Universidad de la Ciencias Informáticas

Facultad

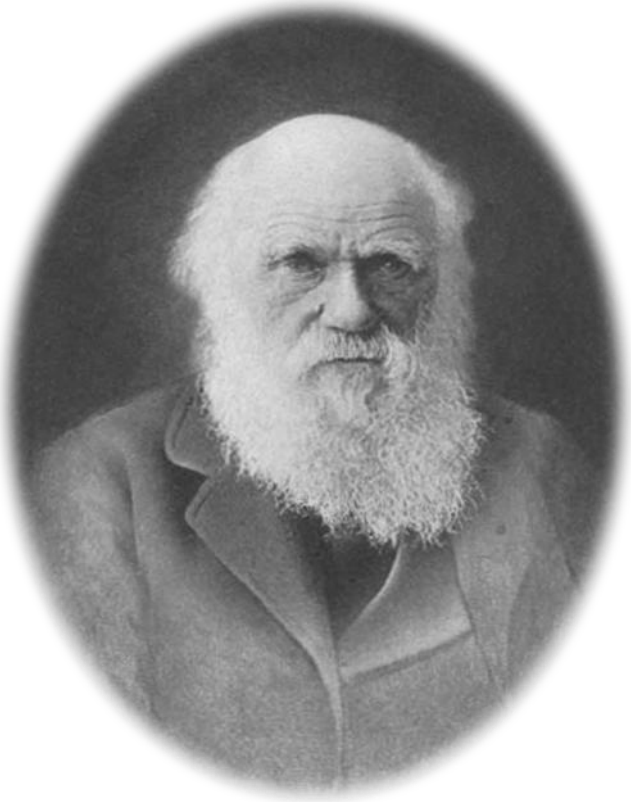
Título: Inferencia filogenética molecular en ambiente de alto procesamiento computacional.

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

Autor(es): Yudeily Ledesma Tamayo

Tutor(es): M.Sc. Orlando Martínez Pérez

La Habana, junio de 2013
"Año 55 de la Revolución"



CHARLES ROBERT DARWIN

Un naturalista inglés, creador de la teoría de la selección natural de las especies.

A los 16 años empezó a estudiar medicina. Muy pronto adquirió afición por las ciencias naturales y fue nombrado naturalista del buque británico Beagle. En 1839 apareció su primer libro: Viaje de un naturalista alrededor del mundo. En 1858 presentó, junto con Alfred Wallace, la teoría de la selección, a la que habían arribado de forma independiente. En 1859 publicó El origen de las especies por medio de la selección natural, la cual dejó sentir su influencia en los ámbitos del pensamiento.

"No es el más fuerte de las especies el que sobrevive, tampoco es el más inteligente el que sobrevive. Es aquel que es más adaptable al cambio".

Charles Darwin

Declaración de Autoría

Yo Yudeily Ledesma Tamayo declaro ser la autora del trabajo de diploma con título: “Inferencia filogenética molecular en ambiente de alto procesamiento computacional” y admito los derechos patrimoniales del mismo con carácter exclusivo a la Universidad de las Ciencias Informáticas, específicamente a la facultad 6.

Para que así conste firmo la presente a los ____ días del mes de _____ del año _____.

Yudeily Ledesma Tamayo **M.Sc. Orlando Martínez Pérez**

_____ _____

Firma del Autor Firma del Tutor

Datos de Contacto

Autor

Yudeily Ledesma Tamayo
Universidad de las Ciencias Informáticas, La Habana, Cuba
e-mail: yledesma@estudiantes.uci.cu

Tutor

M.Sc.. Orlando Martínez Pérez
Universidad de las Ciencias Informáticas, La Habana, Cuba
e-mail: orlandomp@uci.cu

Dedicatoria

Le dedico mi título, mi carrera, lo que he sido y lo que seré, a las personas más importantes de mi vida:

A mimi, papi y pipo por su apoyo incondicional y ser fuente inagotable de la única fuerza que mueve el mundo:
El Amor.

Agradecimientos

Llega el momento más difícil para mí, y es el de expresar lo agradecida que en este momento me siento con todas las personas que de una forma u otra me han dado su apoyo, comprensión y ayuda desinteresada para cumplir mi sueño, y si existe alguien que no sienta su presencia en mis palabras, reciba mis más sinceras disculpas y tenga la certeza que siempre le estaré agradecida.

