda de patrones estructurales en el Protein Data Bank (PDB). La interfaz gráfica permitiría diseñar un patrón estructural compuesto por un ligando y los aminoácidos a su alrededor, simulando su sitio de unión dentro de las macromoléculas.

Para dar forma al patrón de interés la interfaz trataría a los componentes biológicos como elementos de grafo, es decir, los nodos representarían el ligando y los aminoácidos y las aristas corresponderían a sus asociaciones. Los grafos se adaptan de manera natural a este tipo de estructuras moleculares, siendo muy útiles como medio de abstracción en estos casos.

Luego, el patrón estructural debe ser transformado a una consulta en lenguaje SQL que se ejecutaría sobre una base de datos relacional que contiene la información de PDB. Así se lograría encontrar todas las coincidencias que existen para el patrón

objetivo de la búsqueda.

Finalmente, cada resultado debe ser mostrado de forma independiente, siguiendo un formato de fácil interpretación, y apoyándose de complementos comunes dentro de las herramientas de biología estructural como, por ejemplo, un visor de moléculas para ver su representación tridimensional.

1.3. Hipótesis

Es posible diseñar una interfaz gráfica que permita dar forma a un patrón estructural basado en elementos de grafo, y que luego tal patrón pueda ser transformado a una consulta en SQL para buscar coincidencias dentro de las macromoléculas almacenadas en el Protein Data Bank (PDB).

1.4. Metodología

Primero, se investigan las interfaces de consulta existentes para PDB. En esta primera etapa se hace una revisión de las herramientas y servicios, en su mayoría aplicaciones web, que hacen uso de PDB, y se analizan las herramientas de búsqueda más representativas del conjunto revisado.

Segundo, se identifican y definen los requisitos para la búsqueda de patrones estructurales en PDB. En principio, se consulta con expertos acerca de búsquedas recurrentes que puedan tratarse de forma innovadora, luego se especifican los requisitos funcionales para modelar y representar un patrón estructural.

Tercero, se diseña la interfaz gráfica definiendo sus componentes y su dinámica. Aquí se empieza por definir elementos adecuados para constituir la interfaz gráfica, luego se bosquejan las partes principales de la interfaz abarcando todos los requisitos especificados, por último se diseña la interfaz empleando tecnologías web.

Cuarto, se implementa el prototipo funcional de la herramienta. En esta fase se acondiciona una base de datos predefinida para dar respuesta a las consultas que conlleva la búsqueda de patrones. Luego, se implementa la lógica de la aplicación, se integra con la interfaz, y se conecta a la base de datos. Finalmente, se ejecutan pruebas para comprobar el funcionamiento del sistema.

Quinto, se evalúa el funcionamiento y la usabilidad de la interfaz. Aquí se validan las funcionalidades y se aplica un tipo de optimización a la base de datos. Poste-

riormente, se diseñan los instrumentos para evaluar la usabilidad de la interfaz y se aplican a un conjunto restringido de usuarios. Por último, se analizan los resultados del proyecto y se obtienen conclusiones.

2. Contexto

Este proyecto se centra en el diseño de un nuevo tipo de interfaz que permita buscar patrones estructurales sobre los datos del Protein Data Bank, por esto, a continuación se da a conocer una introducción a PDB y a su contenido, herramientas existentes para utilizar su información, y una introducción a la búsqueda de patrones estructurales y herramientas basadas en grafos que sirven para encontrarlos.

2.1. El Protein Data Bank

El Protein Data Bank (PDB) es un repositorio para almacenar información sobre estructuras cristalinas tridimensionales de macromoléculas biológicas, tales como proteínas y ácidos nucleicos. PDB se establece en el año 1971 en el Brookhaven National Laboratory, a manos de Walter Hamilton y Edgar Meyer, como respuesta al gran interés mostrado de parte de cristalógrafos, bioinformáticos y químicos por analizar y visualizar estructuras de proteínas [5].

En principio PDB fue lanzado con sólo siete estructuras, pero desde entonces ese número ha crecido en forma exponencial con el pasar de los años. Esto es debido principalmente al crecimiento de la biología estructural que trajo consigo nuevos métodos para la producción de proteínas, cristalización, recolección de datos, y análisis estructural en continuo desarrollo [4]. Además, se debe a la mejor disposición sobre compartir nueva información dentro de la comunidad científica, y a las facilidades que presenta Internet en el proceso de deposición y distribución de los datos.

Conforme avanza el tiempo, PDB se ha convertido en un recurso importante no sólo para grupos expertos, sino que también para investigadores de diversas áreas como bioquímica, física, informática y estudiantes de todos niveles que hacen uso de

su información alrededor del mundo. Esta condición hace que PDB deba permanecer global, públicamente disponible y uniforme [4]. De este modo, en 2003 se forma el Worldwide Protein Data Bank (wwPDB) entre los centros Research Collaboratory for Structural Bioinformatics (RCSB), European Bioinformatics Institute (EMBL-EBI) y la Universidad de Osaka en Japón, para asegurar sus propiedades.

Debido a la cantidad de entradas que existen hoy en día (131.205, Jun. 2017) y a las nuevas publicadas cada semana (200 en promedio), el proceso de gestión y mantenimiento de los datos en PDB es una carga de trabajo bastante costosa [18].

Cada estructura de macromolécula se identifica por un código de 4 caracteres alfanuméricos (PDB ID), el contenido de los archivos en PDB es información primaria y puede pertenecer a proteínas, ADN, ARN e híbridos. Por ejemplo, cada entrada referente a proteínas posee las secuencias de una o más cadenas de aminoácidos, donde se incluye la posición en coordenadas espaciales de sus átomos y de los átomos de los ligandos. Además, se incorpora información general (e.g., título y autor), y anotaciones con datos estandarizados. Un ejemplo de estas entradas se puede observar en el Anexo A. Todo el contenido en los archivos es material de gran utilidad, como evidencia en 2015 más de 500 millones de conjuntos de coordenadas fueron descargados desde wwPDB [3].

2.2. Proteínas, aminoácidos y ligandos

Puesto que la información de PDB es mayormente de proteínas, aquí se revisa qué son estas macromoléculas, sus elementos base, y su organización a distintos niveles. También se explica qué son los ligandos. Finalmente, se enseña qué distancias son útiles de las que se pueden medir entre estos componentes biológicos.

Las proteínas son herramientas moleculares que se encuentran en los seres vivos. Ellas poseen diversas funciones, entre las que destacan: catálisis de las enzimas, estructura, movimiento celular, defensa, regulación metabólica, transporte, almacenamiento de nutrientes y respuesta al estrés. Estas moléculas proteicas son formadas a partir de la unión de aminoácidos. Los aminoácidos son moléculas pequeñas que contienen un átomo de carbono central (carbono α) al que se le unen un grupo amino, un grupo carboxilo, un átomo de hidrógeno, y un grupo R o cadena lateral [15].

Las estructuras de las proteínas son extraordinariamente complejas, así que para facilitar su estudio los bioquímicos han diferenciado cuatro niveles de organización,

las que se ilustran en la Figura 2.1. La estructura primaria corresponde a la secuencia de aminoácidos. Al plegarse las cadenas de aminoácidos se forman determinadas disposiciones localizadas, estos patrones repetitivos constituyen la estructura secundaria. Las formas tridimensionales únicas que asumen las proteínas se denominan estructura terciaria. Por último, el término de estructura cuaternaria se le da a las proteínas que están formadas por varias cadenas de aminoácidos o subunidades.

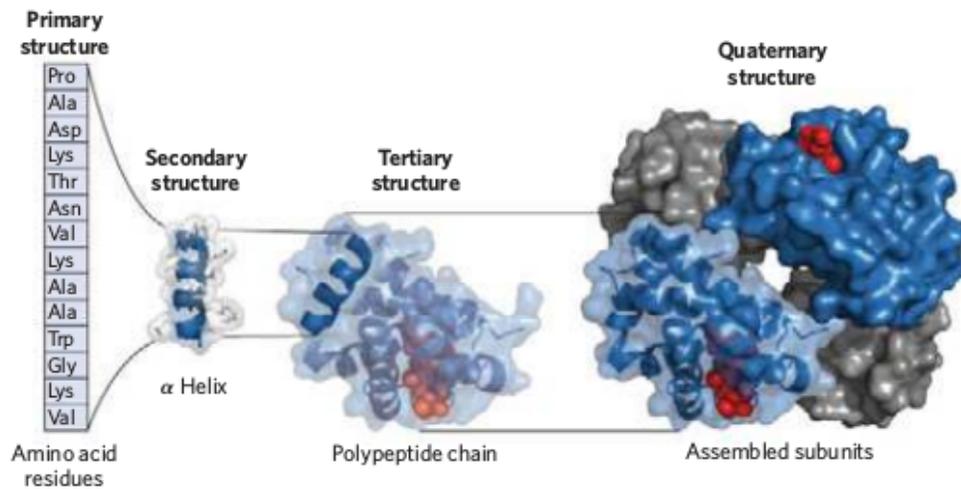


Figura 2.1: Niveles de estructura en las proteínas [17].

Los ligandos son moléculas pequeñas, iones u otros residuos que interaccionan con las proteínas. Su emparejamiento tiene la naturaleza de ser transitorio y, por ende, reversible. Estas interacciones entre proteínas y ligandos son críticas para la vida, permitiendo a los organismos responder rápidamente a cambios en las condiciones del entorno y de su metabolismo. La parte de la proteína donde se adhiere el ligando se denomina sitio de unión. Este lugar es complementario al ligando de acuerdo a características como tamaño, forma y carga, entre otras. En una misma proteína pueden existir sitios unión para muchos ligandos diferentes. Cada interacción es específica y selectiva, ya que la proteína discrimina entre miles de moléculas y sólo se empareja con una o unas pocas [17].

2.3. Herramientas de consulta para PDB

El modo de utilizar la información almacenada en PDB es mediante software especializado. En los sitios web de los miembros de la wwPDB se pueden encontrar variadas herramientas y servicios de búsqueda, exploración, visualización y análisis de datos, que han sido desarrolladas por las propias organizaciones, por instituciones de investigación o por colaboradores particulares. En general, la web alberga diversas aplicaciones dedicadas a resolver cuestiones específicas sobre el contenido de esta base de datos biológica.

Considerando el objetivo de este proyecto, que se centra en la búsqueda de patrones estructurales, se revisan algunas herramientas a modo de ejemplo, abarcando los métodos que ellas emplean para formular una consulta y sus funcionalidades.

PDBj Mine es una interfaz web para realizar búsquedas en Protein Data Bank Japan (PDBj). La interfaz, que se muestra en la Figura 2.2, permite ejecutar consultas “rápidas” al ingresar el identificador de la macromolécula o palabras claves, la secuencia de aminoácidos o el nombre del autor. Para consultas avanzadas se deben ingresar condiciones con el fin de lograr un mejor filtro. Además, se puede hacer uso de consultas a través de expresiones SQL, pero para esto se debe conocer la estructura de PDBML¹ [14].

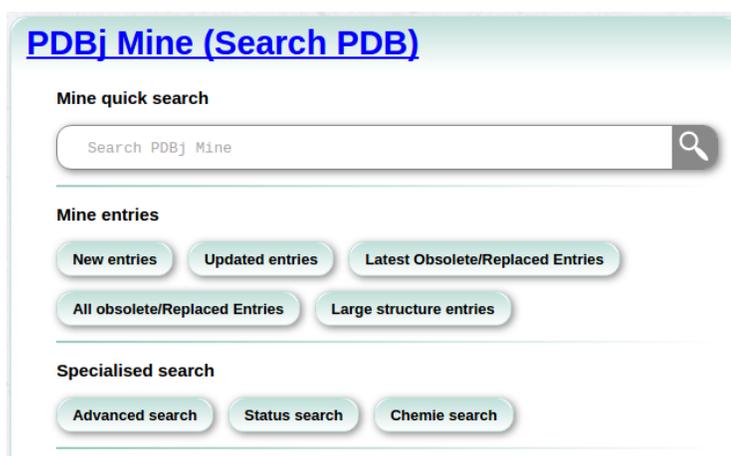


Figura 2.2: Página oficial de PDBj Mine.

Fuente: <http://pdbj.org/mine/>

¹PDBML: Formato XML canónico de PDB.

Ligand Expo es un servicio que provee información química y estructural sobre pequeñas moléculas dentro de entradas de PDB. Esta aplicación web (ver Figura 2.3) dispone de un buscador que, por medio de entradas en formularios, encuentra componentes químicos dentro del Diccionario de Componentes de PDB. También dispone de un explorador para ver tablas de componentes que contienen piezas modificadas, tales como aminoácidos, nucleótidos, drogas populares y otros [8].

ID	View Options	Description
00B	Chemical details Coordinates files	<p>Name: (4r)-5,5-dimethyl-1,3-thiazolidine-4-carboxylic acid</p> <p>Synonyms: RCSB REL</p> <p>(4r)-5,5-dimethyl-1,3-thiazolidine-4-carboxylic acid (4r)-5,5-dimethyl-1,3-thiazolidine-4-carboxylic acid</p> <p>SMILES: <chem>CC1([C@H](NCS1)C(=O)O)C</chem></p> <p>Formula: C6 H11 N O2 S</p>

Figura 2.3: Página oficial de Ligand Expo.

Fuente: <http://ligand-expo.rcsb.org/>

Protein-Protein Interactions (PPI) es una herramienta dentro de *ProteinsPlus*, un servidor con soporte para el modelado basado en estructuras. PPI mediante comparación de cadenas, que son seleccionadas a través de formularios, entrega en forma textual la clasificación de un complejo proteína-proteína en cuanto a su tipo de interacción [22]. Asimismo, indica el alineamiento de sus secuencias y permite ver la estructura en el visor NGL. Se muestra en la Figura 2.4.

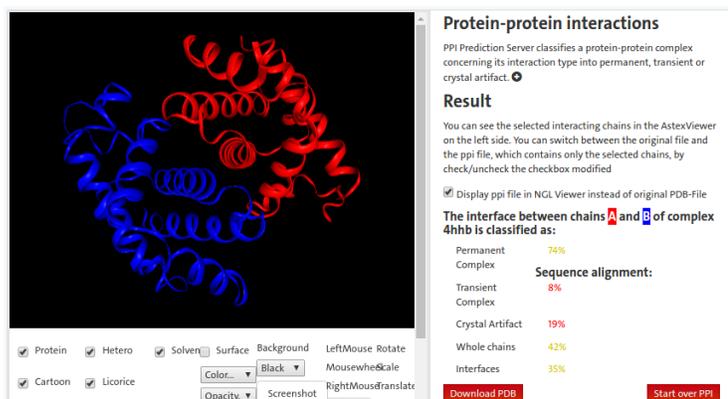


Figura 2.4: Página oficial de PPI.

Fuente: <http://poseview.zbh.uni-hamburg.de/>

2.4. Búsqueda de patrones estructurales en PDB

Tal como se hacía referencia en la introducción a PDB, el principal interés de la biología estructural es la estructura molecular de macromoléculas biológicas, cómo ellas consiguen las estructuras que poseen y cómo alteraciones en sus estructuras repercuten en sus funciones. Este asunto tiene gran relevancia para los biólogos ya que las macromoléculas llevan a cabo la mayoría de las funciones dentro de las células, y porque sólo por enrollarse en ciertas formas tridimensionales ellas son capaces de desempeñar estas funciones.

Un aspecto particularmente importante es que muchos procesos biológicos están basados en la interacción selectiva entre proteínas y ligandos, que se produce a través de un complejo emparejamiento entre ambos. Normalmente, el sitio de unión está localizado en la superficie de la proteína, pero a veces el adosamiento del ligando en la proteína objetivo conduce a la formación transitoria de un enlace intermolecular [9]. Estas interacciones proteína-ligando establecen la selectividad donde una gran cantidad de reacciones bioquímicas tienen origen.

En múltiples oportunidades un ligando específico es encontrado en diversos complejos de proteínas. En esos casos, las comparaciones de residuos presentes en los sitios de unión pueden revelar relaciones ocultas entre la naturaleza del ligando y la composición (aminoácidos) del sitio de unión. Además, si las proteínas involucradas son estructuralmente equivalentes y sus respectivas interacciones con los ligandos también lo son, entonces pueden ser comparadas. Un ejemplo de esto es Imatinib,

droga anticáncer que se une a cinasas específicas, de la cual existen varios complejos similares en PDB [9].

Debido a lo anterior, y a la conocida riqueza de información 3D almacenada en PDB, es que buscar patrones en el interior de las macromoléculas se ha convertido en un proceso popular del cual se pueden extraer y deducir una gran variedad de propiedades estructurales. Un patrón estructural puede ser cualquier parte de una macromolécula, esto es, el esqueleto de la proteína, ligandos o metales unidos o rodeando la proteína, secuencias específicas de aminoácidos o nucleótidos, y conjuntos de átomos o residuos que satisfacen criterios dados como distancias, composición o conectividad, entre otros.

Es importante mencionar que la herramienta desarrollada en este proyecto considera buscar patrones estructurales, justamente, a un nivel donde se produce la relación entre un ligando y los aminoácidos que constituyen su sitio de unión. Algo verdaderamente útil en este sentido, es conocer qué aminoácidos rodean un determinado ligando y la distancia que separa a estas moléculas.

En la Figura 2.5 se puede advertir que las distancias medidas entre un aminoácido y un ligando son las distancias extremas entre sus átomos, es decir, la distancia mínima (átomos más cercanos) y la distancia máxima (átomos más alejados). Esto es para tener datos exactos a la hora de analizar un ligando, no obstante, también es válido cuando se miden distancias entre dos aminoácidos dentro de una proteína para determinar posibles coordinadores (ver Figura 2.6).

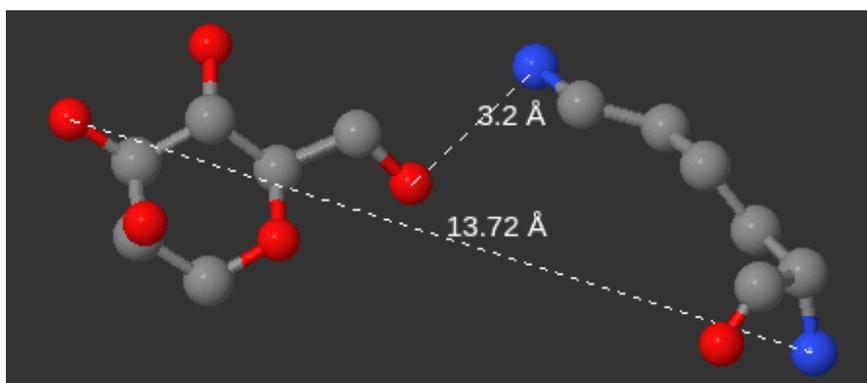


Figura 2.5: Distancias mínima y máxima entre un ligando Alfa-D-Manosa (MAN #183; a la izquierda) y un aminoácido Lisina (LYS #40; a la derecha), en la macromolécula con PDB ID 1LX5.

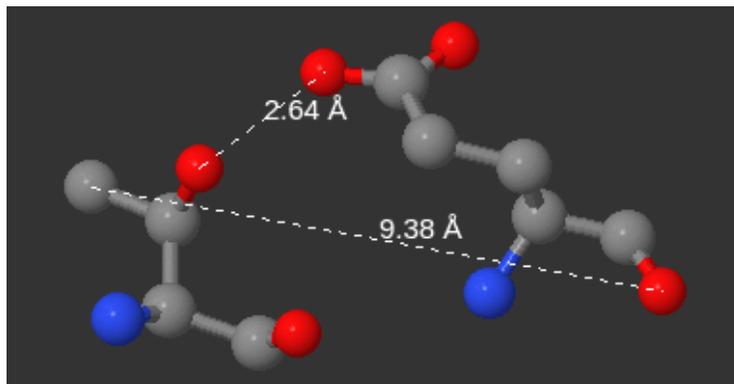


Figura 2.6: Distancias mínima y máxima entre un aminoácido Treonina (THR #41; a la izquierda) y un aminoácido ácido glutámico (GLU #43; a la derecha), en la macromolécula con PDB ID 4X3K.

En el presente, la búsqueda de patrones estructurales se lleva a cabo mediante aplicaciones que han diseñado sus propios lenguajes de consulta para definirlos. Algunas de ellas se enfocan en componentes específicos, mientras que otras permiten definir y analizar patrones bastante generales. Un ejemplo del primer tipo es **Molecular Query Language** (MQL), una gramática libre de contexto para encontrar subestructuras coincidentes. El objetivo de MQL es reconocer un patrón o estructura química (consulta) en una o más estructuras químicas diferentes. Además, posee una sintaxis fácil de leer y está implementado como una biblioteca para Java [20].

Una herramienta de propósito más general es **PatternQuery** (PQ). Esta aplicación web es usada para definir patrones estructurales en forma óptima, los que son detectados y extraídos desde PDB completo o desde un conjunto de datos determinado por el usuario. En PQ los patrones son descritos por expresiones únicas basadas en la sintaxis de Python, las cuales están diseñadas para definir patrones basados en la naturaleza y relaciones que existen entre átomos, residuos y otros elementos estructurales. Sus expresiones especifican composición, topología, conectividad y estructura 3D del patrón [23]. Un ejemplo de consulta se puede observar en la Figura 2.7.

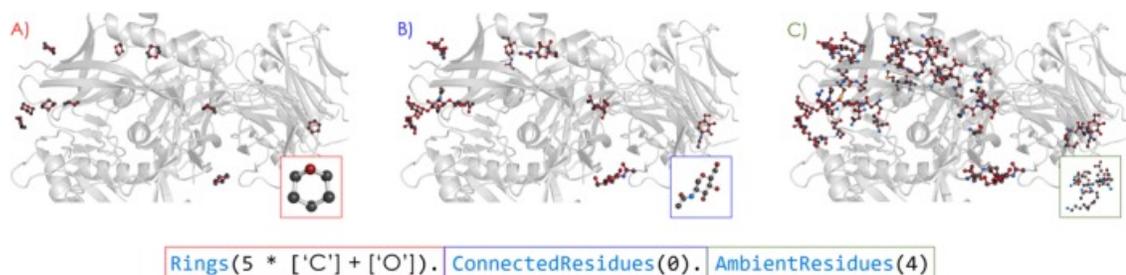


Figura 2.7: La consulta reconoce el sitio de unión de cualquier residuo que contiene Piranosa como parte de la envoltura de la glicoproteína gp160. A) Primero, la consulta identifica la Piranosa. B) Luego, son identificados todos los residuos que contienen el patrón en su estructura. C) Por último, son detectados también todos los residuos que están a lo más a 4Å desde cualquier Piranosa, para asegurar el reconocimiento de potenciales coordinadores [23].

2.5. Herramientas basadas en grafos para búsqueda de patrones en PDB

Luego de haber revisado más de cincuenta aplicaciones desarrolladas para suplir diversas necesidades en torno a búsqueda, exploración, visualización y análisis de la información de PDB (lista completa de aplicaciones en Anexo B), este proceso de estudio se redujo a herramientas que hacen uso de representaciones basadas en grafos, ya que esta estructura matemática se ajusta de manera natural a las estructuras moleculares, permitiendo así estudiar las interrelaciones entre unidades que interactúan unas con otras. Además, casi no se observaron servicios que obtengan mayor provecho de ellos.

Los grafos son altamente efectivos para representar patrones estructurales. De hecho, existe un tipo de grafo molecular muy conocido que es utilizado en química matemática. En él se representan átomos o grupos de átomos (nodos) y sus conexiones por medio de enlaces covalentes (aristas). Un ejemplo de lo anterior se puede observar en el editor **Marvin** de ChemAxon (Figura 2.8), el cual permite dibujar una estructura química y buscar tal patrón en forma de componentes exactos, similares o subestructuras de moléculas como ligandos y residuos de proteínas [7].

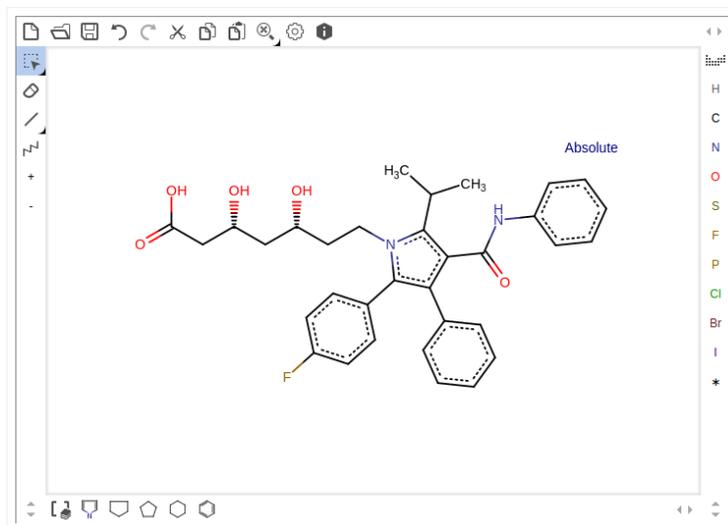


Figura 2.8: Marvin mostrando la estructura química de la Atorvastatina.

Fuente: <http://www.rcsb.org/pdb/ligand/chemAdvSearch.do>

A parte de Marvin no se logró encontrar más servicios en los que se pueda manipular una estructura basada en grafos para formar un patrón de consulta. Es por esto que a continuación se presentan algunos ejemplos de otras herramientas que utilizan grafos, principalmente, para hacer la representación visual de elementos relacionados con las macromoléculas biológicas pertenecientes a PDB.

SCOP2 (Structural Classification of Proteins 2) es un servicio enfocado a organizar proteínas estructuralmente caracterizadas, de acuerdo a sus relaciones estructurales y evolutivas [1]. El diseño de clasificación y, en consecuencia, el de visualización de la organización, se encuentran descritos en términos de grafos, en donde sus nodos forman una red compleja con múltiples relaciones que son representadas por regiones de estructuras de proteínas y secuencias.

Otra herramienta destacada que hace uso de grafos para representar información es **STRING**, una base de datos de interacciones proteína-proteína conocidas y predichas. Como se presenta en la Figura 2.9, en los resultados de búsqueda de STRING se visualiza una red que resume todas las asociaciones predichas para un grupo particular de proteínas. Aquí, los nodos son las proteínas, y las aristas simbolizan asociaciones físicas o asociaciones funcionales predichas [25].

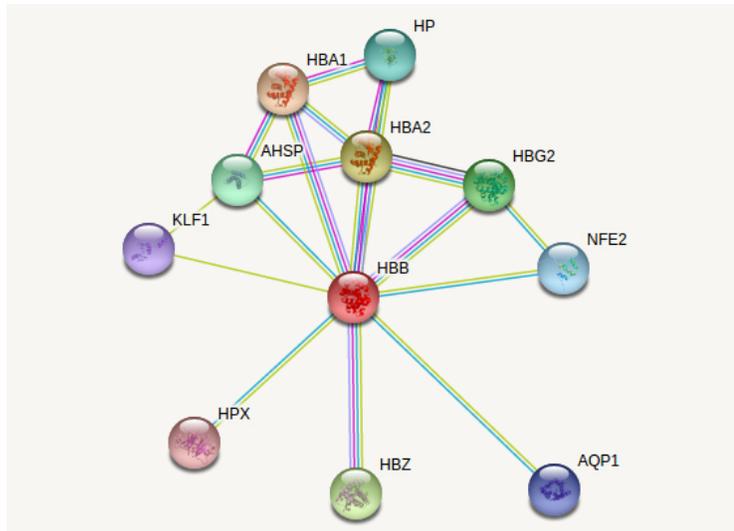


Figura 2.9: STRING mostrando la red de interacciones para Hemoglobina Beta.

Fuente: <http://string-db.org/>

Una última aplicación sobresaliente dentro de las que incorporan grafos es **Cytoscape**. Esta plataforma fue originalmente diseñada para investigación en materia biomolecular, pero ahora es posible visualizar y analizar redes complejas de diversos campos de estudio. Cytoscape es principalmente utilizada para visualizar redes de interacción de moléculas y secuencias biológicas, e integrar estas redes con anotaciones, perfiles de expresión génica y otros datos de estado que pueden ser asociados a cada nodo [24]. Es aplicable a cualquier sistema de componentes moleculares y permite la integración de nuevas características mediante extensiones.

Como se había mencionado, en lo que respecta a la investigación realizada en este trabajo, en general existen pocas aplicaciones para la búsqueda de patrones estructurales, y esta cantidad se ve en extremo reducida cuando se quiere formar un patrón con la ayuda de elementos gráficos. Ciertamente, no está disponible ningún software, con interfaz gráfica manipulable, que permita diseñar y consultar por patrones a nivel de interacción entre ligandos y aminoácidos, los que en efecto, podrían revelar relaciones ocultas responsables de muchas funciones biológicas.

3. Diseño y Construcción

Para desarrollar una interfaz gráfica que permita buscar patrones estructurales en PDB, primero es necesario especificar algunos puntos importantes que abarcan desde describir cómo se comporta la aplicación y qué funciones permite realizar hasta qué arquitectura soporta este sistema, pasando por el diseño de elementos gráficos, su interacción y su transformación a un código de consulta. Todos estos aspectos son detallados en este capítulo.

3.1. Requisitos de la aplicación

Los requisitos de esta aplicación se clasifican en dos tipos, funcionales y no funcionales. Los primeros definen las operaciones que se pueden llevar a cabo con el software, mientras que los del segundo tipo hacen referencia a atributos de calidad que sirven como criterios para juzgar la aplicación.

Todos los requisitos expuestos a continuación fueron solicitados por el profesor de bioinformática Mauricio Arenas y por Renzo Angles, escritos por Diego Cisterna, y luego revisados en conjunto.

3.1.1. Requisitos funcionales

Las funcionalidades requeridas están enmarcadas en el proceso de diseñar un patrón de interés, buscar coincidencias para ese patrón dentro de PDB y mostrar los resultados de esta búsqueda de una manera comprensible. Por tanto, se pide lo siguiente:

- Agregar componentes al patrón. Estos son: un ligando, aminoácidos y aminoácidos comodín que pueden tomar el valor de cualquiera de los veinte aminoácidos estándar.
- Agregar asociaciones entre los componentes. Estas corresponden a: conectores de distancia entre ligando y aminoácidos y entre pares de aminoácidos, y conectores siguiente entre aminoácidos para reflejar la secuencia que se da en las cadenas dentro de las proteínas.
- Incorporar propiedades. Estas son: una clasificación para los aminoácidos comodín de acuerdo a su polaridad, y establecer un rango de revisión para los conectores de tipo distancia.
- Eliminar y modificar los elementos ya nombrados.
- Buscar el patrón estructural en la base de datos de PDB.
- Mostrar los resultados de búsqueda en forma de imagen y a través de un visualizador de moléculas 3D, e incluir en ellos un enlace hacia la página web de la macromolécula.

En el Anexo C se puede revisar la especificación detallada de cada uno de los requisitos descritos anteriormente.

3.1.2. Requisitos no funcionales

Existen muchos atributos de calidad para medir un software, sin embargo, por la naturaleza de este proyecto y siguiendo cabalmente el objetivo de implementar un prototipo centrado en la funcionalidad, aquí se consideran sólo algunos de ellos. Estos son: usabilidad, correctitud, disponibilidad y mantenibilidad.

- Usabilidad es la característica más relevante en este proyecto, puesto que la interfaz gráfica para búsqueda de patrones estructurales intenta facilitar la interacción que tienen las personas con esta clase de software particular. Se busca que la aplicación sea fácil de usar y requiera de mínimo aprendizaje para el usuario, y que este último pueda alcanzar efectivamente su objetivo de forma eficiente y satisfactoria.

- Correctitud asegura que las funcionalidades de la aplicación están presentes y que cumplen su objetivo, es decir, permiten diseñar patrones estructurales, buscarlos en una base de datos y entregar resultados correctos. Además, se puede evaluar mediante un método experimental, verificando resultados de diversas pruebas.
- Disponibilidad se refiere a que el software al ser implementado para un entorno web se encuentra acondicionado para que personas de todo el mundo puedan hacer uso de él.
- Mantenibilidad asegura que la aplicación sea desarrollada con integridad conceptual y siguiendo una estructura clara, que permita localizar errores y corregirlos fácilmente, realizar cambios e incorporar nuevas funcionalidades en forma sencilla.

3.2. Arquitectura del sistema

La arquitectura utilizada es del tipo cliente-servidor de tres capas. Esta arquitectura estaba predispuesta, ya que el servidor de base de datos que tiene PDB y el servidor web que aloja aplicación son diferentes, y los computadores clientes pueden ser cualquiera sin requisitos especiales además de conexión a Internet y un navegador web con entorno gráfico.

Este tipo de arquitectura es bastante útil puesto que la separación en capas aprovecha la potencia de procesamiento de cada máquina en cada capa, además facilita el mantenimiento y la introducción de cambios o nuevas características, y por sobre todo, aísla los datos de la lógica del negocio y de la presentación de la información.

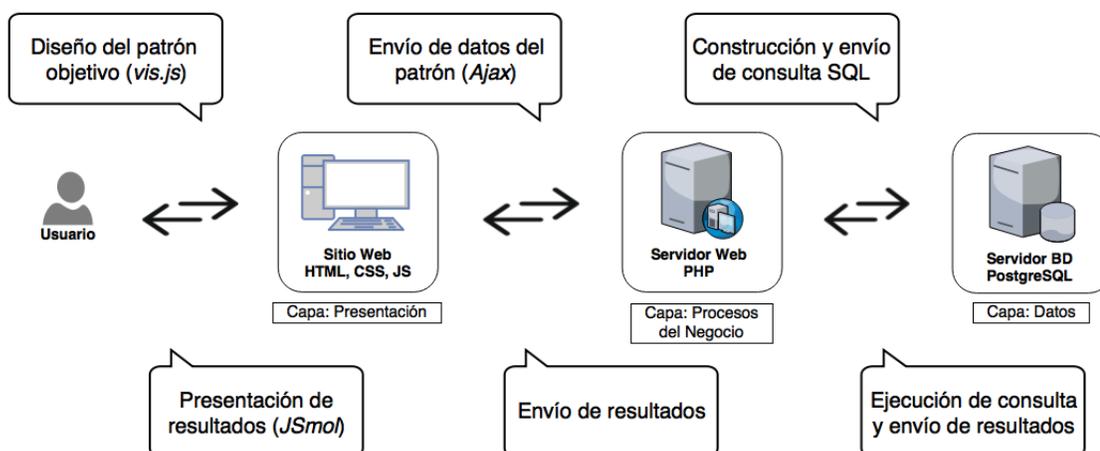


Figura 3.1: Arquitectura y proceso de datos.

Como se puede observar en el diagrama de la Figura 3.1, la capa de presentación se encarga de soportar la interacción con el usuario, permitiendo diseñar el patrón objetivo y mostrando los resultados en una forma comprensible. Aquí también se comprueba que el patrón contiene todos los elementos requeridos para iniciar el proceso de búsqueda, es decir, verifica que sea un patrón válido.

La capa de procesos del negocio recibe como entrada los datos contenidos en el patrón en forma de listas bien organizadas para facilitar la construcción de la consulta en lenguaje SQL, luego la consulta es enviada al servidor de base de datos.

Finalmente, en la capa de datos la consulta SQL es procesada y los resultados obtenidos son enviados de regreso al servidor web, y desde este último se envían hacia el computador cliente que utiliza el usuario.

Con relación a los aspectos técnicos, la capa de presentación emplea las tecnologías web HTML5, CSS3 y JavaScript con jQuery (versión 3.1.1). Para dar forma al patrón de interés se emplea una biblioteca de visualización dinámica para JavaScript llamada *vis.js* (versión 4.16.1), mientras que para la visualización tridimensional de las moléculas se hace uso de *JSmol* (versión 14.6.4). El servidor web es Debian 8 y utiliza PHP (versión 5.6.27), y el servidor de base de datos es Debian 8 y utiliza PostgreSQL (versión 9.4.5) para almacenar y consultar PDB.

3.3. Componentes y asociaciones en un patrón

Desde los requisitos funcionales se puede deducir que la herramienta debe permitir diseñar patrones estructurales como una agregación de componentes que están asociados mediante sólo dos tipos de conectores. Para diseñar un patrón correctamente es necesario entender cada componente, asociación, y su importancia para el proceso de búsqueda, todo esto se explica a continuación.

Ligando

Es el componente principal del patrón objetivo, ya que es la molécula de interés en este proceso de estudio. Se puede agregar sólo uno, y su existencia es obligatoria para que se pueda efectuar la búsqueda de coincidencias.

Aminoácidos

Corresponden a los aminoácidos fijos, es decir, representan a uno de los veinte aminoácidos estándar. Se puede agregar un número indeterminado de ellos, sin embargo entre mayor sea el número más complejo será el patrón, esto por un lado ralentiza el proceso de búsqueda pero por otro lado incrementa la precisión de los resultados.

Aminoácidos comodín

Corresponden a los aminoácidos variables (denominados ANY), es decir, ellos toman el valor de varios aminoácidos estándar. Lo anterior se acota asignando una clase a estos comodines. La clasificación va de acuerdo a la polaridad del aminoácido, existiendo cinco opciones:

- (1) Sin clasificación, no considera la polaridad, toma 20 valores.
- (2) No polar, toma 8 valores {ALA, ILE, LEU, MET, PHE, PRO, TRP, VAL}.
- (3) Polar sin carga, toma 7 valores {ASN, CYS, GLN, GLY, SER, THR, TYR}.
- (4) Polar cargado positivamente, toma 3 valores {ARG, HIS, LYS}.
- (5) Polar cargado negativamente, toma 2 valores {ASP, GLU}.

Se puede agregar un número indeterminado de estos aminoácidos, sin embargo el tiempo de búsqueda aumenta considerablemente con cada nuevo comodín en el

patrón objetivo.

La presencia de al menos un aminoácido fijo o uno comodín es obligatoria para que se pueda efectuar la búsqueda de coincidencias.

Asociación de distancia Ligando-Aminoácido

Se refiere a la asociación mediante un conector de distancia entre el ligando y un aminoácido fijo o uno comodín. Aquí se toma en cuenta la proximidad que existe entre los componentes, medida en angstroms (\AA). Al conector se le asigna un rango con valores de distancia mínima y máxima, estos límites van desde 0.0 hasta 7.0 \AA , y el valor mínimo siempre debe ser menor que el valor máximo. Este es el tipo de asociación principal y debe existir al menos una para que se pueda llevar a cabo la búsqueda del patrón en PDB.

Asociación de distancia Aminoácido-Aminoácido

Se refiere a la asociación mediante un conector de distancia entre un aminoácido fijo y un aminoácido comodín, entre dos fijos, o entre dos comodines. Al igual que en el caso anterior se considera la proximidad existente entre los componentes y las condiciones para el rango de distancia entre ellos. Este tipo de asociación es de carácter opcional.

Asociación de siguiente Aminoácido-Aminoácido

Hace referencia a la asociación mediante un conector de siguiente entre un aminoácido fijo y un aminoácido comodín, entre dos fijos, o entre dos comodines. Aquí se revisa la secuencia de aminoácidos dentro de una cadena de la proteína para extraer un par de aminoácidos consecutivos que cumplan con las propiedades de los componentes asociados. Este tipo de asociación es de carácter opcional.

Aclaración sobre las asociaciones de distancia

Las distancias consideradas para comprobar si un aminoácido está dentro del rango establecido en torno al ligando o en torno a otro aminoácido, son la distancia mínima real y la distancia máxima. La distancia mínima real se define como la distancia que existe entre los átomos más cercanos de los dos componentes involucrados en la asociación. Por su parte, la distancia máxima se define como la distancia entre

los átomos más distantes entre sí pertenecientes a los componentes, o en su defecto si parte de los átomos de los componentes distan en más de 7\AA , la distancia máxima corresponde a 7\AA . Esto se debe a que los aminoácidos que están a más de 7\AA casi no influyen en la coordinación que se da en las proteínas al recibir un ligando.

Luego, en la verificación para saber si la distancia entre los componentes está dentro del rango fijado por el usuario, las distancias antes definidas se comparan contra los límites inferior (valor mínimo) y superior (valor máximo) de ese rango.

En cuanto a las coincidencias encontradas respecto del patrón objetivo, los resultados textuales muestran la distancia mínima relativa. Esta distancia se define como la distancia mínima real, si es que esta última se encuentra dentro del rango, pero si esa distancia está por debajo del límite inferior, entonces la distancia mínima relativa corresponde justamente a este límite inferior.

3.4. Mockups y elementos de la interfaz

Las siguientes imágenes (Figuras 3.2, 3.3 y 3.4) muestran bosquejos que representan las partes principales de la interfaz gráfica de la aplicación, en cada una se explica el significado de sus elementos constituyentes.

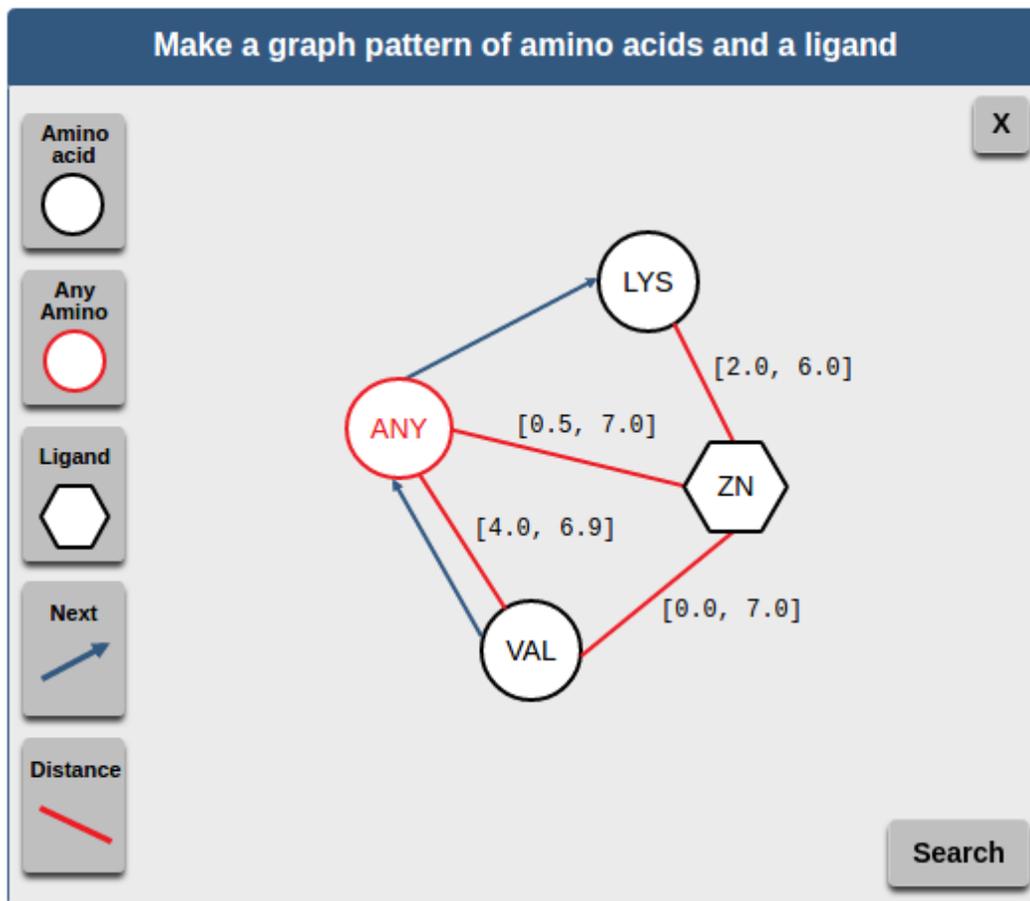


Figura 3.2: Panel donde se diseña el patrón estructural.

En la Figura 3.2, los círculos con borde color negro representan aminoácidos fijos (LYS y VAL). El círculo con borde color rojo representa un aminoácido comodín (ANY). El hexágono simboliza un ligando (ZN). Las líneas color rojo representan los conectores de distancia entre ligando-aminoácidos y aminoácido-aminoácido, las cuales están etiquetadas con sus respectivos rangos (e.g. [0.5, 7.0]). Las flechas color azul simbolizan los conectores al siguiente aminoácido en la cadena.

En el lado izquierdo se muestran cinco botones que sirven para añadir los componentes antes descritos en el panel. Al costado derecho, arriba se muestra un pequeño botón (X) que sirve para eliminar componentes del panel, y abajo aparece otro botón (Search) que sirve para iniciar la búsqueda del patrón estructural en PDB. Por último, en la parte superior se muestra el título del panel de trabajo indicando la actividad que se debe llevar a cabo.

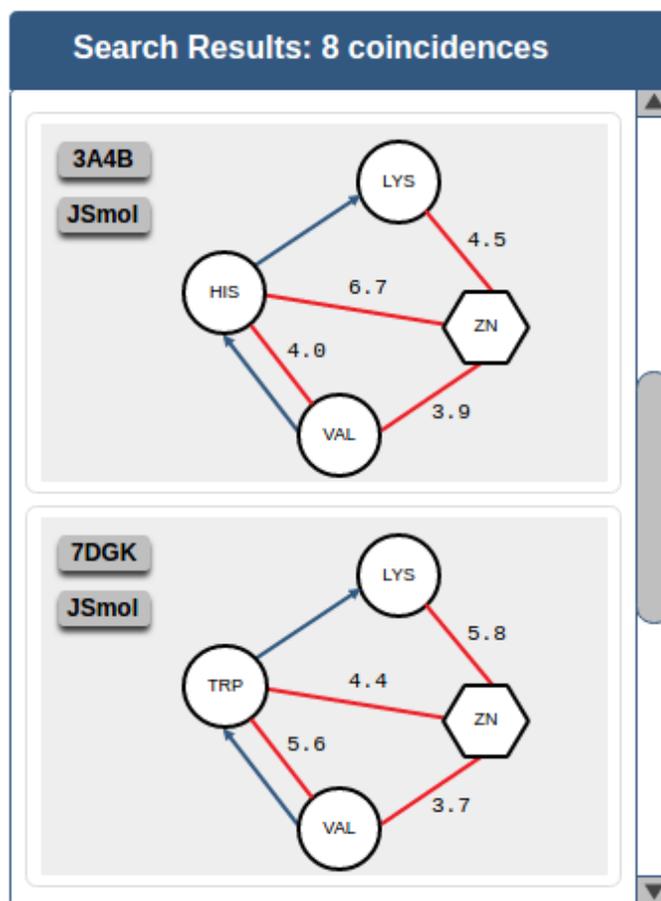


Figura 3.3: Panel donde se muestran los resultados de la búsqueda.

En la Figura 3.3, se muestra dos resultados de búsqueda. En cada uno de ellos aparece el patrón diseñado para la consulta de la Figura 3.2, y cómo éstos representan casos estructurales reales, el aminoácido comodín ha tomado el valor correspondiente en cada caso. Además, en cada resultado, en su esquina superior izquierda, se muestran dos botones, el de arriba indica el PDB ID (e.g. 3A4B) y sirve para dirigirse a la página web oficial de la macromolécula, mientras que el de abajo (JSmol) sirve para desplegar la visualización tridimensional del patrón.