Quiero agradecer:

A mimi y a papi por ser las personas que más quiero en el mundo, por brindarme su apoyo incondicional y cariño infinito en cada uno de retos que me ha puesto la vida. Por darme fuerzas, por forjarme como persona y enseñarme que en la vida uno no logra lo que no se propone. Por confiar en mí, por ser mis amigos, mis padres y mis guías. Por tenerme siempre como su "niña", por dármelo todo sin pedir nada cambio, por ser quienes sin escatimar esfuerzo alguno ha sacrificado por mí gran parte de su vida. Porque sin ustedes yo nunca hubiese logrado este sueño, por ser lo más preciado que tengo en la vida.

A pipo, mi querido y adorado hermano porque sin percatarse desde pequeña inculcó en mí los deseos de ser alguien en la vida, por sembrar en mí semillas de esperanza y seguridad para alcanzar mis metas, por lo mucho que te quiero y representas para mí. A Erialis su esposa, por darme la felicidad de tener los sobrinos más lindos del mundo Dereck y el que viene en camino.

A mis queridos abuelos maternos (Micaela y Miguel) por estar siempre pendientes de mi futuro como profesional, y a mi abuelita Nona (Mireya) que donde quiera que esté en este momento sé que estaría orgullosa de mí.

A mis segundos padres (Esperanza y Alberto) por cuidarme y quererme desde pequeña como uno más de sus hijos, a mama (Yudelkis) y a papo (Alberto) por acogerme como un miembro más de la familia.

A Elianis y Yanelis por ser para mí como las hermanas que nunca tuve, por ser compañía cuando me sentía sola y por todo el cariño que me han dado.

A mis primos, tíos, en fin, a toda mi familia en general, por ser tan especiales y por hacerme sentir privilegiada al poder contar con su amor y cariño, demostrándome que no hay nada más valioso en la

vida que la familia.

A Eduardo, mi novio, compañero y amigo, por estar siempre a mi lado, por apoyarme en cada paso que doy, por su cariño y comprensión, pero por sobre todas las cosas por quererme tal y como soy. A su familia por darme su apoyo y mostrar su preocupación.

A mi tutor Orlando Matinés Pérez por contribuir con mi formación como profesional y guiarme en el desarrollo de la tesis. A Osvel Chávez Hernández y Andry Daniel Díaz León porque a pesar de no ser mis tutores se comportaron como tal y me bridaron su ayuda incondicional cada vez que lo necesité. En especial a Andry por su paciencia, por acogerme como una más de sus tesistas y ayudarme a combatir mi estrés con los portlets.

A mis amistades de la universidad: Liuva, Yisel, Martha, Karel, Ferna, Félix y en especial a Mayde por ser mi fiel amiga, por compartir conmigo los buenos y malos momentos, por hacer suya mis preocupaciones, por su compañía, pero lo más importante por demostrarme el significado de la palabra "amistad".

Al único grupo que he tenido, el grupo 19 de la vocacional, por darme la oportunidad de conocer a mis eternos amigos: Yoli, Mary, Nery, Tay, Alfri, Alain en fin a todos, por los momentos inolvidables que pasamos en la escuela que sentó las bases para lograr lo que soy hoy.

A todos mis amistades en general, los que han pasado y los que se han quedado, quienes en todo momento me acompañaron a caminar por este sendero tan duro, pero a la vez muy grato, y porque todos ustedes han sido parte de la historia que de aquí en adelante podré contar.

A todos los profesores que aportaron algo para hacer de mí un ser útil a la sociedad. En especial a los profes de la vocacional, al maestro Miguel Torres, Osniel Toledano y María Emilia.

Y por último pero no menos importante a esta hermosa Revolución y a su máximo líder Fidel Castro por darnos la maravillosa oportunidad de ser parte de esta Universidad que ha sido casa y escuela y permitir que mi sueño se hiciera realidad.

A todos muchas gracias…

Resumen

En los últimos años las Tecnologías de la Información y las Comunicaciones (TICs) han sido las principales protagonistas de los grandes avances científicos ocurridos en las Ciencias Biológicas, lo que ha propiciado el surgimiento de la Bioinformática como nueva disciplina científica. La Inferencia Filogenética Molecular es una de las líneas de investigación que emplea la reciente disciplina para el estudio de las diferentes relaciones evolutivas de los organismos.

Para muchos centros de investigación de nuestro país realizar estimaciones filogenéticas constituye un desafío, debido al restringido acceso a los servidores internacionales que poseen y a la baja calidad de los recursos computacionales con la que disponen, teniendo en cuenta además el alto potencial de cómputo que requiere el empleo de dichos métodos para realizar análisis filogénicos. Por esta razón la presente investigación está enmarcada en proveer a la comunidad científica de un módulo de Análisis Filogenético, el cual aprovechará el clúster de computadoras Átropos del departamento de Bioinformática de la Universidad de las Ciencias Informáticas (UCI), para disminuir el tiempo de respuesta de los análisis filogenéticos, contribuyendo de este modo al desarrollo de la Bioinformática en Cuba. La confección de este módulo estuvo guiada por la metodología de desarrollo OpenUp, el lenguaje de programación Java y Eclipse como entorno de desarrollo.

Palabras claves: Bioinformática, clúster, Inferencia Filogenética Molecular, métodos filogenéticos.

Índice de Contenido

Índice de Figuras

Índice de Tablas

Introducción

El desarrollo científico ha estado unido durante los últimos años a los avances de las Tecnologías de la Información y las Comunicaciones (TICs), propiciando el surgimiento de una nueva era, la era de la información y del conocimiento. Las TICs indudablemente, han tenido un profundo impacto en todos los ámbitos de la sociedad, generando el desarrollo de diversas áreas del conocimiento, siendo el campo de las Ciencias Biológicas, uno de los más favorecidos. Los avances en esta ciencia, especialmente los avances en área de la genómica, sentaron las bases para el desarrollo del Proyecto Genoma Humano (1), en el cual se logró la revelación de la secuencia completa del genoma humano y constituye, por tanto, uno de los acontecimientos con mayor repercusión en el progreso de la ciencia en general y de la biomedicina en particular.

Estos resultados han propiciado la aparición de un enorme volumen de información experimental, que requiere ser almacenada y gestionada antes de ser procesada, para la obtención del conocimiento biológico. En este contexto, las TICs adquirieren un papel significativo y por tanto, conducen a la creación de nuevas disciplinas para dar solución a las demandas de los avances científicos, surgiendo de este modo, la Bioinformática como resultado de la intersección entre las ciencias de la vida y las TICs.

La Bioinformática es una disciplina científica emergente, que utiliza la tecnología de la información para organizar, analizar y distribuir información biológica con la finalidad de responder preguntas complejas en el área de la biología y mejorar la condición y calidad de vida del ser humano. Es un área de investigación multidisciplinaria, donde convergen varias ciencias como las matemáticas, la estadística, las ciencias computacionales y las tecnologías de la información (2). Entre las líneas de investigación más importantes de la Bioinformática se encuentra precisamente, la que se encarga de utilizar las secuencias de aminoácidos o de nucleótidos para determinar la relación evolutiva de las mismas, campo conocido como Inferencia Filogenética Molecular.

La inferencia filogenética molecular es el proceso que permite describir la relación en el tiempo entre diferentes especies, reconstruyendo la posible relación ancestro-descendiente de los organismos o sea, conformando el árbol filogenético para representar una hipótesis evolutiva. El proceso de inferencia filogenética molecular lleva consigo el análisis de una gran cantidad de hipótesis alternativas, para lo cual es importante contar con criterios suficientemente rigurosos para optimizar la selección de una o más hipótesis, preferentemente aquella que sea más plausible. Dentro de los criterios empleados en la construcción de estas filogenias se encuentran el de máxima parsimonia, evolución mínima, cuadrados mínimos, máxima verosimilitud y el bayesiano.

Precisamente estos dos últimos, a pesar que son costosos desde el punto de vista computacional, son los más utilizados por la confiabilidad de los árboles filogenéticos arrojados.

En nuestro país, existen varios centros de investigación como el Centro Nacional de Sanidad Agropecuaria (CENSA), el Centro de Sanidad Vegetal y el Instituto de Ciencia Animal (ICA), que tienen como objetivo fundamental el control y eliminación de diferentes plagas y enfermedades que pueden afectar a la nación. En este sentido utilizan los métodos filogenéticos para determinar posibles patrones de propagación y en consecuencia elaborar las medidas de control más eficaces. El restringido acceso a servidores internacionales y las pocas prestaciones que ofrecen los recursos computacionales que poseen, constituyen un desafío a la hora de estimar una confiable relación filogenética entre las especies de interés.

Tomando en cuenta que el Centro de Tecnología y Gestión de Datos (DATEC) de la Universidad de Ciencias Informáticas (UCI) dispone de una cantidad considerable de recursos computacionales, entre los que se encuentra un clúster de 8 computadoras, y que los métodos de estimación filogenética requieren de potentes recursos de cómputo, llegando a demorar hasta semanas para obtener una relación filogenética confiable, se identifica el siguiente **problema a resolver**: ¿Cómo aprovechar los recursos computacionales existentes en el Centro de Tecnología y Gestión de Datos (DATEC), para disminuir el tiempo en obtener la respuesta del análisis de inferencia filogenética molecular y hacer esto accesible a la Comunidad Científica Cubana?.

Determinando como **objeto de estudio** la inferencia computacional de relaciones filogenéticas en secuencias de ADN[1] y AA[2] que involucra como **campo de acción** la inferencia de relaciones filogenéticas moleculares determinadas por métodos bayesianos y de máxima verosimilitud en sistemas paralelos.