El panel, en su lado derecho, incorpora una barra de desplazamiento vertical para ver cada uno de los resultados, y en su parte superior se muestra el título que indica la cantidad de coincidencias encontradas en la búsqueda.

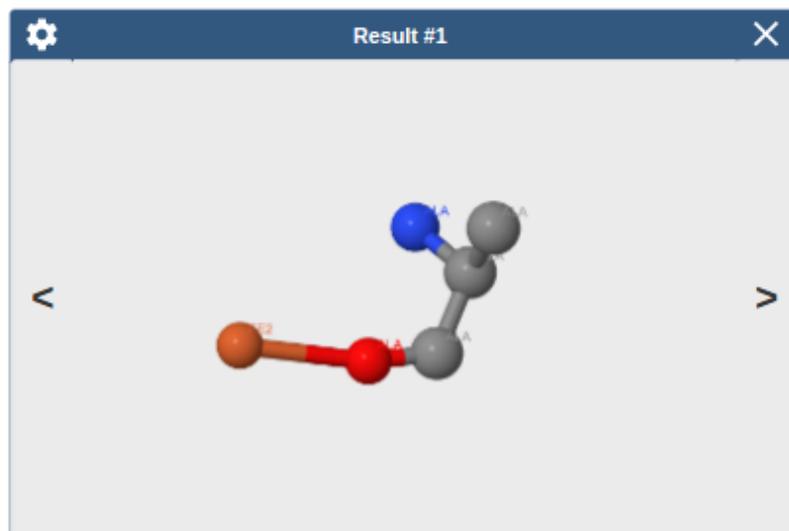


Figura 3.4: Modal donde se muestra la representación 3D de cada resultado.

En la Figura 3.4, la parte principal es el gráfico del centro porque es donde se muestra la representación tridimensional de los átomos que forman parte de los componentes, ligando y aminoácidos, que constituyen un resultado del patrón objetivo. En la parte superior, de izquierda a derecha, se exhibe un botón engranaje que sirve para desplegar las opciones del gráfico, el título que indica el número del resultado y un segundo botón “X” para cerrar el modal. Además, se incorporan a ambos costados botones para retroceder y avanzar en la visualización de los patrones coincidentes.

3.5. Interacción con la interfaz

Existen varios tipos de técnicas de interacción con los computadores y es importante conocer cuál es la tarea que se debe llevar a cabo, el tipo de dispositivo en que se utiliza el software, y la perspectiva del usuario sobre la interrelación entre la tarea y el dispositivo. La principal característica de la interfaz desarrollada es que permite manipular elementos gráficos que representan componentes de un patrón estructural. Por lo anterior, casi el cien por ciento de las funciones que realiza la aplicación se controlan por medio de clics, efectuados con el *mouse* del ordenador.

Para favorecer la dinámica dispuesta por la interfaz y, principalmente, para facilitar el trabajo de búsqueda de patrones al interior de macromoléculas, se ha escogido

un estilo de interacción denominado *Point-and-Click*. Este tipo de interacción, es más rápido y conduce a menos errores que, por ejemplo, el conocido estilo *Drag-and-Drop*, además es preferido por los usuarios [12].

Siguiendo esta técnica de apuntar y hacer clic, es que se consigue dar forma a un patrón de interés para luego buscar coincidencias. Pero en ocasiones también se realizará doble-clic para añadir atributos y arrastre de componentes para obtener una mejor distribución espacial de ellos. A continuación, se detallan los pasos necesarios para generar un patrón de búsqueda sencillo a modo de ejemplo.

En el siguiente patrón se relaciona un ligando Adenosín Trifosfato (ATP) con un aminoácido Asparagina (ASN) por medio de un conector de distancia, y este último se relaciona además con un aminoácido comodín (ANY) a través de un conector siguiente. La estructura resultante, de ejecutar los siguientes seis pasos, se muestra en la Figura 3.5.

1. En el panel de trabajo (Figura 3.2), hacer clic en el botón “Ligand”, luego en el cuadro de diálogo ingresar “ATP” confirmar la acción. Esto agrega un ligando “ATP”.
2. Hacer clic en el botón “Amino acid”, luego en el cuadro de diálogo seleccionar “Asparaginez” confirmar la acción. Esto agrega un aminoácido “ASN”.
3. Hacer clic en el botón “Any Amino” luego confirmar la acción en el cuadro de diálogo. Esto agrega un aminoácido comodín “ANY”.
4. Hacer clic en el botón “Distance” luego sobre el ligando “ATP” seguido del aminoácido “ASN”. Realizar doble-clic sobre el conector distancia que se acaba de agregar si se desea ajustar los límites de búsqueda.
5. Hacer clic en el botón “Next” luego sobre el aminoácido “ANY” seguido del aminoácido “ASN”. Esto agrega un conector siguiente entre los aminoácidos, en relación a la secuencia en la cadena.
6. Por último, hacer clic en el botón “Search” para iniciar la búsqueda del patrón estructural en PDB. Luego de unos segundos, en el panel de resultados (Figura 3.3) aparece una lista con los resultados que representan los patrones coincidentes de la búsqueda.

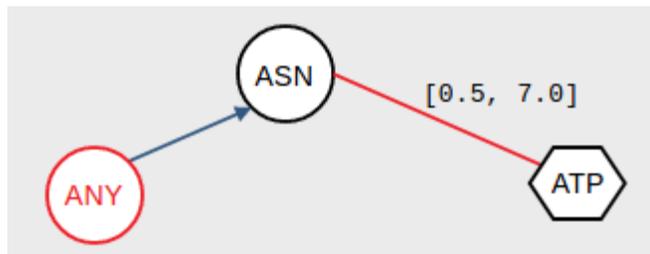


Figura 3.5: Patrón estructural resultante siguiendo los pasos del ejemplo.

3.6. Estructura de la base de datos

La base de datos original, es decir, PDB como tal es simplemente un conjunto de archivos de texto codificados en un formato particular (.pdb). Según la wwPDB, en enero de 2016 descargar PDB comprimido requería alrededor de 145GB [26]. Cada archivo contiene la información de una entrada única perteneciente a una macromolécula. No obstante, se abarcan muchos datos que permiten completitud en la información e integridad a nivel del repositorio en general. Los archivos al tener miles de líneas en un formato especial dificultan su procesamiento, por ejemplo para extraer datos particulares, realizar consultas específicas, y más aún cuando se intenta analizar la base de datos completa en busca de casos de interés dentro de las macromoléculas.

Debido a lo anterior, algunos centros de investigación han traspasado PDB a un modelo de base de datos relacional, ya que éste facilita la manipulación de sus datos. Por ejemplo, para utilizar la herramienta Mine, PDBj desarrolló Mine 2 RBD que consiste en un esquema de 400 tablas, ellos se basaron en la versión XML de PDB para este propósito [13]. Tal esquema es enorme e ineficiente cuando se trata de aplicaciones que cubren funciones puntuales, es decir, cuando se requiere sólo parte de los datos de los archivos en PDB. Por esto es que en 2015, Ana González desarrolló un esquema que se restringe a satisfacer consultas relacionadas especialmente con las distancias entre los componentes de las macromoléculas, aminoácidos y ligandos. Ella se encargó de identificar las propiedades más relevantes, haciendo especial énfasis en las que cubren lo relativo a la identificación de las estructuras y la posición espacial de sus partes elementales [10].

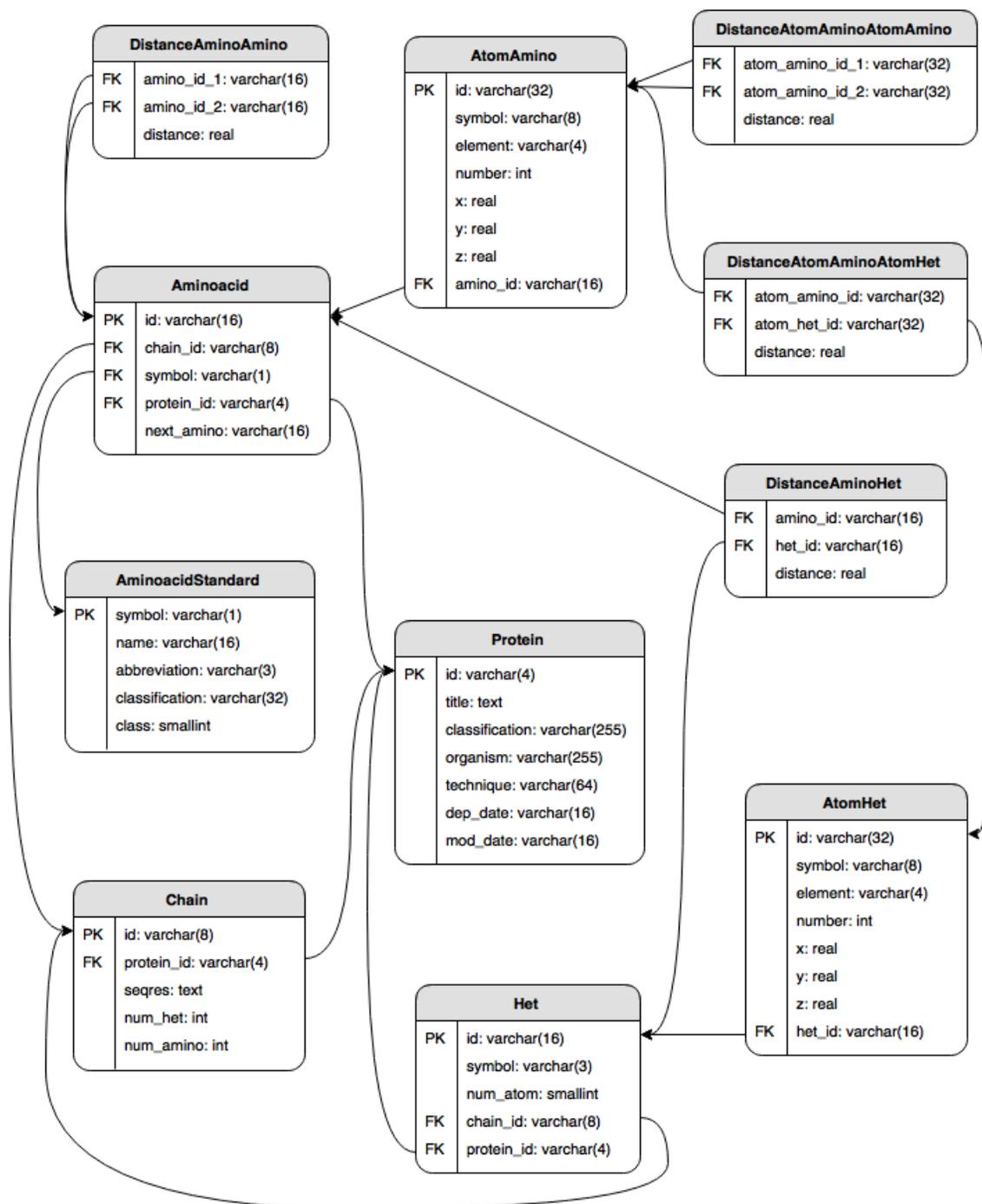


Figura 3.6: Esquema relacional para PDB.

El esquema de la Figura 3.6 es una versión mejorada del que diseñó Ana González. El esquema abarca todos los datos que se necesitan en este proyecto. Sin embargo ejecutar las consultas para buscar patrones sobre este modelo en su estado puro ocasionaría largo tiempo de procesamiento para la obtención de resultados. De hecho, la base de datos utiliza actualmente 462GB de espacio en disco. Entonces, una forma de solucionar esto es construir vistas materializadas ajustadas a cada consulta, las que se presentan en la Figura 3.7 y se describen a continuación:

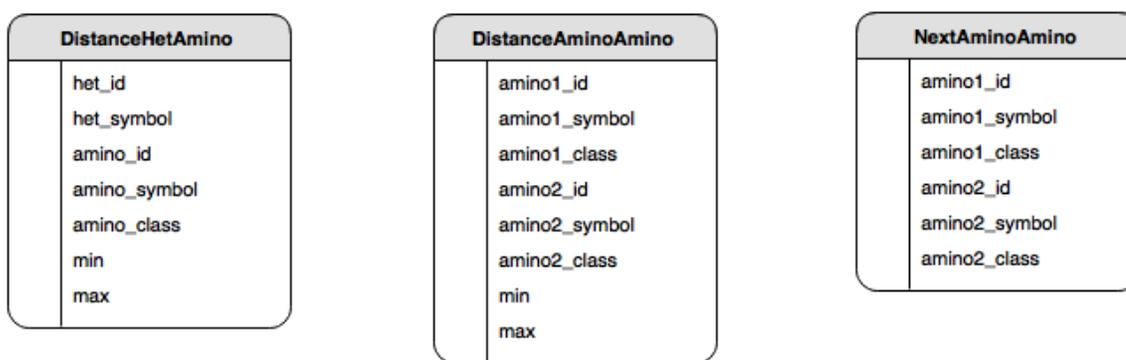


Figura 3.7: Vistas materializadas.

- **DistanceHetAmino** relaciona los datos de identificación y distancia entre ligandos y aminoácidos. Se construye a partir de la operación JOIN entre las tablas *Het*, *Aminoacid*, *AminoacidStandard*, *AtomHet*, *AtomAmino* y *DistanceAtomAminoAtomHet*.
- **DistanceAminoAmino** relaciona los datos de identificación y distancia entre aminoácidos. Se construye a partir de la operación JOIN entre las tablas *Aminoacid*, *AminoacidStandard*, *AtomAmino* y *DistanceAtomAminoAtomAmino*.
- **NextAminoAmino** relaciona los datos de identificación entre aminoácidos consecutivos en las secuencias dentro de las proteína. Se construye a partir de la operación JOIN entre las tablas *Aminoacid* y *AminoacidStandard*.

En las vistas, los atributos que comienzan con *het* y *amino* representan la información de ligandos y aminoácidos respectivamente. Los atributos *min* y *max* repre-

sentan las distancias mínimas y máximas entre los componentes que participan en cada tupla. El espacio utilizado por las vistas es de aproximadamente 27GB.

Los *scripts* SQL que permiten generar las vistas materializadas se encuentran en el Anexo D.

3.7. Confección de consultas desde el patrón gráfico

En esta sección se explica cómo es que a un patrón estructural diseñado en forma gráfica se transforma en una consulta SQL que puede ser ejecutada en una base de datos relacional. Además, se describe un caso particular donde la búsqueda de un patrón entrega resultados repetidos.

3.7.1. Asociaciones de componentes a consultas SQL

Para buscar el patrón de interés dentro de PDB primero se debe transformar cada asociación nodo-arista-nodo en una consulta en lenguaje SQL. Esto es debido a que las vistas de la base de datos construidas para responder a estas consultas relacionan de a dos componentes en cada tupla. A continuación se muestra la transformación de cada una de las asociaciones posibles entre los componentes introducidos en la Sección 3.3.

A. Ligando - conector distancia - Aminoácido¹

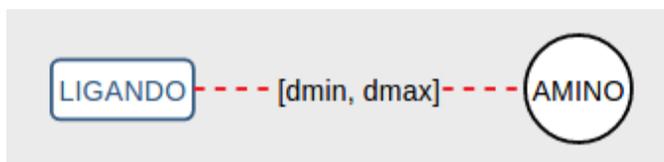


Figura 3.8: Ligando asociado a un aminoácido fijo mediante distancia.

La asociación presentada en la Figura 3.8 se traduce a la siguiente consulta:

¹Desde este punto la representación del ligando deja de ser un hexágono de borde negro y pasa a ser un rectángulo de borde azul redondeado. Además, el conector de distancia pasa de ser una línea continua a una discontinua.

```

SELECT het_id, amino_id AS amino<<id amino>>_id, amino_symbol AS amino<<id amino>>_symbol,
      min AS min_het_amino<<id amino>>
FROM distance_het_amino
WHERE het_symbol = '<<código ligando>>'
      AND amino_symbol = '<<código amino>>'
      AND ((min < <<dmin>> AND max >= <<dmin>>) OR (min <= <<dmax>> AND max > <<dmax>>))
      OR (min >= <<dmin>> AND max <= <<dmax>>) OR (min < <<dmin>> AND max > <<dmax>>));

```

En el código anterior, son entradas que provienen de los elementos gráficos:
 «id amino»: identificador numérico único y autoincremental del aminoácido.
 «código ligando»: código de tres letras identificador del ligando.
 «código amino»: código de una letra identificador del aminoácido.
 «dmin» y «dmax»: límites del rango en el conector de distancia.

B. Ligando - conector distancia - Aminoácido comodín

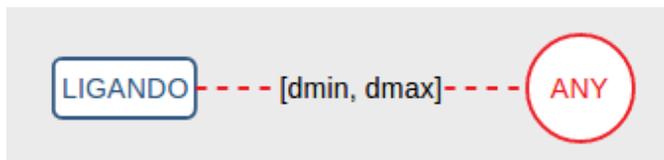


Figura 3.9: Ligando asociado a un aminoácido comodín mediante distancia.

La asociación presentada en la Figura 3.9 se traduce a la siguiente consulta:

```

SELECT het_id, amino_id AS amino<<id any>>_id, amino_symbol AS amino<<id any>>_symbol,
      min AS min_het_amino<<id any>>
FROM distance_het_amino
WHERE het_symbol = '<<código ligando>>'
      AND ((min < <<dmin>> AND max >= <<dmin>>) OR (min <= <<dmax>> AND max > <<dmax>>))
      OR (min >= <<dmin>> AND max <= <<dmax>>) OR (min < <<dmin>> AND max > <<dmax>>));

```

En el código anterior, son entradas que provienen de los elementos gráficos:
 «id any»: identificador numérico único y autoincremental del aminoácido any.
 «código ligando»: código de tres letras identificador del ligando.
 «dmin» y «dmax»: límites del rango en el conector de distancia.

C. Aminoácido - conector distancia - Aminoácido

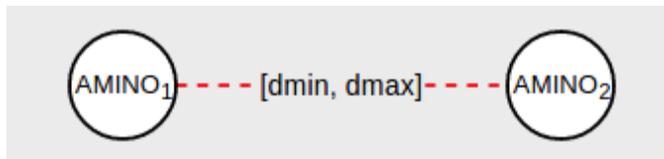


Figura 3.10: Dos aminoácidos fijos asociados mediante distancia.

La asociación presentada en la Figura 3.10 se traduce a la siguiente consulta:

```

SELECT * FROM (
  SELECT amino1_id AS amino<<id amino 1>>_id, amino1_symbol AS amino<<id amino 1>>_symbol,
         amino2_id AS amino<<id amino 2>>_id, amino2_symbol AS amino<<id amino 2>>_symbol,
         min AS min_amino<<id amino 1>>_amino<<id amino 2>>
  FROM distance_amino_amino
  WHERE ((min < <<dmin>> AND max >= <<dmin>>) OR (min <= <<dmax>> AND max > <<dmax>>))
         OR (min >= <<dmin>> AND max <= <<dmax>>) OR (min < <<dmin>> AND max > <<dmax>>))
         AND amino1_symbol = '<<código amino 1>>'
         AND amino2_symbol = '<<código amino 2>>'
  UNION
  SELECT amino2_id AS amino<<id amino 1>>_id, amino2_symbol AS amino<<id amino 1>>_symbol,
         amino1_id AS amino<<id amino 2>>_id, amino1_symbol AS amino<<id amino 2>>_symbol,
         min AS min_amino<<id amino 1>>_amino<<id amino 2>>
  FROM distance_amino_amino
  WHERE ((min < <<dmin>> AND max >= <<dmin>>) OR (min <= <<dmax>> AND max > <<dmax>>))
         OR (min >= <<dmin>> AND max <= <<dmax>>) OR (min < <<dmin>> AND max > <<dmax>>))
         AND amino2_symbol = '<<id amino 1>>'
         AND amino1_symbol = '<<id amino 2>>'
) AS q_<<id amino 1>>_<<id amino 2>>

```

En el código anterior, son entradas que provienen de los elementos gráficos: «id amino 1» y «id amino 2»: identificadores numéricos únicos y autoincrementales de los aminoácidos.

«código amino 1» y «código amino 2»: códigos de una letra identificadores de los aminoácidos.

«dmin» y «dmax»: límites del rango en el conector de distancia.

D. Aminoácido - conector distancia - Aminoácido comodín

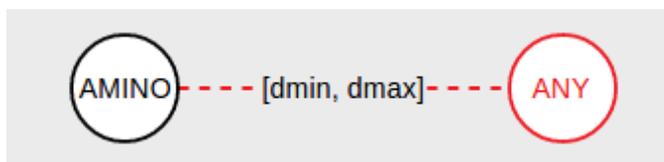


Figura 3.11: Aminoácido fijo asociado a un aminoácido comodín mediante distancia.

La asociación presentada en la Figura 3.11 se traduce a la siguiente consulta:

```
SELECT * FROM (
  SELECT amino1_id AS amino<<id amino>>_id, amino1_symbol AS amino<<id amino>>_symbol,
    amino2_id AS amino<<id any>>_id, amino2_symbol AS amino<<id any>>_symbol,
    min AS min_amino<<id amino>>_amino<<id any>>
  FROM distance_amino_amino
  WHERE ((min < <<dmin>> AND max >= <<dmin>>) OR (min <= <<dmax>> AND max > <<dmax>>))
    OR (min >= <<dmin>> AND max <= <<dmax>>) OR (min < <<dmin>> AND max > <<dmax>>))
    AND amino1_symbol = '<<código amino>>'
```

```
UNION
  SELECT amino2_id AS amino<<id amino>>_id, amino2_symbol AS amino<<id amino>>_symbol,
    amino1_id AS amino<<id any>>_id, amino1_symbol AS amino<<id any>>_symbol,
    min AS min_amino<<id amino>>_amino<<id any>>
  FROM distance_amino_amino
  WHERE ((min < <<dmin>> AND max >= <<dmin>>) OR (min <= <<dmax>> AND max > <<dmax>>))
    OR (min >= <<dmin>> AND max <= <<dmax>>) OR (min < <<dmin>> AND max > <<dmax>>))
    AND amino2_symbol = '<<código amino>>'
```

```
) AS q_<<id amino>>_<<id any>>
```

En el código anterior, son entradas que provienen de los elementos gráficos: «id amino» y «id any»: identificadores numéricos únicos y autoincrementales de los aminoácidos.

«código amino»: código de una letra identificador del aminoácido.

«dmin» y «dmax»: límites del rango en el conector de distancia.

E. Aminoácido comodín - conector distancia - Aminoácido comodín

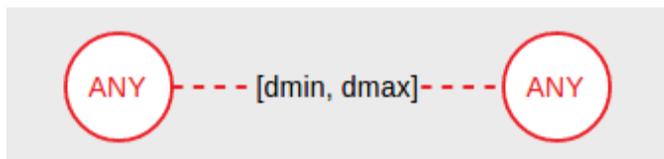


Figura 3.12: Dos aminoácidos comodín asociados mediante distancia.