El **objetivo general** para dar solución a la problemática planteada es desarrollar un servicio de inferencia filogenética molecular, en la Plataforma de Servicios Bioinformáticos de la UCI.

En el presente trabajo se determinaron los siguientes **objetivos específicos**:

- Caracterizar las aplicaciones para realizar el análisis de inferencia filogenética molecular y su utilización en los sistemas de alto poder computacional.
- Realizar el análisis y diseño del servicio de inferencia filogenética molecular de la Plataforma de Servicios Bioinformáticos.

[1] ADN: Es un ácido nucleico que contiene instrucciones genéticas usadas en el desarrollo y funcionamiento de todos los organismos vivos conocidos y algunos virus, responsable de su transmisión hereditaria.
[2] AA: El aminoácido es un compuesto orgánico que se combina para formar proteínas.

- Implementar el servicio de inferencia filogenética molecular de la Plataforma de Servicios Bioinformáticos.
- Validar el correcto funcionamiento del servicio de inferencia filogenética molecular de la Plataforma de Servicios Bioinformáticos.

Para dar cumplimiento a los objetivos planteados se identificaron las siguientes **tareas investigación**:

- Análisis del estado del arte de las principales tecnologías, metodologías y herramientas a utilizar en la implementación del sistema.
- Levantamiento de requisitos funcionales del sistema.
- Definición de la arquitectura de la aplicación.
- Elaboración del diseño de interfaz del servicio.
- Implementación de los requisitos funcionales para el servicio de inferencia filogenética molecular.
- Elaboración del diseño de los casos de prueba.
- Ejecución de los casos de prueba para validar el funcionamiento del servicio de inferencia filogenética molecular.

Métodos de investigación

Los métodos de investigación constituyen el camino para llegar al conocimiento científico a partir de un conjunto de procedimientos que sirven de instrumento para alcanzar los fines de la investigación. Para el éxito de la presente investigación, se utilizan varios métodos científicos de investigación como son: el **método teórico** para estudiar las características del objeto de investigación y dentro de este el **método sistémico** para evaluar los factores que intervienen en obtener un eficiente análisis filogenético a partir del uso de sistemas de alto poder de procesamiento.

Para validar la propuesta de solución fue necesario el uso del **método empírico** y **práctico,** empleando de cada uno de ellos la **observación** y la **entrevista** respectivamente, para la recopilación de información y guía de todo el proceso de investigación.

El contenido estará estructurado de la siguiente forma:

Capítulo 1: "Fundamentación Teórica". Aborda el estudio del estado del arte de los algoritmos de Inferencia Filogenética, exponiendo las definiciones fundamentales para lograr una buena comprensión de la solución propuesta, se fundamentan las tecnologías a utilizar en el desarrollo del mismo.

Capítulo 2: "Características del Sistema". Se realiza un análisis del sistema partiendo de los conceptos relevantes en el desarrollo del módulo de Inferencia Filogenética, además se definen y describen las principales funciones que brindará el mismo.

Capítulo 3: "Diseño del sistema". Se especifica la arquitectura del sistema, así como los patrones y estilos arquitectónicos más utilizados, los cuales viabilizan la confección de los diagramas de interacción y clases del diseño.

Capítulo 4: "Implementación y pruebas". En este capítulo se confecciona el modelo de implementación para la construcción del sistema y se elaboran los casos de pruebas para comprobar la calidad de la solución.

1

FUNDAMENTACIÓN TEÓRICA

En los últimos tiempos el perfeccionamiento y desarrollo de las TICs, ha propiciado el surgimiento de la Bioinformática como ciencia emergente, la cual se auxilia de potentes herramientas computacionales con el propósito de facilitar el descubrimiento de nuevos conocimientos biológicos. En el presente capítulo se hace un estudio de los fundamentos teóricos y herramientas utilizadas en el campo de la Bioinformática para realizar análisis de inferencia filogenética molecular. Además se analizan las tecnologías a emplear para el desarrollo de la aplicación, teniendo en cuenta la arquitectura definida por el departamento de Bioinformática para el desarrollo de la Plataforma de Servicios Bioinformáticos.

1.1 Algoritmos computacionales de inferencia filogenética molecular

La inferencia de una filogenia es un proceso de estimación donde se trata de obtener la mejor hipótesis posible de una historia evolutiva, basada en la información contenida en los datos. Un análisis filogenético permite conocer la cercanía entre las especies, así como determinar cuáles de ellas son homólogas es decir, que han surgido de un ancestro común (3). Para cumplir con este fin, los métodos de inferencia filogenética siguen una de las estrategias computacionales:

1. Definiendo un algoritmo de agrupamiento que determina los pasos a seguir para la reconstrucción de la topología.
2. Definiendo un criterio de optimización que nos permita evaluar la relación entre los datos (el alineamiento de secuencias) y los árboles filogenéticos.