La asociación presentada en la Figura 3.12 se traduce a la siguiente consulta:

```
SELECT * FROM (
  SELECT amino1_id AS amino<<id any 1>>_id, amino1_symbol AS amino<<id any 1>>_symbol,
         amino2_id AS amino<<id any 2>>_id, amino2_symbol AS amino<<id any 2>>_symbol,
         min AS min_amino<<id any 1>>_amino<<id any 2>>
  FROM distance_amino_amino
  WHERE ((min < <<dmin>> AND max >= <<dmin>>) OR (min <= <<dmax>> AND max > <<dmax>>)
         OR (min >= <<dmin>> AND max <= <<dmax>>) OR (min < <<dmin>> AND max > <<dmax>>))
 UNION
  SELECT amino2_id AS amino<<id any 1>>_id, amino2_symbol AS amino<<id any 1>>_symbol,
         amino1_id AS amino<<id any 2>>_id, amino1_symbol AS amino<<id any 2>>_symbol,
         min AS min_amino<<id any 1>>_amino<<id any 2>>
  FROM distance_amino_amino
  WHERE ((min < <<dmin>> AND max >= <<dmin>>) OR (min <= <<dmax>> AND max > <<dmax>>)
         OR (min >= <<dmin>> AND max <= <<dmax>>) OR (min < <<dmin>> AND max > <<dmax>>))
) AS q_<<id amino>>_<<id any>>
```

En el código anterior, son entradas que provienen de los elementos gráficos: «id any 1» y «id any 2»: identificadores numéricos únicos y autoincrementales de los aminoácidos.

«dmin» y «dmax»: límites del rango en el conector de distancia.

F. Aminoácido - conector siguiente - Aminoácido



Figura 3.13: Dos aminoácidos fijos asociados mediante secuencia.

La asociación presentada en la Figura 3.13 se traduce a la siguiente consulta:

```
SELECT amino1_id AS amino<<id amino 1>>_id, amino1_symbol AS amino<<id amino 1>>_symbol,
       amino2_id AS amino<<id amino 2>>_id, amino2_symbol AS amino<<id amino 2>>_symbol
FROM next_amino_amino
WHERE amino1_symbol = '<<código amino 1>>'
      AND amino2_symbol = '<<código amino 2>>';
```

En el código anterior, son entradas que provienen de los elementos gráficos:
 «id amino 1» y «id amino 2»: identificadores numéricos únicos y autoincrementales de los aminoácidos.
 «código amino 1» y «código amino 2»: códigos de una letra identificadores de los aminoácidos.

G. Aminoácido - conector siguiente - Aminoácido comodín



Figura 3.14: Aminoácido fijo asociado a un aminoácido comodín mediante secuencia.

La asociación presentada en la Figura 3.14 se traduce a la siguiente consulta:

```
SELECT amino1_id AS amino<<id amino>>_id, amino1_symbol AS amino<<id amino>>_symbol,
       amino2_id AS amino<<id any>>_id, amino2_symbol AS amino<<id any>>_symbol
FROM next_amino_amino
WHERE amino1_symbol = '<<código amino>>';
```

En el código anterior, son entradas que provienen de los elementos gráficos:
 «id amino» y «id any»: identificadores numéricos únicos y autoincrementales de los aminoácidos.
 «código amino»: código de una letra identificador del aminoácido.

H. Aminoácido comodín - conector siguiente - Aminoácido



Figura 3.15: Aminoácido comodín asociado a un aminoácido fijo mediante secuencia.

La asociación presentada en la Figura 3.15 se traduce a la siguiente consulta:

```
SELECT amino1_id AS amino<<id any>>_id, amino1_symbol AS amino<<id any>>_symbol,
       amino2_id AS amino<<id amino>>_id, amino2_symbol AS amino<<id amino>>_symbol
FROM next_amino_amino
WHERE amino2_symbol = '<<código amino>>';
```

En el código anterior, son entradas que provienen de los elementos gráficos:
«id any» y «id amino»: identificadores numéricos únicos y autoincrementales de los aminoácidos.

«código amino»: código de una letra identificador del aminoácido.

I. Aminoácido comodín - conector siguiente - Aminoácido comodín



Figura 3.16: Dos aminoácidos comodín asociados mediante secuencia.

La asociación presentada en la Figura 3.16 se traduce a la siguiente consulta:

```
SELECT amino1_id AS amino<<id any 1>>_id, amino1_symbol AS amino<<id any 1>>_symbol,
       amino2_id AS amino<<id any 2>>_id, amino2_symbol AS amino<<id any 2>>_symbol
FROM next_amino_amino
```

En el código anterior, son entradas que provienen de los elementos gráficos:
«id any 1» y «id any 2»: identificadores numéricos únicos y autoincrementales de los aminoácidos.

3.7.2. Definición inductiva de patrones y ejemplo de transformación

Siguiendo el método de inducción estructural es posible demostrar que un patrón está siempre bien formado si en su construcción se emplean las asociaciones presentadas en la Subsección anterior. Para realizar la definición inductiva es necesario que el conjunto de todos los patrones válidos se pueda construir a partir de un conjunto de elementos base y de un conjunto finito de reglas de construcción.

El conjunto de elementos base corresponde a los dos patrones más simples:

- Ligando - *conector distancia* - Aminoácido
- Ligando - *conector distancia* - Aminoácido comodín

Las reglas de construcción corresponden a las siguientes:

1. Agregar aminoácido mediante *conector distancia*
2. Agregar aminoácido comodín mediante *conector distancia*
3. Crear nueva asociación mediante *conector distancia*
4. Agregar aminoácido mediante *conector siguiente*
5. Agregar aminoácido comodín mediante *conector siguiente*
6. Crear nueva asociación mediante *conector siguiente*

Si el patrón estructural se ve en términos de grafos los aminoácidos fijos, aminoácidos comodines y ligandos serían sus vértices, y los *conectores distancia* y *siguiente* serían sus aristas.

Cabe destacar que todas las reglas anteriores aplican desde uno de los vértices del patrón más sencillo, convirtiéndolo en uno más complejo. Las reglas 1 y 2 se pueden aplicar a cualquier patrón, un vértice soporta una cantidad indeterminada de *conectores distancia*. La regla 3 se puede aplicar sólo cuando no existe un *conector distancia* entre dos vértices. Las reglas 4 y 5 se pueden aplicar exclusivamente en vértices aminoácidos fijos o comodines, además como el tipo de *conector siguiente* posee dirección, este conector puede ser usado como llegada o salida de un aminoácido. La regla 6 se puede aplicar sólo cuando un vértice aminoácido fijo o comodín

carece de *conector siguiente* de llegada, salida o ambos, es decir, un vértice soporta únicamente dos *conectores siguiente*, uno de llegada y uno de salida.

Teniendo en cuenta lo anterior, se puede afirmar que el patrón estructural de la Figura 3.17 está diseñado correctamente. En él se hacen presente las asociaciones A, D y H de la Subsección 3.7.1, por tanto para su transformación a código SQL se utilizan precisamente los códigos mostrados en esos puntos. No obstante, aquí se incorpora una condición adicional que no se incluyó anteriormente por tener carácter de opcional. Esto se refiere a añadir una clasificación en el aminoácido comodín, específicamente que pertenezca al grupo de los aminoácidos polares con carga negativa, modificando levemente los códigos de las asociaciones D y H.

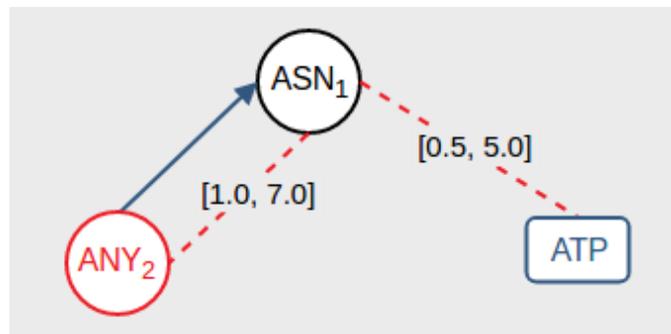


Figura 3.17: Patrón de ejemplo para transformación.

Como es apreciable en la consulta mostrada a continuación, la técnica para reunir los resultados de cada asociación independiente es utilizar la operación NATURAL JOIN. Esta técnica es factible ya que siempre debe existir al menos un camino para llegar desde un vértice hacia todos los demás. Cada vértice, con excepción del ligando, es renombrado haciendo uso de los identificadores únicos que se asignan en la interfaz gráfica. Al final se incluye una cláusula WHERE, la razón de su existencia se da a conocer en la siguiente subsección.

```

SELECT * FROM (
  SELECT * FROM (
    SELECT het_id, amino_id AS amino1_id, amino_symbol AS amino1_symbol, min AS min_het_amino1
    FROM distance_het_amino
    WHERE het_symbol = 'ATP' AND amino_symbol = 'N'
      AND ((min < 0.5 AND max >= 0.5) OR (min <= 5.0 AND max > 5.0)
      OR (min >= 0.5 AND max <= 5.0) OR (min < 0.5 AND max > 5.0))
  ) AS q0 NATURAL JOIN (
    SELECT * FROM (
      SELECT amino1_id AS amino2_id, amino1_symbol AS amino2_symbol, amino2_id AS amino1_id,
      amino2_symbol AS amino1_symbol, min AS min_amino2_amino1
      FROM distance_amino_amino
      WHERE ((min < 1.0 AND max >= 1.0) OR (min <= 7.0 AND max > 7.0)
      OR (min >= 1.0 AND max <= 7.0) OR (min < 1.0 AND max > 7.0))
      AND amino1_class = 4 AND amino2_symbol = 'N'
    UNION
      SELECT amino2_id AS amino2_id, amino2_symbol AS amino2_symbol, amino1_id AS amino1_id,
      amino1_symbol AS amino1_symbol, min AS min_amino2_amino1
      FROM distance_amino_amino
      WHERE ((min < 1.0 AND max >= 1.0) OR (min <= 7.0 AND max > 7.0)
      OR (min >= 1.0 AND max <= 7.0) OR (min < 1.0 AND max > 7.0))
      AND amino2_class = 4 AND amino1_symbol = 'N'
    ) AS q_21
  ) AS q1 NATURAL JOIN (
    SELECT amino1_id AS amino2_id, amino1_symbol AS amino2_symbol, amino2_id AS amino1_id,
    amino2_symbol AS amino1_symbol
    FROM next_amino_amino
    WHERE amino1_class = 4 AND amino2_symbol = 'N'
  ) AS q2
) AS query
WHERE amino1_id <> amino2_id;

```

3.7.3. Resultados repetidos para un patrón diseñado correctamente

Esto ocurre cuando en el patrón se incluyen dos o más aminoácidos, sean estos aminoácidos fijos o aminoácidos comodín, asociados por conectores de distancia ya sea con el ligando o entre ellos. Para entender esto más fácilmente se considera el siguiente patrón de ejemplo (Figura 3.18):

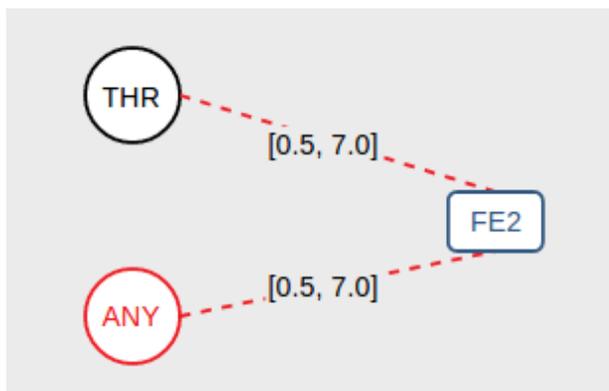


Figura 3.18: Patrón estructural con resultados repetidos.

Sin perder generalidad, se considera que todos los conectores poseen el mismo rango de distancia entre el ligando y cada uno de los aminoácidos. Además, se debe tener claro que ANY puede tomar el valor de cualquier aminoácido que alcance a entrar en ese rango. Por lo tanto, como THR busca coincidir con una Treonina bien podría representar a THR #294 que se encuentra a una distancia de 5 amstrongs de FE2 #6 en la macromolécula con PDBID 1E08. ANY por su parte toma el valor de todos los aminoácidos que entran en el rango $[0.5, 7.0]$, por consiguiente éste también representaría en una oportunidad a THR #294 al mismo tiempo que THR. Lo que provocaría un resultado donde los dos aminoácidos son exactamente el mismo. Sucede algo similar si se toman en cuenta los casos presentados en la Figura 3.19.

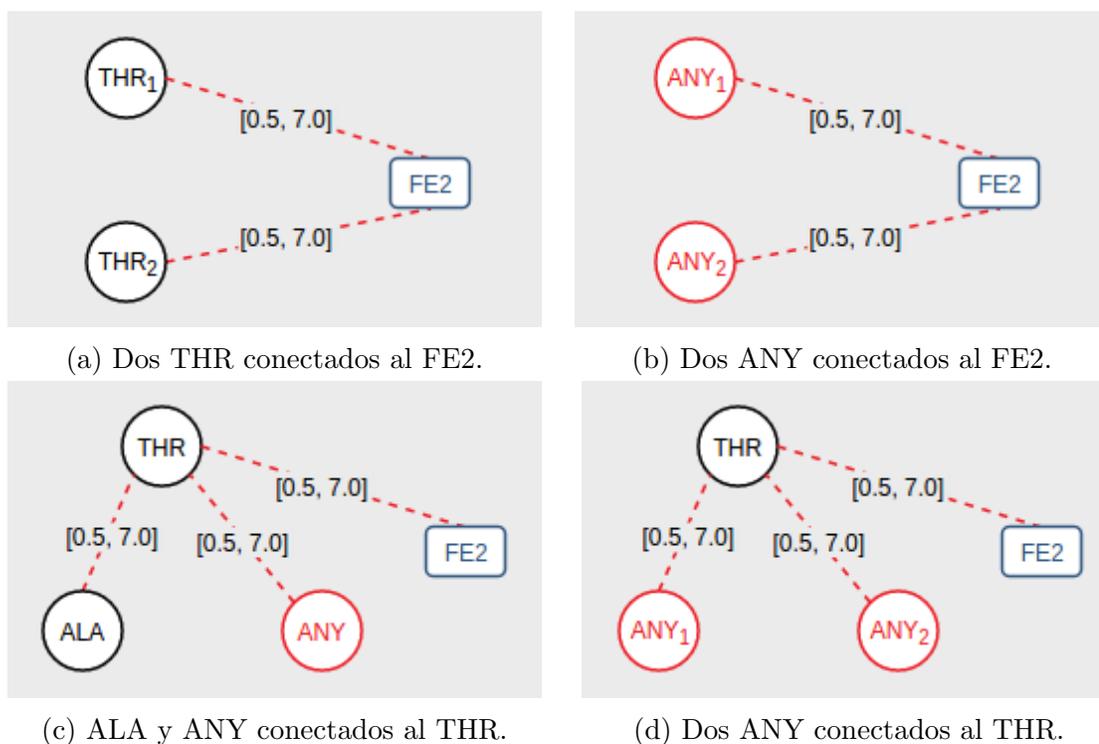


Figura 3.19: Ejemplos de casos donde se repiten los aminoácidos en el resultado.

En términos de probabilidad, un patrón estructural que incluye únicamente aminoácidos del tipo ANY siempre genera más resultados repetidos que un patrón donde se incluyen aminoácidos fijos. La probabilidad también varía en función de los rangos de distancia en los conectores, ya que si estos son iguales existe una mayor posibilidad de obtener resultados repetidos, no obstante entre mayor sea la diferencia entre ellos la probabilidad de ocurrencia es menor, llegando a cero cuando no existe intersección entre todos los rangos definidos.

Puesto que estos resultados no poseen utilidad práctica, se ha incluido una restricción que determina que sólo existan aminoácidos diferentes en una coincidencia encontrada, es decir, los aminoácidos de un resultado son todos diferentes entre sí. Entonces, en el patrón de la Figura 3.18, si THR representa a THR #294, obligatoriamente ANY tendría que representar a otro aminoácido cercano al ligando FE2 #6 y viceversa.

3.8. Prototipo final

De manera similar a lo expuesto en la Sección 3.4, a continuación se presentan los componentes principales de la interfaz gráfica construida para la aplicación web, explicando cada uno de los elementos² que los constituyen. Capturas adicionales de la interfaz se pueden ver en el Anexo E.

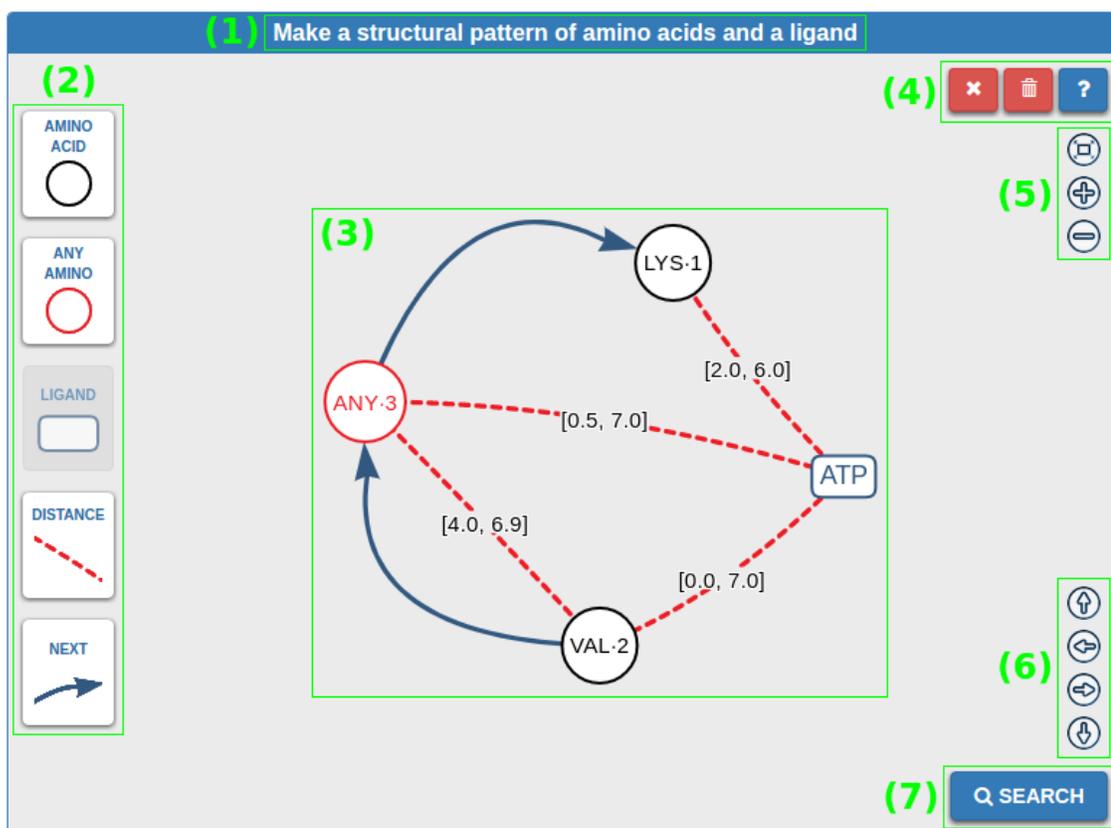


Figura 3.20: Panel donde se diseña el patrón estructural.

En la Figura 3.20 se puede observar: (1) Título del panel; (2) Botones para añadir componentes al patrón estructural; (3) Zona del patrón estructural; (4) Botones rojos sirven para eliminar componentes, aquí se ha agregado un botón (basurero) para limpiar el panel. Se ha agregado también el botón azul para obtener ayuda sobre las funciones de cada botón en el panel; (5) Nuevos botones para controlar el zoom, de

²Los elementos gráficos que tienen relación entre sí están encerrados en áreas rectangulares con bordes de color verde, cada una es representada por un dígito entre paréntesis.

arriba hacia abajo, botón para centrar, acercar, y alejar; (6) Nuevos botones para mover el patrón en cuatro direcciones; (7) Botón (SEARCH) para iniciar la búsqueda del patrón estructural en PDB.

The screenshot displays a search results interface with the following elements:

- (1)** Search results: 48 coincidencias
- (2)** Format of the Results
- (3)** PDB ID: 3RTG
- (4)** RESULT #1
- (5)** 3D STRUCTURE
- (6)** LIGAND-AMINO distances
 - [ATP]{ATP #494}-----2.8----- (LYS·1){LYS #402}
 - [ATP]{ATP #494}-----6.0----- (ANY·3){LEU #401}
 - [ATP]{ATP #494}-----6.6----- (VAL·2){VAL #400}
- (7)** AMINO-AMINO distances
 - (ANY·3){LEU #401}-----4.0----- (VAL·2){VAL #400}
- (8)** AMINO-AMINO chain sequence
 - (VAL·2){VAL #400}-----NEXT-----> (ANY·3){LEU #401}
 - (ANY·3){LEU #401}-----NEXT-----> (LYS·1){LYS #402}
- (9)** PDB ID: 3RTE
- RESULT #2
- 3D STRUCTURE
- Navigation: < PREV, 1 - 48, NEXT >

Figura 3.21: Panel donde se muestran los resultados de la búsqueda.

En la Figura 3.21 se puede observar: (1) Título del panel, indica la cantidad de coincidencias encontradas para el patrón objetivo; (2) Nuevo botón para desplegar el formato de los resultados a fin de que sean bien interpretados; (3) Botón PDB ID para dirigirse a la página de la macromolécula del resultado; (4) Título con número del resultado; (5) Botón (3D STRUCTURE) para mostrar la representación tridimensional del resultado; (6) Asociaciones de distancia entre el ligando y los aminoácidos conectados a él; (7) Asociaciones de distancia entre aminoácidos interconectados; (8) Asociaciones de siguiente en la cadena de aminoácidos para aquellos que están en

secuencia; (9) Barra con botones (PREV y NEXT) para controlar la paginación de resultados, además se incluye al centro el intervalo de los resultados mostrados.

Notar que el cambio más relevante entre el mockup de la Figura 3.3 y el panel de la figura anterior, es que cada coincidencia del patrón estructural ya no se muestra en una imagen sino que se expresa a través de una representación textual equivalente. Esto se debe a que la edición requerida por la imagen del patrón, para incluirle los valores pertenecientes a cada resultado, es un proceso complejo que hubiera implicado tener que invertir mucho tiempo extra en desarrollo. Por esta razón se ha optado por una alternativa más sencilla que no deja de satisfacer la necesidad de mostrar los resultados de forma comprensible.

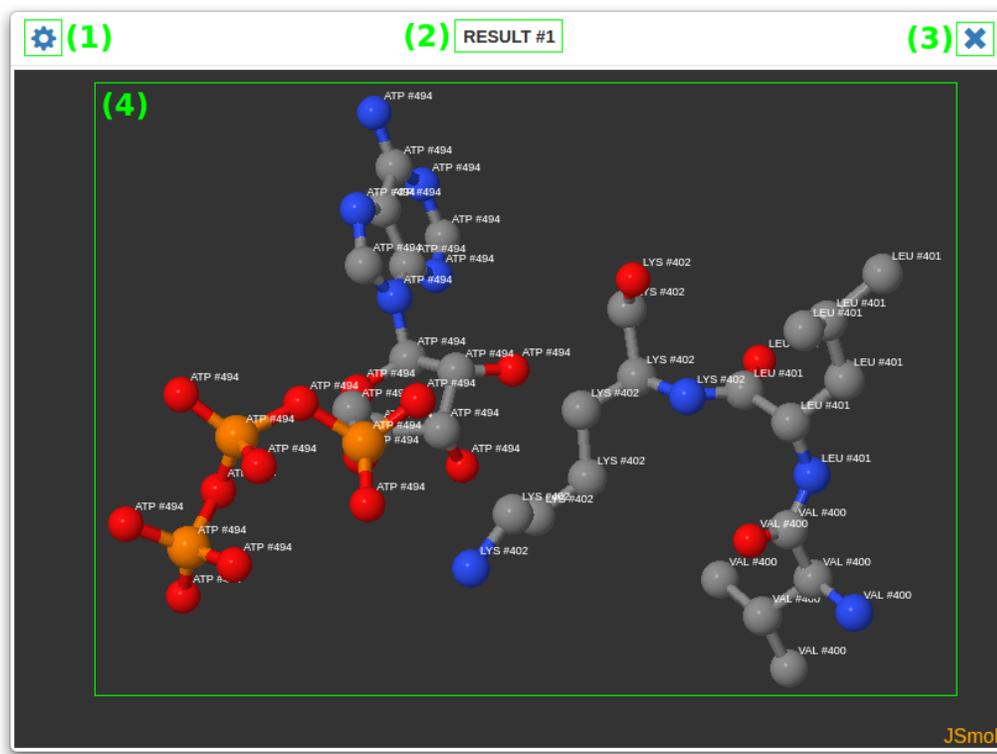


Figura 3.22: Modal donde se muestra la representación 3D de cada resultado.

En la Figura 3.22 se puede observar: (1) Botón para desplegar las opciones del visor de moléculas; (2) Título con número del resultado; (3) Botón para cerrar el modal; (4) Zona con la representación tridimensional del resultado, usando el modelo de barras y esferas. Se ha descartado la incorporación de botones laterales para

retroceder y avanzar en la visualización de los patrones coincidentes, porque no contribuirían de forma significativa al control de este proceso.

3.9. Optimización de la base de datos

Aunque la base de datos ya poseía un alto nivel de optimización mediante las vistas materializadas, igualmente se realizaron algunas pruebas para comprobar si se podía acelerar el proceso de búsqueda con el uso de índices. Se tomó un subconjunto de 889 macromoléculas para crear una base de datos de prueba, todas ellas comparten una característica en común, y es que todas contienen el ligando Adenosín Trifosfato (ATP).

La comparación se llevó a cabo ejecutando tres consultas que fueron extraídas de los patrones estructurales que se muestran en las Figuras 3.23, 3.24 y 3.25. Cada patrón tiene diferente complejidad, la que es determinada por su cantidad de nodos y aristas. En el experimento se midió el tiempo de procesamiento implicado al hacer las consultas sobre las vistas en su estado puro, y luego de haber agregado índices B-tree en cada una. Cabe destacar que sólo se agregaron índices a las columnas de los atributos que almacenan los códigos de los componentes, tanto del ligando como de los aminoácidos estándar. Lo anterior significa que en la vista *DistanceHetAmino* se agregaron índices a *het_symbol* y a *amino_symbol*, y en las vistas *DistanceAminoAmino* y *NextAminoAmino* se agregaron a *amino1_symbol* y a *amino2_symbol*.

Se ha escogido utilizar índices B-tree ya que por experiencias anteriores ellos han mostrado un mejor desempeño que, por ejemplo, los índices Hash, tanto en el incremento de velocidad para ejecutar una consulta como en ser más ligeros en su creación. Además, en la documentación de PostgreSQL se desaconseja el uso de índices Hash [11].

A continuación se muestran los resultados de las pruebas realizadas:

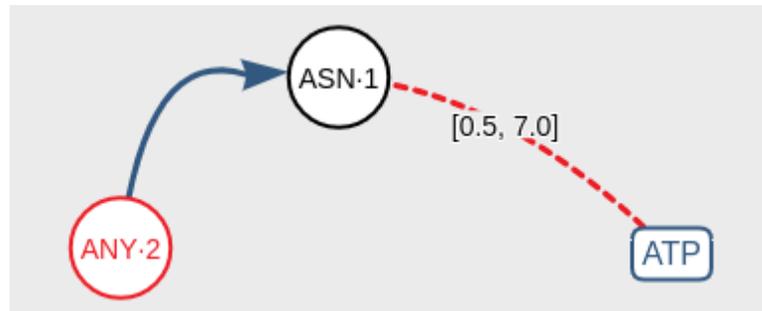


Figura 3.23: Patrón de prueba número uno.

El tiempo de procesamiento para el patrón de tres nodos y dos aristas (Figura 3.23), sin índices fue de 335 milisegundos y con índices fue de 42 milisegundos. Por lo tanto, el tiempo disminuyó en un 87,46 %.

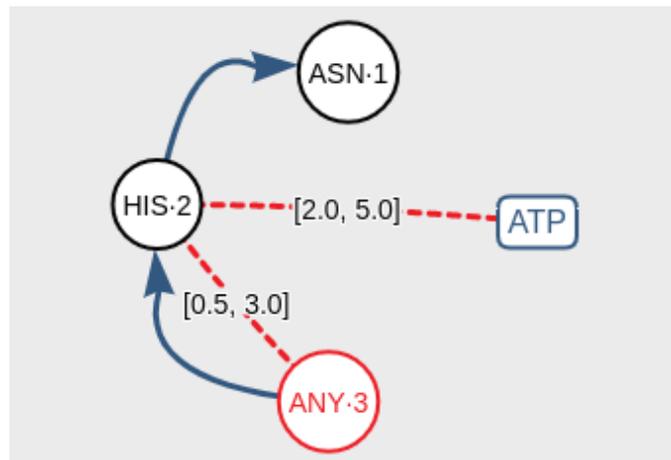


Figura 3.24: Patrón de prueba número dos.

El tiempo de procesamiento para el patrón de cuatro nodos y cuatro aristas (Figura 3.24), sin índices fue de 3,6 segundos y con índices fue de 395 milisegundos. Por lo tanto, el tiempo disminuyó en un 89,03 %.

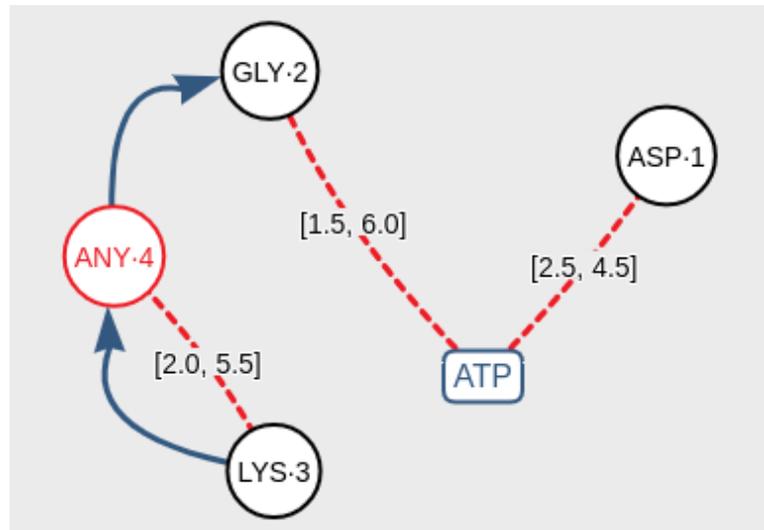


Figura 3.25: Patrón de prueba número tres.

El tiempo de procesamiento para el patrón de cinco vértices y cinco aristas (Figura 3.25), sin índices fue de 1024 segundos y con índices fue de 16 segundos. Por lo tanto, el tiempo disminuyó en un 98,44 %.

En los tres casos el tiempo de procesamiento invertido en la búsqueda de coincidencias del patrón en PDB se redujo considerablemente. Los *scripts* SQL correspondientes a cada consulta de prueba se pueden observar en el Anexo F.

4. Verificación, Validación y Evaluación

Debido a la naturaleza de este proyecto, que se centra esencialmente en un desarrollo de software, es que se deben efectuar algunas etapas en paralelo y posteriores a la construcción para determinar el nivel de completitud y calidad del producto resultante. Por consiguiente, en este capítulo se expone el proceso de verificación y validación de sus funcionalidades, y la evaluación de usabilidad.

4.1. Verificación de funcionalidades

La verificación de software es el proceso donde se comprueba la correspondencia entre el software que se ha construido y la especificación de sus funcionalidades. En otras palabras, se responde a la pregunta ¿Estamos construyendo el producto correctamente? [6].

Una forma sencilla para realizar el proceso de verificación es revisar si las funcionalidades implementadas satisfacen, tanto los criterios de aceptación de cada uno de los requisitos funcionales, como las definiciones de los requisitos no funcionales.

A continuación se define el porcentaje de aceptación para cada uno de los requisitos funcionales de la aplicación:

- **Requisito:** Agregar ligando. **Aceptación:** 100 %
Prueba: Se puede agregar un único ligando al patrón estructural, haciendo clic en un botón (LIGAND) destinado para ese propósito.

- **Requisito:** Agregar aminoácidos. **Aceptación:** 100 %
Prueba: Se pueden agregar aminoácidos estándar al patrón estructural, haciendo clic en un botón (AMINO ACID) destinado para ese propósito.
- **Requisito:** Agregar aminoácidos comodín. **Aceptación:** 100 %
Prueba: Se pueden agregar aminoácidos comodín al patrón estructural, haciendo clic en un botón (ANY AMINO) destinado para ese propósito.
- **Requisito:** Identificar ligando y aminoácidos. **Aceptación:** 100 %
Prueba: Se puede identificar el ligando introduciendo su código de tres letras en una caja de texto destinada para ese fin. Se pueden identificar los aminoácidos seleccionando su nombre desde una lista desplegable, luego el nombre es transformado a su abreviatura de tres letras.
- **Requisito:** Agregar conector distancia. **Aceptación:** 100 %
Prueba: Se pueden agregar conectores de distancia entre ligando-aminoácido y aminoácido-aminoácido, haciendo clic en un botón (DISTANCE) destinado para ese objetivo, luego haciendo clic sobre los componentes implicados.
- **Requisito:** Agregar conector siguiente. **Aceptación:** 100 %
Prueba: Se pueden agregar conectores siguiente entre pares de aminoácidos, haciendo clic en un botón (NEXT) destinado para ese objetivo, luego haciendo clic sobre los componentes implicados.
- **Requisito:** Incorporar propiedades. **Aceptación:** 100 %
Prueba: Se puede incorporar una clasificación en los aminoácidos comodín seleccionándola desde una lista desplegable. Se puede modificar el rango de distancia en ese tipo de conectores, introduciendo los valores en cajas de texto destinadas para ese fin.
- **Requisito:** Eliminar componentes. **Aceptación:** 100 %
Prueba: Se pueden eliminar componentes del patrón estructural; por separado, haciendo clic en un botón (x) destinado para ese fin y luego sobre el componente a eliminar; todos al mismo tiempo, haciendo clic en un botón (basurero) destinado para ese propósito.

- **Requisito:** Buscar patrón en PDB. **Aceptación:** 100 %
Prueba: Se puede iniciar el proceso de búsqueda del patrón estructural haciendo clic en un botón (SEARCH) destinado para ese propósito. El patrón gráfico es transformado a una consulta SQL, luego ésta es ejecutada sobre PDB.
- **Requisito:** Mostrar resultados de búsqueda. **Aceptación:** 90 %
Prueba: Se muestran los resultados de la búsqueda en forma de listado. Se puede avanzar y retroceder entre resultados cuando existen muchos de ellos (más de cien), haciendo clic en botones (PREV y NEXT) destinados para ese fin.
- **Requisito:** Enlazar resultados a PDB. **Aceptación:** 100 %
Prueba: Se puede dirigir a la página de la macromolécula donde se encontró el patrón objetivo, haciendo clic en un botón (PDB ID) destinado para ese fin que se encuentra en cada resultado.
- **Requisito:** Visualizar resultados en JSmol. **Aceptación:** 100 %
Prueba: Se puede ver una representación tridimensional de cada resultado en el visor JSmol, haciendo clic en un botón (3D STRUCTURE) destinado para ese objetivo que se encuentra en cada resultado.

A continuación se define el porcentaje de cumplimiento para cada uno de los requisitos no funcionales de la aplicación:

- **Requisito:** Usabilidad. **Cumplimiento:** 87 %
Detalles: Este requisito se revisa en extenso más adelante en la Sección 4.3. Por lo pronto, se puede señalar que tras llevar a cabo una evaluación de usabilidad aplicando un método cuantitativo y uno cualitativo, el prototipo fue bien valorado. En números, se alcanzó una nota de 4,35 en una escala con tope en 5. En palabras, el software cumple bastante bien su objetivo, es fácil de usar y permite una interacción fluida.
- **Requisito:** Correctitud. **Cumplimiento:** 98 %
Detalles: Tal como se explica en la Sección 4.2, lo más delicado en cuanto al prototipo es su proceso de búsqueda de coincidencias para los patrones estructurales. En este sentido, se logró comprobar que el software es capaz de encontrar todas las apariciones para un patrón dado y que cada resultado de

manera independiente posee valores correctos en sus asociaciones. Todo lo anterior fue conseguido por medio de pruebas experimentales ejecutadas en forma manual.

- **Requisito:** Disponibilidad. **Cumplimiento:** 95 %

Detalles: El prototipo desarrollado es accesible para todo el mundo a través de la dirección web <https://structuralbio.utralca.cl/gsp4pdb>. Los servidores que mantienen la aplicación en línea están activos la mayor parte del tiempo, bajando el sistema sólo cuando se producen cortes en el suministro de corriente. No obstante están configurados para reactivarse automáticamente en cuanto se restablece la corriente.

- **Requisito:** Mantenibilidad. **Cumplimiento:** 100 %

Detalles: Los archivos del proyecto están en directorios separados según su tipo (HTML, CSS, JS, PHP, etc.). Esto sigue la organización estructural recomendada para aplicaciones HTML5. El código fue desarrollado empleando integridad conceptual, por ejemplo, en nombres de variables y firma de funciones. Igualmente, se incluyeron muchos comentarios aclaradores y separadores de código. El proyecto se encuentra respaldado en la plataforma de desarrollo GitHub, para manejar diferentes versiones y para compartirlo fácilmente.

4.2. Validación de funcionalidades

La validación de software es el proceso donde se evalúa el sistema o un componente durante o al final de la etapa de construcción para determinar si las funcionalidades satisfacen los requisitos en forma correcta. Informalmente, se contesta a la pregunta ¿Estamos construyendo el producto correcto? [6].

En este caso como se trata de un software pequeño, el proceso de validación está relacionado completamente con la búsqueda del patrón estructural dentro de PDB. Por lo anterior, se realizaron diversas pruebas durante la implementación de las funcionalidades, lo que llevó a descubrir, entre otras cosas, el caso de los resultados repetidos para un patrón correctamente diseñado.

A continuación se presenta un método de prueba manual, para evidenciar que al buscar un patrón utilizando la herramienta desarrollada se obtienen los mismos resultados que por medio de consultas SQL simples y directas. Se considera el siguiente

patrón objetivo para ejecutar el método (ver Figura 4.1):

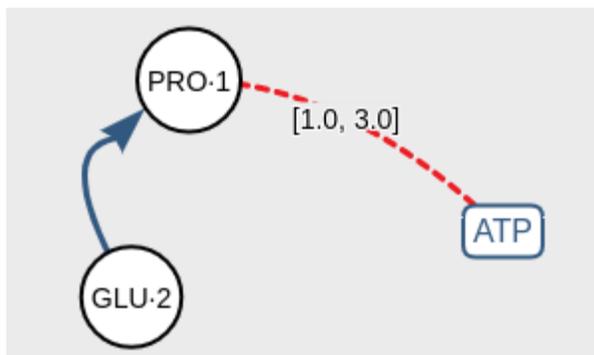


Figura 4.1: Patrón objetivo para ejecutar el método de prueba manual.

Primero, se ejecuta una consulta para obtener todos los ligando ATP, resultan ser 925. Se debe aclarar que estas pruebas son realizadas sobre la base de datos de prueba (889 macromoléculas) mencionada en la Sección 3.9.

```
SELECT id, protein_id FROM het WHERE symbol = 'ATP'
```

Segundo, se ejecuta una consulta para obtener todos los aminoácidos Prolina (PRO; P), resultan ser 14.235.

```
SELECT id, protein_id FROM aminoacid WHERE symbol = 'P'
```

Tercero, se ejecuta una consulta donde se reúnen los resultados anteriores junto a los átomos de los ATP y de las Prolinas, para poder hacer JOIN con la tabla que almacena las distancias entre átomos, se obtienen 22.750 tuplas. Desde aquí en adelante cada nueva consulta incluye la anterior, con excepción del sexto paso.

```

SELECT atp.id AS atp_id, proline.id AS proline_id, distanceatomaminoatomhet.distance
FROM
  (SELECT id, protein_id FROM het WHERE symbol = 'ATP') AS atp
  JOIN (SELECT id, protein_id FROM aminoacid WHERE symbol = 'P') AS proline
    ON proline.protein_id = atp.protein_id
  JOIN atomhet ON atomhet.het_id = atp.id
  JOIN atomamino ON atomamino.amino_id = proline.id
  JOIN distanceatomaminoatomhet ON distanceatomaminoatomhet.atom_het_id = atomhet.id
    AND distanceatomaminoatomhet.atom_amino_id = atomamino.id

```

Cuarto, se ejecuta una consulta para obtener las distancias mínima y máxima que existe entre los ligandos ATP y los aminoácidos Prolinas, se consiguen 1.005 tuplas.

```

SELECT atp_id, proline_id, min(distance), max(distance)
FROM (
  SELECT atp.id AS atp_id, proline.id AS proline_id, distanceatomaminoatomhet.distance
  FROM
    (SELECT id, protein_id FROM het WHERE symbol = 'ATP') AS atp
    JOIN (SELECT id, protein_id FROM aminoacid WHERE symbol = 'P') AS proline
      ON proline.protein_id = atp.protein_id
    JOIN atomhet ON atomhet.het_id = atp.id
    JOIN atomamino ON atomamino.amino_id = proline.id
    JOIN distanceatomaminoatomhet ON distanceatomaminoatomhet.atom_het_id = atomhet.id
      AND distanceatomaminoatomhet.atom_amino_id = atomamino.id
  ) AS sub1
GROUP BY sub1.atp_id, sub1.proline_id

```

Quinto, se ejecuta una consulta para extraer las tuplas que satisfacen la condición de distancia del rango [1.0, 3.0], se obtienen 31 tuplas.

```

SELECT atp_id, proline_id, min AS min_distance, max AS max_distance
FROM (
  SELECT atp_id, proline_id, min(distance), max(distance)
  FROM (
    SELECT atp.id AS atp_id, proline.id AS proline_id, distanceatomaminoatomhet.distance
    FROM
      (SELECT id, protein_id FROM het WHERE symbol = 'ATP') AS atp
      JOIN (SELECT id, protein_id FROM aminoacid WHERE symbol = 'P') AS proline
        ON proline.protein_id = atp.protein_id
      JOIN atomhet ON atomhet.het_id = atp.id
      JOIN atomamino ON atomamino.amino_id = proline.id
      JOIN distanceatomaminoatomhet ON distanceatomaminoatomhet.atom_het_id = atomhet.id
        AND distanceatomaminoatomhet.atom_amino_id = atomamino.id
    ) AS sub1
  GROUP BY sub1.atp_id, sub1.proline_id
) AS sub2
WHERE ((min < 1.0 AND max >= 1.0) OR (min <= 3.0 AND max > 3.0)
OR (min >= 1.0 AND max <= 3.0) OR (min < 1.0 AND max > 3.0))

```

Sexto, se ejecuta una consulta para obtener todos los aminoácidos Ácido Glutámico (GLU; E), resultan ser 24.020.

```

SELECT id, next_amino FROM aminoacid WHERE symbol = 'E'

```

Séptimo, se ejecuta la consulta definitiva, aquí se reúnen los resultados del quinto y sexto paso para obtener las tuplas que satisfacen la condición de secuencia entre los aminoácidos GLU y PRO, se consiguen 4 tuplas.

```

SELECT atp_id, proline_id, glutamicacid.id AS glutamicacid_id, min_distance, max_distance
FROM (
  SELECT atp_id, proline_id, min AS min_distance, max AS max_distance
  FROM (
    SELECT atp_id, proline_id, min(distance), max(distance)
    FROM (
      SELECT atp.id AS atp_id, proline.id AS proline_id, distanceatomaminoatomhet.distance
      FROM
        (SELECT id, protein_id FROM het WHERE symbol = 'ATP') AS atp
        JOIN (SELECT id, protein_id FROM aminoacid WHERE symbol = 'P') AS proline
          ON proline.protein_id = atp.protein_id
        JOIN atomhet ON atomhet.het_id = atp.id
        JOIN atomamino ON atomamino.amino_id = proline.id
        JOIN distanceatomaminoatomhet ON distanceatomaminoatomhet.atom_het_id = atomhet.id
          AND distanceatomaminoatomhet.atom_amino_id = atomamino.id
    ) AS sub1
    GROUP BY sub1.atp_id, sub1.proline_id
  ) AS sub2
  WHERE ((min < 1.0 AND max >= 1.0) OR (min <= 3.0 AND max > 3.0)
    OR (min >= 1.0 AND max <= 3.0) OR (min < 1.0 AND max > 3.0))
) AS sub3
JOIN (SELECT id, next_amino FROM aminoacid WHERE symbol = 'E') AS glutamicacid
  ON glutamicacid.next_amino = sub3.proline_id

```

	atp_id character varying(16)	proline_id character varying(16)	glutamicacid_id character varying(16)	min_distance real	max_distance real
1	3LMI_A_1760	3LMI_A_714	3LMI_A_713	2.908	6.885
2	4B0S_A_1501	4B0S_A_230	4B0S_A_229	2.446	6.991
3	4WAE_A_801	4WAE_A_612	4WAE_A_611	2.792	6.945
4	4ZS4_A_903	4ZS4_A_714	4ZS4_A_713	2.953	6.879

Figura 4.2: Resultados obtenidos al ejecutar el método de prueba manual.

Haciendo uso de la herramienta desarrollada se obtienen exactamente los mismos resultados (incluso en el mismo orden). Esto se puede observar al comparar las filas de la Figura 4.2 y los cuadros de la Figura 4.3.

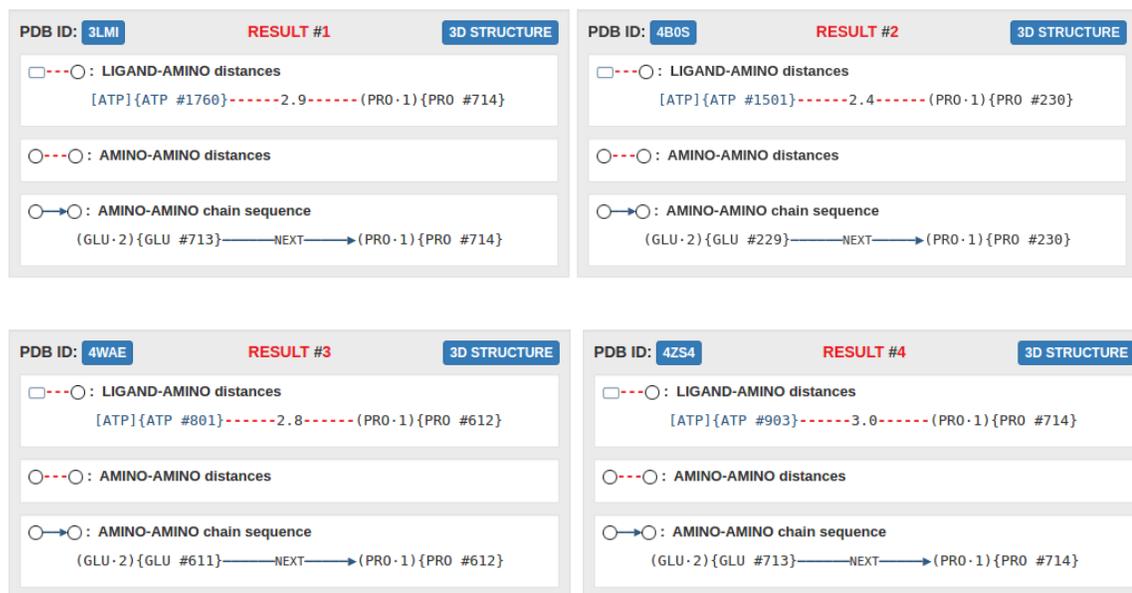


Figura 4.3: Resultados obtenidos de la búsqueda del patrón en la herramienta.