Los algoritmos de agrupamiento (*clustering methods* en inglés), tienen la ventaja de ser muy rápidos y fáciles de implementar, por otra parte solo producen un árbol como resultado. Esta combinación de velocidad y respuesta única son las responsables de la popularidad adquirida por los algoritmos de agrupamiento para realizar análisis filogenéticos.

El resultado que se obtiene como respuesta con el empleo de este algoritmo depende a menudo del orden en el cual se añaden las secuencias para la construcción de los árboles, siendo por tanto, una de sus principales limitaciones. Sin embargo, la mayor limitación es que no permiten evaluar hipótesis alternativas, debido a que se limitan a producir un único árbol de respuesta (4).

Entre los algoritmos de agrupamiento más usados se encuentran el Neighbor-joining y el UPGMA.
Los métodos basados en criterio de optimización (CO) asignan a cada árbol una puntuación que no es más que la función de relación entre el árbol y los datos, es decir tienen la gran ventaja de requerir una función probabilística explícita que relaciona los datos con la topología, lo que posibilita valorar la calidad de cualquier árbol, permitiendo evaluar la exactitud con la que las distintas hipótesis evolutivas en competición se ajustan a los datos. Tienen como limitación que son computacionalmente muy costosos (4). Entre los métodos de búsqueda de árboles por criterio de optimización se encuentran los métodos de máxima verosimilitud y bayesiano.

El método de máxima verosimilitud busca la topología que hace más probable el patrón de sustituciones presente en un alineamiento múltiple de secuencias moleculares, dado un modelo evolutivo explícitamente incluido por el investigador. Por otra parte el método bayesiano permite estimar la probabilidad de cada una de las topologías dado los datos, constituyendo un enfoque más cercano a la lógica cotidiana (5).

1.2 Herramientas que permiten realizar análisis de inferencia filogenética

Existen varios programas computacionales que posibilitan crear árboles filogenéticos utilizando los métodos antes descritos como el PhyML, MEGA, PAUP, MrBayes, BEAST y el RxML. El PhyML y MrBayes son dos programas de libre distribución que permiten realizar estimaciones filogenéticas mediante los métodos de máxima verosimilitud y los métodos bayesianos respectivamente.

1.2.1 MrBayes

MrBayes es un programa para la estimación filogenética mediante métodos bayesianos. La inferencia bayesiana de las filogenias está basada en la distribución de probabilidad posterior de los árboles, la que viene dada por la probabilidad de un árbol condicionada por las observaciones, entiéndase el alineamiento múltiple de secuencias (Pr(Árbol | Alineamiento)) y es a su vez determinada utilizando el teorema de Bayes (6). La distribución de probabilidad posterior de los árboles es imposible de calcular analíticamente, por lo que el MrBayes emplea la técnica de simulación conocida como Cadenas de Markov y Monte Carlo para arribar a una aproximada probabilidad posterior de los árboles.
Las características principales del programa incluyen:

- Una interfaz por línea de comandos.
- Ayuda detallada por línea de comandos.
- Habilidad para analizar datos de nucleótidos, aminoácidos, sitios de restricción y morfológicos.
- Capacidad de mezclar tipos de datos, tales como caracteres moleculares y morfológicos en un

solo análisis.

- Abundancia de modelos evolutivos para secuencias de nucleótidos, de codones y de aminoácidos.
- Habilidad para distribuir los trabajos en un clúster de computadores utilizando MPI (en inglés Message Passing Interface) únicamente en los ambientes Macintosh y UNIX (7).

1.2.2 PhyML

Es un programa cuyo principal propósito es la estimación de filogenias a partir de secuencias de nucleótidos o aminoácidos, empleando el método de máxima verosimilitud. Proporciona una amplia gama de opciones diseñadas para facilitar el análisis filogenético. La principal fortaleza de PhyML radica en el gran número de modelos de evolución que presenta así como en las varias opciones de búsquedas en el espacio de topologías de árboles filogenéticos que posee, que comprenden desde métodos eficientes y rápidos hasta enfoques más lentos pero más precisos. Implementa dos métodos para evaluar soportes de ramas en un entorno estadístico sólido (*bootstrap* no paramétrico y test de radio de verosimilitud) (8).

1.3 Plataforma de Servicios Bioinformáticos

La Plataforma de Servicios Bioinformáticos perteneciente al Centro de Tecnología y Gestión de Datos (DATEC) de la Universidad de las Ciencias Informáticas es un sistema que integra un conjunto de herramientas de uso común en la Bioinformática, así como los productos desarrollados por el departamento de Bioinformática, con el propósito de brindar servicios que puedan ser consumidos por aplicaciones o usuarios a través de un Portal Web y proporcionar acceso a los recursos computacionales existentes en la Institución.

Esta plataforma tiene una arquitectura orientada a servicios, implementados sobre la plataforma de desarrollo de aplicaciones web Java 2 Enterprise Edition (J2EE), de cara al usuario posee interfaces web en forma de portlets, los cuales se encargan de gestionar los distintos servicios que brinda la Plataforma de Servicios Bioinformáticos, y son controlados por el contenedor de portlets: Liferay Portal. La plataforma utiliza los recursos computacionales existentes en la Universidad, para aumentar las potencialidades de cómputo que la misma demanda.

1.4 Lenguaje de programación. Java

Un lenguaje de programación es un idioma artificial diseñado para expresar procesos que pueden ser llevados a cabo por máquinas como las computadoras. Los lenguajes de programación están

diseñados para describir el conjunto de acciones consecutivas que un equipo debe ejecutar, es decir es una vía práctica para que los seres humanos puedan dar instrucciones a un equipo.

El lenguaje de programación Java es un lenguaje de alto nivel, orientado a objeto, diseñado por James Gosling y publicado en 1995. Es un lenguaje ejecutado en más de 850 millones de ordenadores personales de todo el mundo y en miles de millones de dispositivos, como móviles, aparatos de televisión y consolas de juego (9). Para desarrollar sistemas en el lenguaje Java se pueden utilizar dos compiladores el de Oracle, que no es del todo libre, y el OpenJDK una versión inferior pero completamente disponible y de código abierto. Java recoge los elementos de programación que típicamente se encuentran en todos los lenguajes, permitiendo la realización de programas profesionales. Tiene muchas similitudes con los lenguajes C y C++, eliminando los complejos punteros a dirección de memoria que utilizan estos. Permite crear programas modulares y códigos reutilizables, independiente de la plataforma de desarrollo y del sistema operativo que los ejecuta.

1.5 Plataformas de Java. Java 2, Enterprise Edition (J2EE)

Java ha dado lugar a la creación de nuevos modelos y tecnologías de programación en diferentes campos, desarrollando con el paso de los años tres ediciones de plataformas diferentes, cada una de ellas destinada a cubrir un conjunto diferente de necesidades de programación. La plataforma Java 2 Estándar Edition (J2SE) enfocada en la creación de aplicaciones y *applets*, la plataforma Java 2 Micro Edition (J2ME) permite la creación de aplicaciones Java para "micro-dispositivos" y la plataforma Java 2 Enterprise Edition (J2EE) destinada a crear aplicaciones de servidor (10).

La plataforma (J2EE) especifica tanto la infraestructura para gestionar sus aplicaciones, como los servicios APIs (siglas en inglés de Aplication Programming Interface) para construir las mismas (10). J2EE está regulada por la JSR 58 (Java Specification Requests 58) y engloba dentro de sí un conjunto de especificaciones APIs relacionadas tales como JDBC (siglas en inglés de Java Database Connectivity), RMI **(**siglas en inglés de Remote Method Invocation), JMS (siglas en inglés de Java Message Service), Servicios Web, XML (siglas en inglés de eXtensible Markup Language), etc., además configura algunas especificaciones únicas como *Enterprise JavaBeans*, *servlets*, *Java Server Pages*, portlets y varias tecnologías de servicios web, que le permiten al desarrollador crear una aplicación empresarial portable entre plataformas y a la vez integrable con otras tecnologías (11).

1.5.1 Tecnologías utilizadas de la plataforma J2EE

La plataforma J2EE provee varias tecnologías que facilitan la creación de portales, incluyendo el desarrollo de interfaces de usuario, donde los portlets juegan un papel fundamental. Los portlets son

componentes web de interfaz de usuario, gestionados y visualizados en un portal web, producen lenguaje de marcado y generan contenido dinámico (12). Las JSR 168 y 286 son las especificaciones de J2EE que se encargan de proporcionar el estándar para manipular los portlets, posibilitando que estos puedan ser compatibles entre aquellas plataformas que cumplan con dichas especificaciones.
En un portlet el estado de una ventana determina la cantidad de espacio que podría asignársele al contenido generado por el mismo sobre el portal, posee tres estados (maximizado, minimizado y normal), para indicar lo que hace el usuario cuenta con tres modos diferentes (ayuda, vista y edición). Al compilar un portlet se genera un WAR (siglas en inglés de Web Application Archive) el cual puede ser trasladado y ejecutado en otro contenedor de portlets que cumpla con las JSR 168 y 286 sin necesidad de modificar el código, siendo por tanto una de las principales ventajas de utilizar los portlets.