Por otro lado, también es pertinente revisar algunos resultados para probar que cada uno de ellos posee valores correctos. Existe un método sencillo para saber si las distancias y las secuencias de aminoácidos entregadas en los resultados de la Figura 4.3 son correctas, y es por medio de la representación tridimensional del resultado. El visor JSmol hace uso de los archivos originales en formato PDB para graficar, por ese motivo es posible comparar los resultados textuales con lo que exhiben los gráficos 3D.

Las siguientes figuras muestran las vistas tridimensionales de los resultados #1 y #4 de la Figura 4.3. En el primer caso, la distancia que arroja la herramienta corresponde a 2.9Å mientras que en la Figura 4.4 se puede ver que la distancia es de 2.91Å, por lo tanto el resultado es correcto. En el segundo caso sucede algo similar, ya que la herramienta determina que la distancia es 3.0Å mientras que en la Figura 4.5 se puede ver que la distancia es de 2.95Å, así que también es un resultado correcto. Notar que en la pérdida de precisión la distancia se aproxima hacia el decimal más cercano. Por último, en ambos casos es posible observar que existe un enlace entre los aminoácidos involucrados, lo que evidencia su secuencia dentro de la cadena.

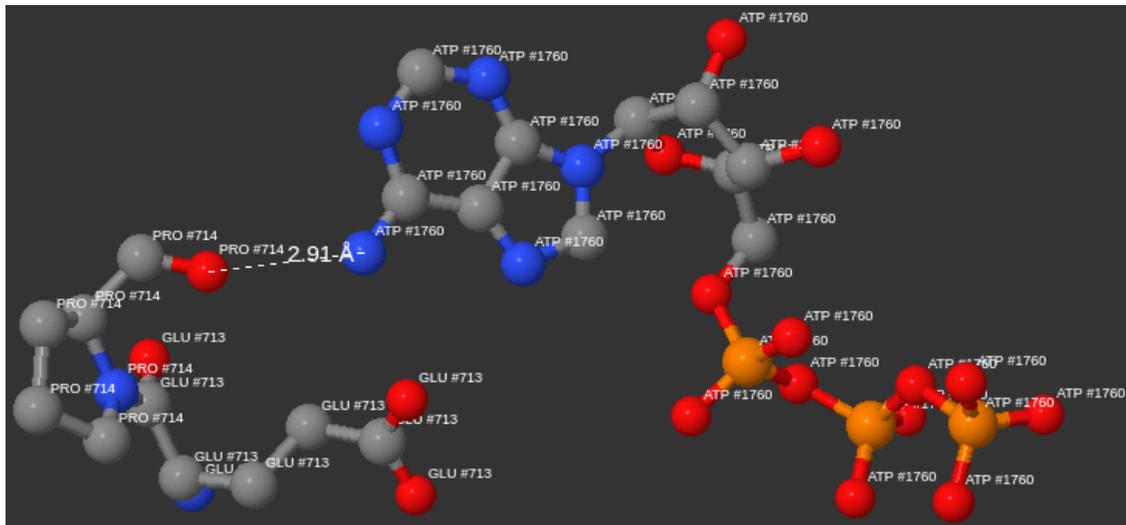


Figura 4.4: Vista tridimensional del resultado #1.

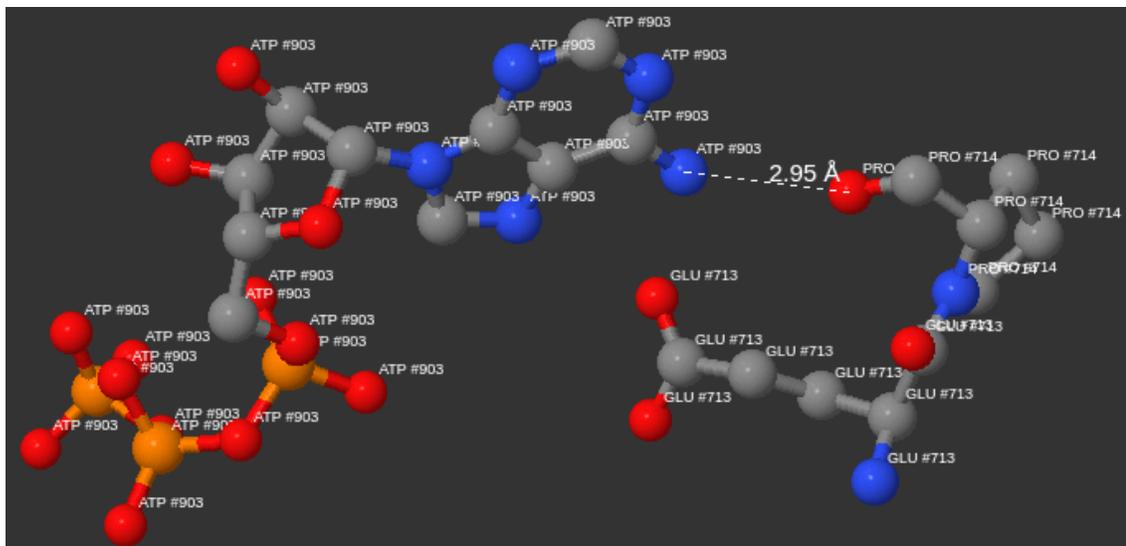


Figura 4.5: Vista tridimensional del resultado #4.

4.3. Evaluación de usabilidad

Considerando que el objetivo central de este proyecto es el diseño de una interfaz gráfica, es necesario evaluar el prototipo construido para conocer qué apreciaciones y observaciones se pueden conseguir de parte de usuarios con conocimientos del área de la biología estructural.

4.3.1. Metodología del proceso

A fin de que los usuarios puedan entregar su percepción basados en una experiencia previa de interacción con el software, el proceso de evaluación se divide en tres etapas.

En primer lugar, los usuarios deben seguir una guía de uso básica de diez pasos, con el objetivo de que ellos puedan experimentar el proceso para diseñar un patrón estructural simple, buscarlo y observar sus resultados.

En segundo lugar, los usuarios deben completar una lista de 30 criterios asignando un valor de cumplimiento a cada uno de ellos, utilizando una escala discreta de 1 (no cumple) a 5 (cumple satisfactoriamente). Los criterios están repartidos en seis categorías (accesibilidad, identidad, navegación, contenido, control y retroalimentación), y guardan relación con los conceptos fundamentales del diseño de interfaces humano-computador [19].

Finalmente, los usuarios deben completar un cuestionario donde proporcionan información sobre ellos, sobre el ambiente de prueba a nivel de software y su opinión abierta respecto a la herramienta revisada.

Los tres documentos empleados (guía de uso, lista de criterios y cuestionario) se encuentran en el Anexo G.

4.3.2. Aspectos de la evaluación

El proceso de evaluación fue apoyado por el profesor del departamento de bioinformática Mauricio Arenas, quien solicitó a catorce de sus estudiantes que participaran de la actividad como evaluadores. Los estudiantes, todos de la carrera Ingeniería en Bioinformática, estaban distribuidos en dos grupos. El primer grupo estaba formado por 10 estudiantes (6 mujeres y 4 hombres) de quinto año pertenecientes a un curso llamado Biotecnología, mientras que el segundo grupo lo integraban 4 estu-

diantes (3 mujeres y 1 hombre) de cuarto año del curso Bioinformática Estructural.

La evaluación se llevó a cabo en instalaciones del Campus Talca. Específicamente, en las salas de clases donde los estudiantes desarrollan las actividades de las asignaturas antes nombradas. Cabe mencionar que el proceso se efectuó en horas de clase.

Por las características del software construido, no fue necesario contar con un tipo de hardware específico para experimentar, así que los estudiantes utilizaron sus propios computadores. Estos equipos debían tener algunos requisitos mínimos: un navegador web y conexión a Internet.

El tiempo destinado para completar toda la actividad fue de 20 minutos, repartido en: 8 minutos para seguir la guía de uso, 7 minutos para completar la lista de criterios y 5 minutos para completar el cuestionario, en ese orden.

4.3.3. Análisis de resultados

Todos los evaluadores declararon que habían tenido experiencia con aplicaciones especializadas relativas a los campos de la biología estructural y bioquímica a nivel molecular, entre las herramientas más nombradas se encuentran: VMD (Visual Molecular Dynamics), Maestro, Gaussian View, BLAST (Basic Local Alignment Search Tool), PyMOL y AutoDock. La mayoría de estas aplicaciones sirven para visualizar y analizar ambientes reales y modelados, otras sirven para buscar similitudes en partes específicas de los organismos o para predecir el acoplamiento de moléculas. Además, en promedio los evaluadores declararon que utilizaban el computador 7,6 horas cada día. Por lo tanto, se puede inferir que los evaluadores poseen la experiencia necesaria para hacer juicios válidos acerca del prototipo desarrollado.

Los siguientes gráficos muestran los tipos de sistemas operativos y navegadores web en los cuales la herramienta fue probada:

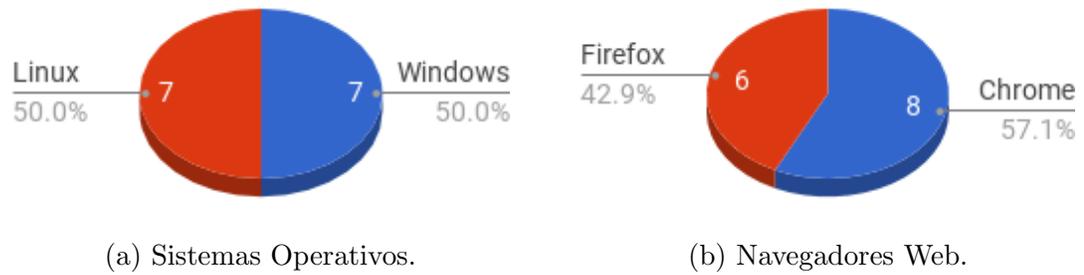


Figura 4.6: Ambiente de evaluación a nivel de software.

En el Cuadro 4.1 se puede observar un resumen del grado de cumplimiento promedio para cada uno de los criterios evaluados por los usuarios. Un resumen de las valoraciones individuales se encuentra tabulado en el Anexo H.

#	Criterio	Valor de cumplimiento \bar{X}
1	El tiempo de carga del sitio web es razonable	4,93
2	El contraste del texto con el fondo es adecuado	4,64
3	El tamaño de letra y espaciado proporcionan fácil lectura	4,86
4	Los iconos permiten distinguir fácilmente su significado	4,43
5	El título principal deja claro el propósito del sitio web	4,00
6	El sitio web es comprensible en 5 segundos (a grandes rasgos)	3,57
7	La presentación de la herramienta es clara y precisa	4,21
8	La información de referencia y contacto es mostrada claramente	4,64
9	La barra de navegación se puede identificar fácilmente	4,07
10	Los enlaces y botones son descriptivos y fáciles de identificar	4,43
11	El contenido es útil y está actualizado	4,57
12	Los estilos y colores son consistentes	4,57
13	El énfasis (negrita, palabras a color, etc.) es usado con moderación	4,71
14	El contenido relacionado está agrupado y bien distribuido	4,64
15	Los ejemplos muestran en forma clara cómo usar la herramienta	4,43
16	Las representaciones gráficas de los elementos biológicos son claras	4,07
17	Las etiquetas de los componentes gráficos son descriptivas	4,36
18	Los cuadros emergentes son comprensibles y llaman la atención	4,79
19	El formato de los resultados es claro y consistente con el patrón estructural	4,43
20	La herramienta es intuitiva, es decir, es posible usarla sin ayuda	3,64
21	La herramienta permite diseñar un patrón estructural en forma sencilla	4,50
22	La herramienta permite un flujo de trabajo fácil	4,71
23	La forma de agregar y eliminar componentes del panel de trabajo es cómoda	4,64
24	El movimiento (arrastrar, uso de flechas) de componentes es fluido	4,64
25	Las opciones de zoom (acercar, alejar, centrar) son útiles	4,50
26	La vista tridimensional facilita el análisis de los resultados	3,71
27	La herramienta guía y limita al usuario	3,64
28	Es siempre claro lo que está pasando en la herramienta	3,93
29	Los mensajes de error explican claramente cómo corregir un problema	3,79
30	La información de ayuda es clara y sin ambigüedad	4,36

Cuadro 4.1: Resumen de cumplimiento para lista de criterios.

Este análisis de tipo cuantitativo indica que el valor promedio del 100 % los criterios están por sobre la mitad de la escala (cumple suficiente), y que el 80 % de

los mismos se encuentra además por sobre la cuarta apreciación (cumple bastante).

El criterio peor evaluado fue “El sitio web es comprensible en 5 segundos”. Esto se puede atribuir a que el criterio en cuestión es genérico para todo tipo de sitios y puede ser poco tiempo para comprender una web con una temática muy particular. Un segundo criterio con baja evaluación fue “La herramienta es intuitiva, es decir, es posible usarla sin ayuda”, el resultado se puede adjudicar al mismo hecho anterior, y es que la temática cubierta por la herramienta es algo difícil hasta para estudiantes de bioinformática. Otro criterio con promedio bajo fue “La vista tridimensional facilita el análisis de los resultados”, esto fue producto a que algunos estudiantes no pudieron apreciar la vista 3D ya que el visor les arrojó un error. Tal error es un problema de jQuery y se da sólo con algunas versiones del navegador Firefox. Los últimos tres criterios que obtuvieron un bajo puntaje pertenecen a la categoría de “Retroalimentación”, es preciso acotar que el prototipo satisface las convenciones básicas para mantener al usuario informado y ayudarlo a corregir errores. Sin embargo, la guía de uso describe pasos claros para diseñar un patrón sin errores, así que debido a eso muchos evaluadores no tuvieron la oportunidad de ver mensajes de error.

Respecto a las preguntas abiertas del cuestionario se obtuvo bastante retroalimentación. En primer lugar, la pregunta asociada a agregar, modificar o eliminar algún contenido o funcionalidad arrojó diversos comentarios, los que se exponen a continuación:

- Descargar los resultados de búsqueda en archivos de texto.
- Agregar filtros para los resultados, por PDB ID, organismo y EC number.
- Incorporar búsqueda de ligandos, ya que son más conocidos por nombre que por su código.
- Agregar el nombre de la estructura asociada al PDB ID dentro de cada resultado.
- Agregar más propiedades a los componentes que ayuden a describir mejor el patrón.
- Utilizar otro visor de moléculas que cargue más rápido.

- Cargar patrones predefinidos.

En segundo lugar, la pregunta asociada a si cambiaría algo de la presentación de la interfaz o su interacción también entregó diversos comentarios, los que se presentan a continuación:

- Modificar el color de la barra de navegación por uno más oscuro que se distinga con mayor facilidad.
- Cambiar el método de eliminación de componentes por separado, haciendo clic una vez en el botón destinado para aquello y luego pudiendo eliminar una cantidad indeterminada de componentes.
- Evitar superposición de aminoácidos nuevos siendo agregados en diferentes partes del panel.
- Colorear más el sitio web, reemplazando principalmente el fondo blanco y resaltando los títulos.
- Colocar el panel de resultados en otro lugar para mejorar su visibilidad.
- Disminuir la cantidad de etiquetas en la representación tridimensional.

Ciertamente, implementar todas las opiniones de los evaluadores ayudaría a que la herramienta, por un lado, agrandaré su conjunto de funcionalidades lo que le daría más potencia y la haría más completa, y por otro lado, mejoraría su apariencia y su modo de interacción.

5. Conclusiones

En este último capítulo se presentan las conclusiones generales que se extrajeron a lo largo del proceso de desarrollo y los resultados obtenidos que dan sustento a la comprobación de la hipótesis. Además, se presentan los aportes del proyecto de memoria y material de trabajo futuro para potenciar la herramienta desarrollada.

5.1. Conclusiones generales

Esta memoria comenzó con una fase de investigación acerca de las interfaces gráficas que son comunes entre las herramientas especializadas para el estudio de la biología estructural de macromoléculas. Aquí se llegó a la conclusión de que la mayoría de esas aplicaciones, muchas de ellas disponibles en sitios web, trabajaban casi exclusivamente mediante introducción de texto en formularios para hacerlas funcionar, es decir, para buscar datos de interés, ver representaciones gráficas, explorar bases de datos biológicos o para realizar análisis de estructuras. A causa de lo anterior, se resolvió diseñar una interfaz con elementos gráficos, en contraposición a lo que se había observado, con el fin de facilitar algún proceso de estudio relativo a la biología estructural.

A continuación, el proyecto se enmarcó dentro de la búsqueda de patrones estructurales, específicamente se determinó trabajar a nivel de interacción entre aminoácidos y ligandos, ya que poder analizar tales patrones es de mucha utilidad, especialmente en el diseño de nuevos fármacos. De inmediato, se investigó sobre herramientas que permiten buscar patrones dentro de macromoléculas y se observó que eran muy parecidas a las ya revisadas, incluso en algunas era necesario conocer lenguajes de programación para hacerlas funcionar. Se concluyó entonces que se utilizaría

una representación basada en la estructura de grafo para los componentes biológicos, ya que se ajusta de forma natural a las estructuras moleculares, facilitando la comprensión del patrón objetivo de la búsqueda.

Luego, se definieron los requisitos que debía cumplir el prototipo funcional, se diseñaron los elementos gráficos que constituirían la interfaz de la herramienta, y a partir de ellos se bosquejaron mockups. La interfaz tuvo una mejora continua a lo largo del proyecto, hasta el término de la etapa de construcción. En este período, primero se adaptó un complemento de visualización de redes para manipular los componentes que dan forma a un patrón estructural, luego se construyeron vistas materializadas a partir de una base de datos relacional para agilizar el proceso de búsqueda, posteriormente se diseñó un método para transformar el patrón buscado en una consulta SQL ejecutable, por último se trabajó en el formato de los resultados y en su representación tridimensional. En suma, todo lo anterior tuvo como producto el prototipo funcional.

Finalmente, se llevó a cabo el proceso de validación de la herramienta, donde se comprobó que la búsqueda de coincidencias para un patrón objetivo entrega todos los resultados y cada resultado posee valores correctos. Además, se logró concluir que la interfaz cumple bastante bien en términos de usabilidad e interacción. Esto se consiguió mediante una evaluación realizada por estudiantes de la carrera Ingeniería en Bioinformática. La evaluación entregó resultados tanto en forma cuantitativa mediante la valoración de criterios de usabilidad como de manera cualitativa a través de respuestas abiertas.

Todo lo anterior, una síntesis del trabajo realizado y sus resultados, permite manifestar la comprobación de la hipótesis propuesta en un comienzo, debido a que sí fue posible diseñar y construir una interfaz gráfica para dar forma a un patrón estructural basado en elementos de grafo, y transformar ese patrón en una consulta SQL para buscar coincidencias en las macromoléculas almacenadas en PDB.

5.2. Aportes del proyecto

El aporte más evidente hacia el campo de la biología estructural y relacionados es el prototipo funcional disponible para todas las personas a través de Internet. Un segundo aporte a estas disciplinas es el proyecto en sí, puesto que intenta empezar a cambiar la forma de interacción entre humanos y las interfaces de herramientas web

especializadas, pasando de los típicos formularios a una representación visual más comprensible en todo sentido.

Por otro lado, se tienen las contribuciones específicas, de las cuales la más relevante es el método para transformar una asociación simple entre un par de componentes gráficos en una consulta en lenguaje SQL y que al unirlos formen una consulta que representa un patrón estructural correctamente, pudiendo dar respuesta a búsquedas de patrones de alta complejidad. Otra contribución en la misma línea, son los esquemas de las vistas materializadas, porque tales vistas contienen la información precisa para buscar las coincidencias sin tener que consultar ninguna otra tabla de la base de datos.

A nivel personal, lo principal fue haber logrado desarrollar un proyecto de larga duración siguiendo una metodología de trabajo flexible, y haber podido aplicar en conjunto muchas de las competencias adquiridas a lo largo de la carrera en sus diferentes cursos. Además, fue muy útil haber tenido la oportunidad de desarrollar un tema innovador, que mezcla conocimientos profundos tanto de computación como de biología a una escala molecular.

5.3. Trabajo futuro

Sin duda todas las herramientas de software se pueden mejorar realizando cambios, implementando nuevas funcionalidades, optimizando recursos o simplemente corrigiendo errores. El software resultante de este proyecto, por su naturaleza de prototipo, no es un producto completo, y a medida que se fue desarrollando surgieron nuevos requisitos que harían de él una herramienta más poderosa y flexible.

Un primer cambio sería incorporar en cada resultado, junto a su forma textual, una imagen que mantenga la estructura del patrón diseñado, pero que incluya los valores del resultado correspondiente. Con esto se cumpliría a cabalidad el requisito “Mostrar resultados de búsqueda”.

Una mejora sería poder asignar otras clasificaciones a los aminoácidos comodín, como por ejemplo: aromático, alifático, azufrado, hidroxilado, etc. Otra mejora sería modificar algunos colores de la interfaz y optimizar su modo de interacción.

Por último, en cuanto a nuevas funcionalidades se consideran las siguientes:

- Agregar un nuevo componente, similar al aminoácido comodín, que permita especificar los tipos de aminoácidos que se deberían incluir en su lugar al realizar la búsqueda del patrón estructural.
- Agregar un nuevo componente que permita representar varios aminoácidos al mismo tiempo. Esto pensando en la secuencia de las cadenas de aminoácidos que son de interés dentro del patrón objetivo.
- Un medio para descargar los resultados de la búsqueda o parte de ellos en varios formatos de archivo (e.g. XML, CSV, XLSX, TXT).
- Mostrar estadísticas de los resultados de la búsqueda. Esto facilitaría su análisis dando a conocer cuántas veces se repite un determinado valor en un aminoácido comodín, cuántas veces se repite una determinada configuración del patrón objetivo, o en qué macromoléculas se encuentra el patrón más veces.
- Incorporar las funcionalidades señaladas por los evaluadores que fueron presentadas en la Subsección 4.3.3. Entre las más relevantes se encuentran: agregar filtros para los resultados y un buscador de códigos de ligandos.

Glosario

Ajax: Tecnología que sirve para comunicarse con un servidor, intercambiar datos, y actualizar una página web sin tener que recargarla.

CSS: Lenguaje de hojas de estilo en cascada, estándar para definir la presentación de un documento escrito en lenguaje de marcado.

Grafo: Estructura constituida por un conjunto de objetos llamados vértices o nodos, donde cada uno de los pares de vértices relacionados se denomina arista.

HTML: Lenguaje de marcado de hipertexto, estándar para elaborar páginas web.

JavaScript: Lenguaje dinámico de alto nivel, estándar para la producción y manipulación de contenido en páginas web modernas.

JOIN: Operación SQL para combinar filas de dos o más tablas, en función de una columna relacionada entre ellas.

JQuery: Biblioteca para JavaScript que simplifica la codificación y uso de funciones, y facilita la compatibilidad con distintos navegadores.

PHP: Lenguaje del lado del servidor más utilizado para desarrollo web.

PostgreSQL: Sistema de base de datos relacional de código abierto, rico en características e interfaces para muchos lenguajes de programación.

Python: Lenguaje de programación de propósito general ampliamente usado.

SQL: Lenguaje de consulta estructurada para acceder a bases de datos relacionales.

XML: Lenguaje de marcado extensible para almacenar datos en forma legible.