Los encargados de manipular y ejecutar el ciclo de vida de un portlet son los contenedores de portlets (13). La Plataforma de Servicios Bioinformáticos utiliza para el desarrollo de sus aplicaciones el contenedor de portlets Liferay Portal que al mismo tiempo es gestor de contenido, ofreciendo el soporte necesario para que de forma sencilla pueda publicarse continuamente la información. Liferay es fácil de usar, gratuito y de código abierto. Incorpora funcionalidades como debates, *wikis*, calendarios, *blogs*, *chat*, foros, etc., además es capaz de adaptarse a más de 22 idiomas, debido a sus características de internacionalización (14).

Un marco de trabajo (en inglés framework*)*, es un *software* formado por componentes personalizables e intercambiables para el desarrollo de una aplicación (15). Acelerar el proceso de desarrollo, reutilizar código ya existente y promover buenas prácticas de desarrollo como el uso de patrones, son algunos de los principales objetivos que se persiguen con la utilización de un marco de trabajo. Los marcos de trabajo pueden ser empleados en muchos ámbitos del desarrollo de sistemas *software*. Posteriormente se citarán los marcos de trabajo de la plataforma J2EE, que se emplean en el desarrollo de la Plataforma de Servicios Bioinformáticos.

Apache Axis2

Apache Axis2 es un marco de trabajo de código abierto para el desarrollo de servicios Web. Creado por la Fundación de Software Apache, basado en el lenguaje Java, tanto del lado del cliente como del lado del servidor. Es compatible con los protocolos simple de acceso a objeto SOAP 1.1 y 1.2 (16).
Está diseñado para permitir la fácil adición de módulos lo que extiende sus funcionalidades de seguridad y fiabilidad. Apache Axis2 permite realizar despliegues instantáneos, lo que significa que es

capaz de agregar nuevos servicios al sistema sin tener que detener la ejecución del servidor.

Spring Framework

El desarrollo de aplicaciones J2EE se hace más fácil utilizando Spring, un marco de trabajo para el desarrollo de aplicaciones en el lenguaje de programación Java, creado en el año 2002 por Rod Johnson. Es de código abierto y por tanto no requiere licencia para su utilización, además es potente pero a la vez ligero, debido a que no requiere de muchos recursos para su ejecución. Spring se basa en el patrón MVC (Modelo Vista Controlador), estructurado en tres capas, la de interfaz de usuario, la controladora y la de acceso a datos (17). Según citan Craig Walls y Ryan Breidenbach "*Con Spring, la complejidad de tu aplicación es proporcional a la complejidad del problema que se está resolviendo*", en su libro "Spring in Action".

En Spring los encargados de procesar las peticiones de los usuarios y generar una respuesta son los controladores. El DispatcherPortlet es el controlador frontal de este marco de trabajo, el cual se encarga de asignar la petición a los controladores secundarios. Existen diferentes tipos de controladores en Spring determinados por su función, estos son algunos de los más utilizados, los controladores simples (Controller o AbstractController) y los controladores de formularios (SimpleFormController), entre otros.

Spring es modular y por tanto cuenta con una arquitectura dividida en módulos como se muestra en la (Figura 1), de tal forma que puedan ser utilizados solo aquellos que sean relevantes para el desarrollo del sistema en cuestión.