Bibliografía

- [1] Antonina Andreeva, Dave Howorth, Cyrus Chothia, Eugene Kulesha, and Alexey G. Murzin. SCOP2 prototype: a new approach to protein structure mining. *Nucleic Acids Research*, 2013.
- [2] Mauricio Arenas-Salinas, Samuel Ortega-Salazar, Fernando Gonzales-Nilo, Ehmke Pohl, David S. Holmes, and Raquel Quatrini. AFAL: a web service for profiling amino acids surrounding ligands in proteins. *Journal of Computer-Aided Molecular Design*, 28(11):1069–1076, 2014.
- [3] Helen M. Berman, Stephen K. Burley, Gerard J. Kleywegt, John L. Markley, Haruki Nakamura, and Sameer Velankar. The archiving and dissemination of biological structure data. *Current Opinion in Structural Biology*, 40:17 – 22, 2016.
- [4] Helen M. Berman, Gerard J. Kleywegt, Haruki Nakamura, and John L. Markley. The Protein Data Bank archive as an open data resource. *Journal of Computer-Aided Molecular Design*, 28(10):1009–1014, 2014.
- [5] Helen M. Berman, John Westbrook, Zukang Feng, Gary Gilliland, Talapady N. Bhat, Helge Weissig, Ilya N. Shindyalov, and Philip E. Bourne. The Protein Data Bank. *Nucleic acids research*, 28(1):235–242, 2000.
- [6] Barry W. Boehm. Verifying and validating software requirements and design specifications. *IEEE software*, 1(1):75, 1984.
- [7] ChemAxon. Marvin: intuitive applications and API for chemical sketching, visualization and data exploration. <https://www.chemaxon.com/products/marvin/>, Consultado el 02 de octubre de 2016.

- [8] Zukang Feng, Li Chen, Himabindu Maddula, Ozgur Akcan, Rose Oughtred, Helen M. Berman, and John Westbrook. Ligand Depot: a data warehouse for ligands bound to macromolecules. *Bioinformatics*, 20(13):2153–2155, 2004.
- [9] Anna M. Gallina, Peer Bork, and Domenico Bordo. Structural analysis of protein-ligand interactions: the binding of endogenous compounds and of synthetic drugs. *Journal of Molecular Recognition*, 27(2):65–72, 2014. JMR-13-0036.
- [10] Ana M. González. Diseño de una base de datos relacional para el Protein Data Bank, Universidad de Talca, 2015.
- [11] The PostgreSQL Global Development Group. PostgreSQL 9.4.12 Documentation. <https://www.postgresql.org/docs/9.4/static/index.html>, Consultado el 4 de junio de 2017.
- [12] Kori M. Inkpen. Drag-and-Drop versus Point-and-Click mouse interaction styles for children. *ACM Trans. Comput.-Hum. Interact.*, 8(1):1–33, March 2001.
- [13] Akira R. Kinjo, Gert-Jan Bekker, Hirofumi Suzuki, Yuko Tsuchiya, Takeshi Kawabata, Yasuyo Ikegawa, and Haruki Nakamura. Protein Data Bank Japan (PDBj): updated user interfaces, resource description framework, analysis tools for large structures. *Nucleic Acids Research*, page gkw962, 2016.
- [14] Akira R. Kinjo, Reiko Yamashita, and Haruki Nakamura. PDBj Mine: design and implementation of relational database interface for Protein Data Bank Japan. *Database*, 2010:baq021, 2010.
- [15] Trudy McKee and James R. McKee. *Bioquímica: las bases moleculares de la vida*. McGraw-Hill, 2014.
- [16] David L. Mobley and Ken A. Dill. Binding of small-molecule ligands to proteins: “what you see” is not always “what you get”. *Structure*, 17(4):489–498, 2009.
- [17] David L. Nelson and Michael M. Cox. *Lehninger Principles of Biochemistry: 6th Edition*. W.H. Freeman and Company, 2012.
- [18] RCSB PDB. RCSB Protein Data Bank. <http://www.rcsb.org/>, Consultado el 27 de septiembre de 2016.

- [19] J. Preece, H. Sharp, and Y. Rogers. *Interaction Design: Beyond Human-Computer Interaction*. Wiley, 2015.
- [20] Ewgenij Proschak, Jörg K. Wegner, Andreas Schüller, Gisbert Schneider, and Uli Fechner. Molecular Query Language (MQL): A context-free grammar for substructure matching. *Journal of Chemical Information and Modeling*, 47(2):295–301, 2007. PMID: 17381167.
- [21] Bernhard Rupp. *Biomolecular Crystallography: principles, practice, and application to structural biology*. Garland Science, 2009.
- [22] N. Schneider, G. Lange, S. Hindle, R. Klein, and M. Rarey. A consistent description of HYdrogen bond and DEhydration energies in protein-ligand complexes: methods behind the HYDE scoring function. *Journal of Computer-Aided Molecular Design*, 27(1):15–29, 2003.
- [23] David Sehnal, Lukáš Pravda, Radka Svobodová Vařeková, Crina-Maria Ionescu, and Jaroslav Koča. PatternQuery: web application for fast detection of biomacromolecular structural patterns in the entire Protein Data Bank. *Nucleic acids research*, page gkv561, 2015.
- [24] Paul Shannon, Andrew Markiel, Owen Ozier, Nitin S. Baliga, Jonathan T. Wang, Daniel Ramage, Nada Amin, Benno Schwikowski, and Trey Ideker. Cytoscape: a software environment for integrated models of biomolecular interaction networks. *Genome research*, 13(11):2498–2504, 2003.
- [25] Christian Von Mering, Lars J. Jensen, Berend Snel, Sean D. Hooper, Markus Krupp, Mathilde Foglierini, Nelly Jouffre, Martijn A. Huynen, and Peer Bork. STRING: known and predicted protein–protein associations, integrated and transferred across organisms. *Nucleic acids research*, 33(suppl 1):D433–D437, 2005.
- [26] wwPDB. wwPDB: Worldwide Protein Data Bank. <http://www.wwpdb.org/>, Consultado el 26 de abril de 2017.

ANEXOS

A. Extracto archivo de la Hemoglobina (4HHB.pdb)

Las entradas en PDB almacenan una gran cantidad de datos. Cada archivo contiene coordenadas atómicas, nombres de las moléculas, información de la estructura primaria y secundaria, información de la unión de ligandos, detalles sobre colección de datos, citas bibliográficas, entre otros [26]. Todo lo anterior se reparte en secciones que ayudan a estructurar los archivos de una manera comprensible. Detalles del formato se pueden encontrar en el sitio web de la wwPDB.

```
HEADER      OXYGEN TRANSPORT                      07-MAR-84  4HHB
TITLE       THE CRYSTAL STRUCTURE OF HUMAN DEOXYHAEMOGLOBIN AT 1.74 ANGSTROMS
COMPND     2 MOLECULE: HEMOGLOBIN (DEOXY) (ALPHA CHAIN);
SOURCE     2 ORGANISM_SCIENTIFIC: HOMO SAPIENS;
KEYWDS     OXYGEN TRANSPORT
EXPDTA     X-RAY DIFFRACTION
AUTHOR     G.FERMI, M.F.PERUTZ
SEQRES    1 A  141  VAL LEU SER PRO ALA ASP LYS THR ASN VAL LYS ALA ALA
           2 A  141  TRP GLY LYS VAL GLY ALA HIS ALA GLY GLU TYR GLY ALA
HET       HEM  A 142      43
HET       HEM  B 148      43
HET       HEM  C 142      43
HET       HEM  D 148      43
HETNAM     HEM PROTOPORPHYRIN IX CONTAINING FE
ATOM       1  N  VAL A  1      6.204  16.869   4.854  1.00  49.05      N
ATOM       2  CA VAL A  1      6.913  17.759   4.607  1.00  43.14      C
ATOM       3  C  VAL A  1      8.504  17.378   4.797  1.00  24.80      C
HETATM    4389  CHA HEM A 142      8.585   7.902 -18.282  1.00  16.31      C
HETATM    4390  CHB HEM A 142     10.355   9.805 -14.208  1.00  26.27      C
HETATM    4391  CHC HEM A 142      8.341   6.363 -11.589  1.00  13.23      C
CONECT     650  4431
CONECT     1771 4475
CONECT     2844 4518
MASTER    780   1   6  32   0   0  16   9 4779   4 180  46
END
```

B. Lista de herramientas para PDB revisadas

La siguiente tabla incluye las herramientas y servicios revisados los cuales hacen uso de la información que dispone PDB. Por cada una se especifica su nombre, tipo (búsqueda, exploración, visualización o análisis) y la fuente donde se encuentra publicada (Sep. 2016). La mayoría de estas aplicaciones provienen de los sitios web oficiales RCSB PDB, PDBj, y PDBe.

Nombre	Tipo	Fuente
Advanced Search Interface	Búsqueda	http://www.rcsb.org/pdb/search/advSearch.do?search=new
ASH	Búsqueda	http://sysimm.ifrec.osaka-u.ac.jp/ash/
Browse by Annotation	Exploración	http://www.rcsb.org/pdb/browse/jbrowse.do?t=4&useMenu=no
Chemie Search	Búsqueda	http://pdbj.org/help/chemie
Cytoscape	Análisis	http://www.cytoscape.org/index.html
Drug and Drug Target Mapping	Búsqueda	http://www.rcsb.org/pdb/ligand/drugMapping.do
EM Navigator	Búsqueda	http://pdbj.org/emnavi/
Explore the PDB Archive	Exploración	http://www.rcsb.org/pdb/secondary.do?p=v2/secondary/search.jsp
HOMCOS	Búsqueda	http://homcos.pdbj.org/?LANG=en

Human Gene View	Visualización	http://www.rcsb.org/pdb/staticHelp.do?p=help/geneView.html
JSmol	Visualización	http://www.rcsb.org/pdb/explore/jmol.do?structureId=4HHB&opt=3&bionumber=1
jV	Visualización	http://pdj.org/jv/index.html
Latest Release Search	Búsqueda	http://pdj.org/pdbj-update
Ligand Expo Browser	Exploración	http://ligand-expo.rcsb.org/pyapps/ldHandler.py?formid=cc-browse-search&operation=smiles&target=\\$&category=aa&first=1
Ligand Explorer	Visualización	http://www.rcsb.org/pdb/static.do?p=help/viewers/ligandExplorer_viewer.html
Ligand Expo Search	Búsqueda	http://ligand-expo.rcsb.org/ld-search.html
MAFFTash	Búsqueda	https://sysimm.ifrec.osaka-u.ac.jp/MAFFTash/
Map Genomic Position to Protein Sequence and 3D Structure	Análisis	http://www.rcsb.org/pdb/chromosome.do
Mine Advanced Search	Búsqueda	http://pdj.org/help/advanced-search
Mine Basic	Búsqueda	http://pdj.org/help/mine
Mine Large Structures	Búsqueda	http://pdj.org/mine/search?query=tag:large_structures
Mine SQL Search	Búsqueda	http://pdj.org/help/mine2-sql
Molmil	Visualización	http://pdj.org/help/molmil
NGL	Visualización	https://github.com/arose/ngl
NMR OLDERADO	Análisis	http://www.ebi.ac.uk/pdbe/nmr/olderado/
NMR Vivaldi	Visualización	http://www.ebi.ac.uk/pdbe/vivaldi/
Omokage	Búsqueda	http://pdj.org/emnavi/omo-search.php?lang=en

Pathway View	Análisis	http://www.rcsb.org/pdb/pathway/pw.do
PDBeChem	Búsqueda	http://www.ebi.ac.uk/pdbe-srv/pdbechem/
PDBeFold	Análisis	http://www.ebi.ac.uk/msd-srv/ssm/
PDBePISA	Análisis	http://www.ebi.ac.uk/msd-srv/prot_int/pistart.html
PDBeMotif	Búsqueda	http://www.ebi.ac.uk/pdbe-site/pdbemotif/
PDBeXpress	Análisis	http://www.ebi.ac.uk/pdbe-srv/pdbexpress/
PoseView	Visualización	http://poseview.zbh.uni-hamburg.de/
Protein-protein Interaction	Análisis	http://poseview.zbh.uni-hamburg.de/pdb_files
Protein Comparison Tool	Análisis	http://www.rcsb.org/pdb/workbench/workbench.do
Protein Feature View	Visualización	http://www.rcsb.org/pdb/staticHelp.do?p=help/featureView.html
Protein Workshop	Visualización	http://www.rcsb.org/pdb/static.do?p=help/viewers/proteinWorkshop_viewer.html
PV	Visualización	http://www.rcsb.org/pdb/pv/pv.do?pdbid=4HHB&bionumber=1
SCOP2	Exploración	http://scop2.mrc-lmb.cam.ac.uk/graph/#/daiN/vR20/RR8th_Z9.dat#1#is_a,part_of,occurred_in
Search by Sequences	Búsqueda	http://www.rcsb.org/pdb/secondary.do?p=v2/secondary/search.jsp
Search by Ligands (Chemical Component Search)	Búsqueda	http://www.rcsb.org/pdb/ligand/chemAdvSearch.do
Sequence Navigator	Búsqueda	http://pdj.org/help/sequence-navigator

Sequence Search	Búsqueda	http://www.ebi.ac.uk/pdbe-srv/PDBeXplore/sequence/
SeSAW	Búsqueda	https://sysimm.ifrec.osaka-u.ac.jp/sesaw.2.0/
Simple Viewer	Visualización	http://www.rcsb.org/pdb/static.do?p=help/viewers/simpleViewer_viewer.html
Status Search	Búsqueda	http://pd bj.org/status-search
String Database	Análisis	http://string-db.org/
Structure Navigator	Búsqueda	http://pd bj.org/help/structure-navigator
UniPDB	Análisis	http://www.ebi.ac.uk/pdbe/widgets/unipdb
Yorodumi	Visualización	http://pd bj.org/emnavi/viewtop.php

C. Requisitos funcionales de la aplicación

Las siguientes tablas contienen la especificación de las funcionalidades requeridas en la aplicación, para cumplir satisfactoriamente con el objetivo principal de esta memoria.

Requisito #: RF001	Agregar ligando
Descripción: Debe existir un procedimiento para agregar un elemento gráfico que represente un ligando en el patrón estructural.	
Restricciones: La búsqueda de patrones se limita a un único ligando.	
Criterio de Aceptación: Existe al menos una manera de agregar un ligando al panel de trabajo donde se da forma al patrón.	
Dependencias: Ninguna	Conflictos: Ninguno

Cuadro C.1: Requisito funcional: Agregar ligando.

Requisito #: RF002	Agregar aminoácidos
Descripción: Debe existir un procedimiento para agregar un elemento gráfico que represente un aminoácido en el patrón estructural.	
Restricciones: El patrón puede incorporar una cantidad variable e indeterminada de aminoácidos.	
Criterio de Aceptación: Existe al menos una manera de agregar aminoácidos al panel de trabajo donde se da forma al patrón.	
Dependencias: Ninguna	Conflictos: Ninguno

Cuadro C.2: Requisito funcional: Agregar aminoácidos.

Requisito #: RF003	Agregar aminoácidos comodín
Descripción: Debe existir un procedimiento para agregar un elemento gráfico que represente un aminoácido cualquiera en el patrón estructural.	
Restricciones: Este componente no requiere de un identificador ya que puede tomar cualquier valor de entre los veinte aminoácidos que existen.	
Criterio de Aceptación: Existe al menos una manera de agregar este tipo especial de aminoácidos al panel de trabajo donde se da forma al patrón.	
Dependencias: Ninguna	Conflictos: Ninguno

Cuadro C.3: Requisito funcional: Agregar aminoácidos comodín.

Requisito #: RF004	Identificar ligando y aminoácidos
Descripción: Se debe presentar un medio que permita añadir o modificar el identificador tanto del ligando como de los aminoácidos.	
Restricciones: En el caso del ligando el identificador es su código de tres letras, mientras que para los aminoácidos es la abreviatura (tres letras) de su nombre.	
Criterio de Aceptación: Existe una vista que permite ingresar texto o seleccionar de un listado el identificador de los componentes del patrón.	
Dependencias: RF001 y RF002	Conflictos: Ninguno

Cuadro C.4: Requisito funcional: Identificar ligando y aminoácidos.

Requisito #: RF005	Agregar conector distancia
Descripción: Debe existir un modo para agregar un elemento gráfico que conecte pares aminoácido-aminoácido y ligando-aminoácido, el cual representa la distancia entre ellos.	
Restricciones: La distancia determina el radio esférico de revisión en torno a un componente del patrón estructural.	
Criterio de Aceptación: Existe al menos una manera para incorporar un conector distancia entre dos componentes dentro del panel de trabajo.	
Dependencias: RF001, RF002 y RF003	Conflictos: Ninguno

Cuadro C.5: Requisito funcional: Agregar conector distancia.

Requisito #: RF006	Agregar conector siguiente
Descripción: Debe existir un modo para agregar un elemento gráfico que conecte pares aminoácido-aminoácido, el cual representa la relación de secuencia entre ellos.	
Restricciones: La relación de un aminoácido con el siguiente es unidireccional, debe quedar explícita su dirección.	
Criterio de Aceptación: Existe al menos una manera para incorporar un conector siguiente entre dos aminoácidos dentro del panel de trabajo.	
Dependencias: RF002 y RF003	Conflictos: Ninguno

Cuadro C.6: Requisito funcional: Agregar conector siguiente.

Requisito #: RF007	Incorporar propiedades
Descripción: Se debe presentar un medio que permita modificar la distancia en ese tipo de conectores. Además, tener la posibilidad de añadir o modificar una clasificación en los aminoácidos comodín.	
Restricciones: En el caso del conector su distancia predeterminada es un rango con mínimo de 0.5Å y máximo de 7.0Å, los que pueden variar dentro de estos límites. Por otro lado, los aminoácidos comodín pueden tener una de las siguientes clasificaciones: cualquiera, no polares, polares sin carga, polares positivos y polares negativos.	
Criterio de Aceptación: Existe una vista que permite ingresar texto o seleccionar de un listado las propiedades de los componentes del patrón.	
Dependencias: RF005	Conflictos: RF003

Cuadro C.7: Requisito funcional: Incorporar propiedades.

Requisito #: RF008	Eliminar componentes
Descripción: Debe existir un procedimiento para eliminar componentes por separado (ligando, aminoácido, conector siguiente y distancia), y otro para eliminarlos todos al mismo tiempo.	
Restricciones: Cuando el ligando o un aminoácido es eliminado, los conectores unidos a él también son eliminados.	
Criterio de Aceptación: Existen al menos dos maneras de eliminar componentes del panel de trabajo donde se da forma al patrón.	
Dependencias: RF005 y RF006	Conflictos: Ninguno

Cuadro C.8: Requisito funcional: Eliminar componentes.

Requisito #: RF009	Buscar patrón en PDB
Descripción: Debe existir un mecanismo que permita elaborar una consulta SQL a partir de un patrón estructural diseñado correctamente, y ejecutar tal consulta sobre PDB.	
Restricciones: Para llevar a cabo la búsqueda, el patrón debe contener al menos un ligando y un aminoácido, unidos mediante un conector de distancia.	
Criterio de Aceptación: Existe al menos una forma para iniciar el proceso de búsqueda.	
Dependencias: RF004, RF006 y RF007	Conflictos: Ninguno

Cuadro C.9: Requisito funcional: Buscar patrón en PDB.

Requisito #: RF010	Mostrar resultados de búsqueda
Descripción: Se debe presentar un medio que permita mostrar en forma gráfica los resultados de la consulta SQL ejecutada sobre PDB.	
Restricciones: Cada coincidencia del patrón buscado debe ser presentada en una imagen, que mantenga la estructura del patrón diseñado, pero que incluya los datos del resultado correspondiente.	
Criterio de Aceptación: Existe una vista de tipo listado que muestra cada resultado en forma independiente.	
Dependencias: RF009	Conflictos: Ninguno

Cuadro C.10: Requisito funcional: Mostrar resultados de búsqueda.

Requisito #: RF011	Enlazar resultados a PDB
Descripción: Se debe vincular cada resultado al PDB ID de la macromolécula donde se encontró la coincidencia estructural, para luego enlazarlo a su página correspondiente en RCSB PDB.	
Restricciones: Abrir nueva “pestaña” en el navegador cada vez que se seleccione el PDB ID de un resultado.	
Criterio de Aceptación: Existe al menos una forma para dirigirse a la página oficial de cada PDB ID.	
Dependencias: RF010	Conflictos: Ninguno

Cuadro C.11: Requisito funcional: Enlazar resultados a PDB.

Requisito #: RF012	Visualizar resultados en JSmol
Descripción: Se debe presentar un medio que permita mostrar los patrones coincidentes en su forma tridimensional a través del visualizador JSmol.	
Restricciones: Sólo se deben exhibir los componentes que integran el patrón estructural diseñado, no la macromolécula completa.	
Criterio de Aceptación: Existe un mecanismo para desplegar una vista donde se muestra la representación 3D de cada patrón coincidente.	
Dependencias: RF010	Conflictos: Ninguno

Cuadro C.12: Requisito funcional: Visualizar resultados en JSmol.

D. Códigos SQL para creación de vistas materializadas

Código para crear la vista *DistanceHetAmino*

```
CREATE MATERIALIZED VIEW distance_het_amino AS
SELECT het_id, het_symbol, amino_id, amino_symbol, amino_class, min(distance), max(distance)
FROM
(
  SELECT hets.id AS het_id, hets.symbol AS het_symbol, aminoacid.id AS amino_id,
         aminoacidstandard.symbol AS amino_symbol, aminoacidstandard.class AS amino_class,
         distanceatomaminoatomhet.distance
  FROM
    (SELECT id, protein_id, symbol FROM het) AS hets
  JOIN aminoacid ON aminoacid.protein_id = hets.protein_id
  JOIN aminoacidstandard ON aminoacidstandard.symbol = aminoacid.symbol
  JOIN atomhet ON atomhet.het_id = hets.id
  JOIN atomamino ON atomamino.amino_id = aminoacid.id
  JOIN distanceatomaminoatomhet ON distanceatomaminoatomhet.atom_het_id = atomhet.id
    AND distanceatomaminoatomhet.atom_amino_id = atomamino.id
) AS sub
GROUP BY sub.het_id, sub.het_symbol, sub.amino_id, sub.amino_symbol, sub.amino_class
ORDER BY het_symbol;
```

Código para crear la vista *DistanceAminoAmino*

```

CREATE MATERIALIZED VIEW distance_amino_amino AS
SELECT amino1_id, amino1_symbol, amino1_class, amino2_id, amino2_symbol, amino2_class,
       min(distance), max(distance)
FROM
(
  SELECT aminos1.id AS amino1_id, aminos1.symbol AS amino1_symbol,
         aminos1.class AS amino1_class, aminos2.id AS amino2_id,
         aminos2.symbol AS amino2_symbol, aminos2.class AS amino2_class,
         distanceatomaminoatomamino.distance
  FROM
  (
    (SELECT id, symbol, class, protein_id
     FROM aminoacid NATURAL JOIN aminoacidstandard) AS aminos1
    JOIN
    (SELECT id, symbol, class, protein_id
     FROM aminoacid NATURAL JOIN aminoacidstandard) AS aminos2
    ON aminos2.protein_id = aminos1.protein_id
    JOIN (SELECT id, amino_id FROM atomamino) AS atomamino1
    ON aminos1.id = atomamino1.amino_id
    JOIN (SELECT id, amino_id FROM atomamino) As atomamino2
    ON aminos2.id = atomamino2.amino_id
    JOIN distanceatomaminoatomamino
    ON distanceatomaminoatomamino.atom_amino_id_1 = atomamino1.id
       AND distanceatomaminoatomamino.atom_amino_id_2 = atomamino2.id
  )
) AS sub
GROUP BY sub.amino1_id, sub.amino1_symbol, sub.amino1_class, sub.amino2_id,
       sub.amino2_symbol, sub.amino2_class;

```

Código para crear la vista *NextAminoAmino*

```

CREATE MATERIALIZED VIEW next_amino_amino AS
SELECT aminos1.id AS amino1_id, aminos1.symbol AS amino1_symbol,
       aminos1.class AS amino1_class, aminos2.id AS amino2_id,
       aminos2.symbol AS amino2_symbol, aminos2.class AS amino2_class
FROM
(
  (SELECT id, symbol, class, next_amino
   FROM aminoacid NATURAL JOIN aminoacidstandard) AS aminos1
  JOIN
  (SELECT id, symbol, class
   FROM aminoacid NATURAL JOIN aminoacidstandard) AS aminos2
  ON aminos2.id = aminos1.next_amino
)
ORDER BY amino1_id;

```

E. Capturas de la aplicación web

El sitio web que da soporte a la herramienta se divide en cinco secciones, las que se exhiben a continuación.

Sección “Presentation”

GSP4PDB Presentation Instructions Tool Examples About

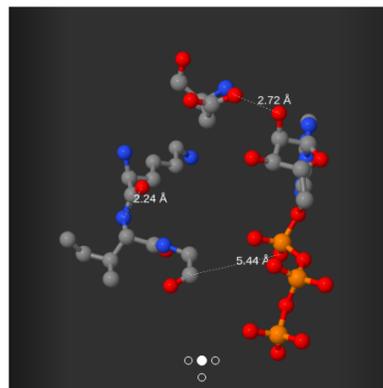
Presentation

Well defined biomacromolecular patterns such as binding sites between a region of a protein and a ligand regulate certain factors that make possible the origin of many important biological phenomena. **GSP4PDB** is a user-friendly web application that lets the users design, search and analyze structural patterns inside biomacromolecules in a graphic way. This application uses the Protein Data Bank ([PDB](#)) to obtain detailed structural information.

The program receives a graph-based pattern formed by graphic components that represent a ligand and the amino acids around it. The pattern also includes some associations that condition the structure: distances between the ligand and the amino acids, distances between amino acids, and sequence of amino acids in the chain.

The user can determine what type of ligand want to analyze and include fixed amino acids but also can include amino acids 'any'. They can take the value of any amino acid, although the user can restrict them to a specific classification according to their polarity.

Finally, the program looks for coincidences inside each macromolecule in the PDB. The results of the search are shown in textual form and graphically by a 3D viewer. This analysis allows comparing proteins, DNA or RNA where the structure exists and evaluating how the amino acids surround the ligand in each coincidence.

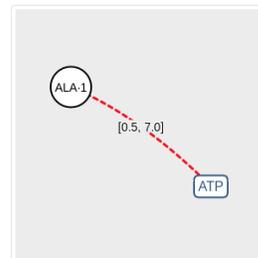


Sección “Instructions”

GSP4PDB Presentation Instructions Tool Examples About

Instructions

1. In the work panel, click on the **LIGAND** button to add a ligand of interest. This is mandatory.
2. Add amino acids by clicking on the buttons **AMINO ACID** and **ANY AMINO** as you like. You must add at least one of these obligatorily.
3. Connect all the components that you added to the panel using the buttons **DISTANCE** and **NEXT**. Click on **DISTANCE** to associate two components according to their proximity measured in angstroms (Å). Default range of distance is [0.5, 7.0]. You can modify this range doing double-click on the red lines. You have to use this association to connect the ligand and one amino acid obligatorily. Click on **NEXT** to associate two amino acids according to the sequence of the chain of amino acids inside of the biomacromolecule.
4. After you have designed the structural pattern correctly, click on the **SEARCH** button and the application will start looking for coincidences of your pattern in PDB.
5. Observe the results of the search shown individually on the panel of the right. You can analyze the structure of each coincidence in the 3D viewer by clicking the **3D STRUCTURE** buttons.



Example of the Simplest Pattern

Sección “The Tool”

GSP4PDB Presentation Instructions Tool Examples About

Make a structural pattern of amino acids and a ligand

Search results: 1266 coincidences

Format of the Results

PDB ID: 1MAU RESULT #1 3D STRUCTURE

---○: LIGAND-AMINO distances
 [ATP]{ATP #400}-----3.0-----(ASN-1){ASN #18}

---○: AMINO-AMINO distances

→○: AMINO-AMINO chain sequence
 (ANY-2){GLY #17}→NEXT→(ASN-1){ASN #18}

PDB ID: 1NM1 RESULT #2 3D STRUCTURE

---○: LIGAND-AMINO distances
 [ATP]{ATP #401}-----4.6-----(ASN-1){ASN #12}

---○: AMINO-AMINO distances

← PREV 1 - 100 NEXT →

Sección “Examples”

GSP4PDB Presentation Instructions Tool Examples About

Examples

GSP4PDB: Graph-based Structural Patterns for PDB - Chromium

Example 1: Ligand ATP and amino acids ASN and one ANY

GSP4PDB Presentation Instructions Tool Examples About

Make a structural pattern of amino acids and a ligand

AMINO ACID
ANY AMINO
LIGAND
DISTANCE
NEXT

ASN-1
ANY-2
ATP

[0, 5, 0]

Search results: 139 coincidences

Format of the Results

PDB ID: 1C3H RESULT #1 3D STRUCTURE

---: LIGAND-AMINO distances
[ATP] [ATP #299]-----4, 6-----[ASN-1] [ASN #136]

---: AMINO-AMINO distances

---: AMINO-AMINO chain sequence
(ANY-2) [ASP #135]-----NEXT-----[ASN-1] [ASN #136]

PDB ID: 3NMS RESULT #2 3D STRUCTURE

---: LIGAND-AMINO distances
[ATP] [ATP #493]-----4, 6-----[ASN-1] [ASN #12]

---: AMINO-AMINO distances

Iniciar grabación
Pausar grabación
Finalizar grabación
Salir

1-100

GSP4PDB: Grap

01:01

Sección “About”

About...

GSP4PDB is part of the services provided by the [Bioinformatic Group](#) of the [University of Talca](#)

When using GSP4PDB, please consult (and cite) the following reference:

Diego A. Cisterna. Diseño de una Interfaz Gráfica para Búsqueda de Patrones Estructurales en el Protein Data Bank, Julio 2017. [C](#)

Send your questions, suggestions or comments to the email contact: dcisterna@live.com

© 2017 Designed by [Diego Cisterna](#) and [Ph.D. Renzo Angles](#)

F. Códigos SQL asociados a los patrones de prueba

Código del patrón de prueba número uno

```
SELECT * FROM (
  SELECT * FROM (
    SELECT het_id, amino_id AS amino1_id, amino_symbol AS amino1_symbol, min AS min_het_amino1
    FROM distance_het_amino
    WHERE het_symbol = 'ATP' AND amino_symbol = 'N'
      AND ((min < 0.5 AND max >= 0.5) OR (min <= 7.0 AND max > 7.0)
      OR (min >= 0.5 AND max <= 7.0) OR (min < 0.5 AND max > 7.0))
  ) AS q0 NATURAL JOIN (
    SELECT amino1_id AS amino2_id, amino1_symbol AS amino2_symbol, amino2_id AS amino1_id,
    amino2_symbol AS amino1_symbol
    FROM next_amino_amino
    WHERE amino1_class = 4 AND amino2_symbol = 'N'
  ) AS q1
) AS query
WHERE amino1_id <> amino2_id;
```

Código del patrón de prueba número dos

```

SELECT * FROM (
  SELECT * FROM (
    SELECT het_id, amino_id AS amino2_id, amino_symbol AS amino2_symbol, min AS min_het_amino2
    FROM distance_het_amino
    WHERE het_symbol = 'ATP' AND amino_symbol = 'H'
      AND ((min < 2.0 AND max >= 2.0) OR (min <= 5.0 AND max > 5.0)
      OR (min >= 2.0 AND max <= 5.0) OR (min < 2.0 AND max > 5.0))
  ) AS q0 NATURAL JOIN (
    SELECT * FROM (
      SELECT amino1_id AS amino2_id, amino1_symbol AS amino2_symbol, amino2_id AS amino3_id,
      amino2_symbol AS amino3_symbol, min AS min_amino2_amino3
      FROM distance_amino_amino
      WHERE ((min < 0.5 AND max >= 0.5) OR (min <= 3.0 AND max > 3.0)
      OR (min >= 0.5 AND max <= 3.0) OR (min < 0.5 AND max > 3.0))
      AND amino1_symbol = 'H'
    UNION
      SELECT amino2_id AS amino2_id, amino2_symbol AS amino2_symbol, amino1_id AS amino3_id,
      amino1_symbol AS amino3_symbol, min AS min_amino2_amino3
      FROM distance_amino_amino
      WHERE ((min < 0.5 AND max >= 0.5) OR (min <= 3.0 AND max > 3.0)
      OR (min >= 0.5 AND max <= 3.0) OR (min < 0.5 AND max > 3.0))
      AND amino2_symbol = 'H'
    ) AS q_23
  ) AS q1 NATURAL JOIN (
    SELECT amino1_id AS amino3_id, amino1_symbol AS amino3_symbol, amino2_id AS amino2_id,
    amino2_symbol AS amino2_symbol
    FROM next_amino_amino
    WHERE amino2_symbol = 'H'
  ) AS q2 NATURAL JOIN (
    SELECT amino1_id AS amino2_id, amino1_symbol AS amino2_symbol, amino2_id AS amino1_id,
    amino2_symbol AS amino1_symbol
    FROM next_amino_amino
    WHERE amino1_symbol = 'H' AND amino2_symbol = 'N'
  ) AS q3
) AS query
WHERE amino2_id <> amino3_id AND amino2_id <> amino1_id AND amino3_id <> amino1_id;

```

Código del patrón de prueba número tres

```

SELECT * FROM (
  SELECT * FROM (
    SELECT het_id, amino_id AS amino1_id, amino_symbol AS amino1_symbol, min AS min_het_amino1
    FROM distance_het_amino
    WHERE het_symbol = 'ATP' AND amino_symbol = 'D'
      AND ((min < 2.5 AND max >= 2.5) OR (min <= 4.5 AND max > 4.5)
      OR (min >= 2.5 AND max <= 4.5) OR (min < 2.5 AND max > 4.5))
  ) AS q0 NATURAL JOIN (
    SELECT het_id, amino_id AS amino2_id, amino_symbol AS amino2_symbol, min AS min_het_amino2
    FROM distance_het_amino
    WHERE het_symbol = 'ATP' AND amino_symbol = 'G'
      AND ((min < 1.5 AND max >= 1.5) OR (min <= 6.0 AND max > 6.0)
      OR (min >= 1.5 AND max <= 6.0) OR (min < 1.5 AND max > 6.0))
  ) AS q1 NATURAL JOIN (
    SELECT * FROM (
      SELECT amino1_id AS amino4_id, amino1_symbol AS amino4_symbol, amino2_id AS amino3_id,
      amino2_symbol AS amino3_symbol, min AS min_amino4_amino3
      FROM distance_amino_amino
      WHERE ((min < 2.0 AND max >= 2.0) OR (min <= 5.5 AND max > 5.5)
      OR (min >= 2.0 AND max <= 5.5) OR (min < 2.0 AND max > 5.5))
      AND amino2_symbol = 'K'
    ) UNION
    SELECT amino2_id AS amino4_id, amino2_symbol AS amino4_symbol, amino1_id AS amino3_id,
    amino1_symbol AS amino3_symbol, min AS min_amino4_amino3
    FROM distance_amino_amino
    WHERE ((min < 2.0 AND max >= 2.0) OR (min <= 5.5 AND max > 5.5)
    OR (min >= 2.0 AND max <= 5.5) OR (min < 2.0 AND max > 5.5))
    AND amino1_symbol = 'K'
  ) AS q_43
  ) AS q2 NATURAL JOIN (
    SELECT amino1_id AS amino3_id, amino1_symbol AS amino3_symbol, amino2_id AS amino4_id,
    amino2_symbol AS amino4_symbol
    FROM next_amino_amino
    WHERE amino1_symbol = 'K'
  ) AS q3 NATURAL JOIN (
    SELECT amino1_id AS amino4_id, amino1_symbol AS amino4_symbol, amino2_id AS amino2_id,
    amino2_symbol AS amino2_symbol
    FROM next_amino_amino
    WHERE amino2_symbol = 'G'
  ) AS q4
) AS query
WHERE amino1_id <> amino2_id AND amino1_id <> amino4_id AND amino1_id <> amino3_id AND
amino2_id <> amino4_id AND amino2_id <> amino3_id AND amino4_id <> amino3_id;

```

G. Documentos para evaluación de usabilidad

Guía de uso

Guía de Uso

A continuación se describe una secuencia de pasos que le ayudarán a diseñar, buscar y observar los resultados de un patrón estructural de ejemplo con la herramienta GSP4PDB.

1. Abra su navegador e ingrese al sitio web <https://structuralbio.otalca.cl/gsp4pdb>, dedique un minuto a observar la página en toda su extensión.

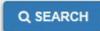
2. Diríjase a la sección "The Tool", haciendo clic en la opción  de la barra de navegación. 

3. En el panel de trabajo, haga clic en el botón , luego en el cuadro emergente ingrese **ATP** y posteriormente haga clic en . Esto agregará un ligando ATP.

4. Haga clic en el botón , luego en el cuadro seleccione **Asparagine** y posteriormente haga clic en . Esto agregará un aminoácido ASN.

5. Haga clic en el botón , luego en el cuadro seleccione **Any Negatively Charged** y posteriormente haga clic en . Esto agregará un aminoácido comodín ANY con carga negativa. Arrastre los componentes en el panel para conseguir una mejor distribución espacial.

6. Haga clic en el botón , luego sobre el ligando  y posteriormente sobre el aminoácido . Realice un doble-clic sobre el conector distancia agregado y ajuste el valor máximo a **5.0 Å**, por último haga clic en . Esto modificará el rango de distancia para revisión.

7. Haga clic en el botón , luego sobre el aminoácido  y posteriormente sobre el aminoácido . Esto agregará un conector siguiente entre los aminoácidos, el que condicionará los resultados de acuerdo a la secuencia de la cadena de aminoácidos.
8. Haga clic en el botón  para iniciar la búsqueda del patrón estructural en PDB. Luego de unos segundos, se mostrará un cuadro emergente avisando que la búsqueda ha concluido, haga clic en  para cerrarlo. En el panel de la derecha aparecerá una lista con los resultados.
- Cada resultado es una coincidencia del patrón encontrada en PDB. Cada uno posee su número de coincidencia, un enlace a la macromolécula donde se encuentra presente, un enlace para desplegar su representación tridimensional, y los valores para sus componentes y conectores en forma textual.
9. En la primera coincidencia del panel de resultados, haga clic en el botón . En el cuadro emergente se mostrará una representación del resultado en tres dimensiones por medio del visor JSmol, interactúe con él.
10. Regrese al panel de trabajo, dedique un minuto a modificar el patrón a su gusto. El botón  le ayudará a eliminar un componente haciendo clic sobre él. Con el botón  podrá limpiar el panel por completo.

¡Muchas Gracias por su Colaboración! :D

Lista de criterios

Medición de Usabilidad

La medición está basada en 30 criterios, repartidos en 6 categorías, a los cuales usted debe asignar un valor de cumplimiento de entre 1 y 5, donde:

- 1: No cumple
- 2: Cumple poco
- 3: Cumple suficiente
- 4: Cumple bastante
- 5: Cumple completamente

Los criterios siguientes aplican al **sitio web** en su totalidad:

#	Criterio	Valor
Accesibilidad		
1	El tiempo de carga del sitio web es razonable	
2	El contraste del texto con el fondo es adecuado	
3	El tamaño de letra y espaciado proporcionan fácil lectura	
4	Los iconos permiten distinguir fácilmente su significado	
Identidad		
5	El título principal deja claro el propósito del sitio web	
6	El sitio web es comprensible en 5 segundos (a grandes rasgos)	
7	La presentación de la herramienta es clara y precisa	
8	La información de referencia y contacto es mostrada claramente	
Navegación		
9	La barra de navegación se puede identificar fácilmente	
10	Los enlaces y botones son descriptivos y fáciles de identificar	
Contenido		
11	El contenido es útil y está actualizado	
12	Los estilos y colores son consistentes	
13	El énfasis (negrita, palabras a color, etc.) es usado con moderación	
14	El contenido relacionado está agrupado y bien distribuido	
15	Los ejemplos muestran en forma clara cómo usar la herramienta	

Los criterios siguientes aplican a la **herramienta** (sección *The Tool*):

#	Criterio	Valor
Contenido		
16	Las representaciones gráficas de los elementos biológicos son claras	
17	Las etiquetas de los componentes gráficos son descriptivas	
18	Los cuadros emergentes son comprensibles y llaman la atención	
19	El formato de los resultados es claro y consistente con el patrón estructural	
20	La herramienta es intuitiva, es decir, es posible usarla sin ayuda	
Control		
21	La herramienta permite diseñar un patrón estructural en forma sencilla	
22	La herramienta permite un flujo de trabajo fácil	
23	La forma de agregar y eliminar componentes del panel de trabajo es cómoda	
24	El movimiento (arrastrar, uso de flechas) de componentes es fluido	
25	Las opciones de zoom (acercar, alejar, centrar) son útiles	
26	La vista tridimensional facilita el análisis de los resultados	
Retroalimentación		
27	La herramienta guía y limita al usuario	
28	Es siempre claro lo que está pasando en la herramienta	
29	Los mensajes de error explican claramente cómo corregir un problema	
30	La información de ayuda es clara y sin ambigüedad	

¡Muchas Gracias por su Colaboración! :D

Cuestionario

Cuestionario de Retroalimentación

Este cuestionario tiene como objetivos conocer al usuario, el ambiente de prueba, y su opinión abierta respecto a la aplicación web revisada. Por favor, responda en el espacio asignado.

Presentación del usuario:

1. Nombre: _____
2. ¿En qué año va de su plan de estudios? _____
3. ¿Cuántas horas al día usa el computador? _____
4. ¿Ha tenido experiencia con aplicaciones especializadas relativas al campo de la biología estructural, biología molecular, bioquímica o similares? ¿Cuál(es)?

Sobre el ambiente:

1. Sistema operativo (ej. Windows, Linux, macOS): _____
2. Navegador web (ej. Google Chrome, Firefox, Safari): _____

Sobre la aplicación GSP4PDB:

1. ¿Agregaría algún contenido o funcionalidad extra? ¿Modificaría o eliminaría algo existente? ¿Qué cosa(s)? ¿Por qué?

2. ¿Cambiaría algo de la presentación de la interfaz (áreas de trabajo, tamaños, colores, formas, distribución, etc.) o de su interacción (clics)? ¿Qué cosa(s)? ¿Por qué?

3. Si tiene algún comentario extra que considere importante no dude en escribirlo en el espacio a continuación:

¡Muchas Gracias por su Colaboración! :D

H. Resumen evaluaciones de criterios de usabilidad

La siguiente tabla incluye la evaluación de criterios de usabilidad realizada por los estudiantes de bioinformática. De izquierda a derecha, la primera columna indica el número del criterio, la segunda muestra la descripción del criterio, de la tercera y hasta la décimo sexta columna se presentan las valoraciones de cada uno de los 14 evaluadores, por último, la columna del extremo derecho indica el promedio de las evaluaciones.

#	Criterio	1	2	3	4	5	6	7	8	9	10	11	12	13	14	\bar{X}
1	El tiempo de carga del sitio web es razonable	5	5	5	5	4	5	5	5	5	5	5	5	5	5	4,93
2	El contraste del texto con el fondo es adecuado	5	3	5	5	5	5	5	4	4	5	5	5	4	5	4,64
3	El tamaño de letra y espaciado proporcionan fácil lectura	4	5	5	5	5	5	5	5	4	5	5	5	5	5	4,86
4	Los iconos permiten distinguir fácilmente su significado	3	5	5	5	5	4	3	5	5	5	5	5	4	3	4,43
5	El título principal deja claro el propósito del sitio web	3	5	5	5	3	5	3	5	5	5	3	1	4	4	4,00
6	El sitio web es comprensible en 5 segundos (a grandes rasgos)	3	4	4	4	4	5	2	4	4	5	3	2	3	3	3,57

7	La presentación de la herramienta es clara y precisa	5	5	5	5	4	4	4	5	5	5	4	2	3	3	4,21
8	La información de referencia y contacto es mostrada claramente	5	5	5	5	5	5	5	5	5	5	5	3	3	4	4,64
9	La barra de navegación se puede identificar fácilmente	4	4	3	5	5	4	4	5	4	3	3	5	3	5	4,07
10	Los enlaces y botones son descriptivos y fáciles de identificar	5	5	5	5	5	5	3	5	5	4	3	5	3	4	4,43
11	El contenido es útil y está actualizado	5	5	5	5	5	5	4	4	5	5	4	3	5	4	4,57
12	Los estilos y colores son consistentes	5	3	5	5	5	5	4	5	4	5	4	5	4	5	4,57
13	El énfasis (negrita, palabras a color, etc.) es usado con moderación	4	5	5	5	5	4	5	5	5	5	5	4	4	5	4,71
14	El contenido relacionado está agrupado y bien distribuido	5	5	5	5	5	4	5	4	5	5	5	4	4	4	4,64
15	Los ejemplos muestran en forma clara cómo usar la herramienta	5	5	3	4	5	5	5	5	5	5	4	5	3	3	4,43
16	Las representaciones gráficas de los elementos biológicos son claras	4	5	4	5	5	5	3	5	5	5	4	1	3	3	4,07
17	Las etiquetas de los componentes gráficos son descriptivas	5	5	5	5	5	4	3	4	5	5	5	3	4	3	4,36
18	Los cuadros emergentes son comprensibles y llaman la atención	5	5	5	5	5	5	5	5	4	5	5	4	4	5	4,79
19	El formato de los resultados es claro y consistente con el patrón estructural	5	4	5	5	5	5	5	4	5	3	5	2	4	5	4,43

20	La herramienta es intuitiva, es decir, es posible usarla sin ayuda	2	3	5	4	5	4	4	5	5	5	2	2	3	2	3,64
21	La herramienta permite diseñar un patrón estructural en forma sencilla	5	5	5	5	5	5	4	5	4	5	4	2	5	4	4,50
22	La herramienta permite un flujo de trabajo fácil	5	5	5	5	5	5	4	5	5	5	5	3	5	4	4,71
23	La forma de agregar y eliminar componentes del panel de trabajo es cómoda	5	4	5	5	5	5	5	5	5	3	4	4	5	5	4,64
24	El movimiento (arrastrar, uso de flechas) de componentes es fluido	5	5	5	5	5	5	3	5	5	4	4	5	5	4	4,64
25	Las opciones de zoom (acercar, alejar, centrar) son útiles	4	5	5	5	5	4	3	5	5	5	3	5	4	5	4,50
26	La vista tridimensional facilita el análisis de los resultados	5	5	5	1	5	5	4	4	5	1	2	1	5	4	3,71
27	La herramienta guía y limita al usuario	4	4	4	3	4	4	4	5	3	1	4	3	4	4	3,64
28	Es siempre claro lo que está pasando en la herramienta	5	5	4	5	4	3	4	4	4	5	3	2	4	3	3,93
29	Los mensajes de error explican claramente cómo corregir un problema	5	5	4	3	4	3	4	5	5	3	3	3	4	2	3,79
30	La información de ayuda es clara y sin ambigüedad	5	5	4	5	5	4	4	5	5	5	3	3	4	4	4,36