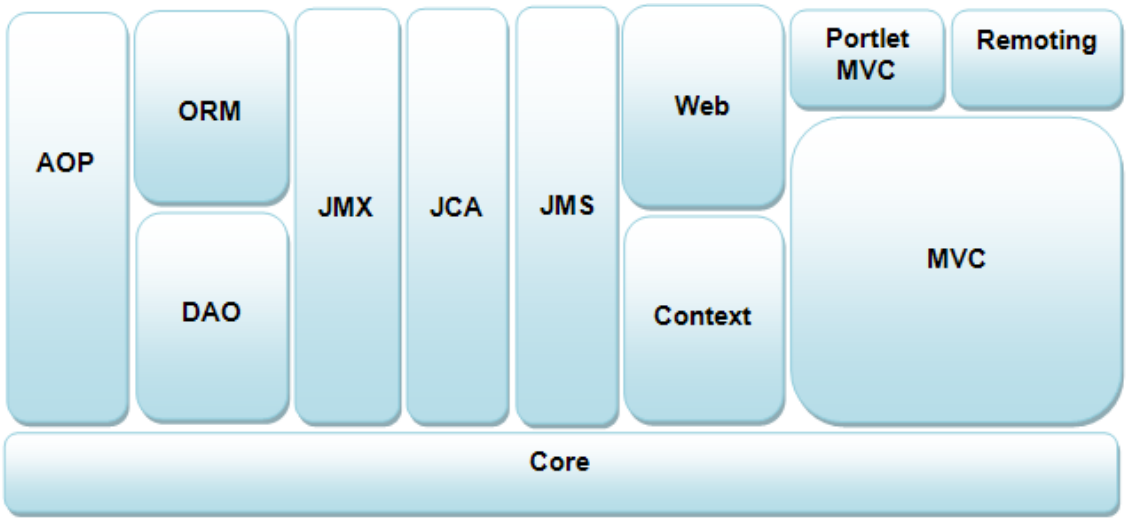


Figura 1. Arquitectura de Spring Framework

El módulo MVC como se describe anteriormente permite separar en tres capas las clases de Spring.

Los módulos DAO (siglas en inglés de Data Access Object) y ORM (siglas en inglés de Object-Relational Mapping) están creados para la manejo de las bases de datos de forma tal que se garantice la persistencia de los datos y la mayor transparencia posible entre los componentes de la aplicación y la fuente de datos. El módulo Portlets MVC facilita una infraestructura para crear portlets. Todos ellos construidos sobre el núcleo (core), el cual es la base de todo en Spring o sea es quien crea, configura y manipula los componentes.

Hibernate

Uno de los módulos que compone la arquitectura de Spring es el de mapeo objeto-relacional ORM, utilizado para la gestión de bases de datos mediante la integración con herramientas como Hibernate, el cual es un marco de trabajo que permite de una manera sencilla la representación y la conversión de datos entre una base de datos y una aplicación. Es una herramienta que posibilita a los desarrolladores manejar los datos de una base de datos sin tener conocimiento del lenguaje de consulta estructurado o SQL (por sus siglas en inglés Structured Query Language).

Hibernate es una solución ORM para manejar la persistencia de los datos. Es un marco de trabajo que facilita el manejo de la información en la base de datos, permitiendo mapear las clases de Java a las base de datos. El HQL (siglas en inglés de Hibernate Query Language) es el lenguaje que define Hibernate para conectarse con la mayoría de las bases de datos, es muy similar al SQL (17).

1.6 Recurso de alto rendimiento. Clúster

Hoy en día se necesitan computadoras muy poderosas para procesar los grandes volúmenes de información que se generan a la hora de solventar la gran variedad de problemas científicos que existen en la actualidad. Con este fin se utilizan las tecnologías computacionales de alto rendimiento, como son los clústeres.

Los clústeres son uno de los recursos disponibles para ejecutar aplicaciones que precisan gran cantidad de recursos de procesamiento. Se definen como una colección de computadoras que están interconectadas, que trabajan en conjunto distribuyéndose las tareas entre ellas, logrando que sean vistas como una sola (18). A grandes rasgos su funcionamiento se basa en dividir las tareas a ejecutar en varias partes para que cada parte pueda ejecutarse en una computadora distinta, de esta forma se logra ejecutar una tarea en forma paralela reduciendo el tiempo total de ejecución y logrando mejores precisiones en los cálculos.

El clúster Átropos del departamento de Bioinformática se utilizará como recurso de alto procesamiento para realizar los análisis de inferencia filogenética, el cual consta de 8 nodos con dos procesadores Intel(R) Core(TM)2Duo CPU E4500 2.20GHz , cada uno de ellos con 1 Gigabyte de memoria RAM.

Funciona con el sistema operativo GNU/Linux Debian Lenny 5.0 y se caracteriza por ser homogéneo, de alto rendimiento y dedicado, pues todos los nodos están disponibles para cálculo científico.

1.7 Entorno de desarrollo integrado. Eclipse

Un entorno de desarrollo integrado IDE (del inglés Integrated Development Environment) es una aplicación que proporciona una serie de servicios integrales a los programadores para facilitar el trabajo de desarrollo de *software*. La aplicación para realizar análisis de inferencia filogenética fue desarrollada empleando el Eclipse como IDE.

Eclipse es un IDE de código abierto y multiplataforma. Es un entorno de programación que ha sido empaquetado como un programa de aplicación, es decir, consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica (GUI del inglés Graphical User Interface). Este IDE provee las herramientas y funciones necesarias para el desarrollo de *software*, recogidas además en una interfaz que lo hace fácil y agradable de usar. Eclipse emplea complementos (en inglés plugins) para extender sus funcionalidades y brindar soporte a otros lenguajes de programación como C, C++, PHP y Python. La definición emitida por el proyecto Eclipse acerca de su *software* es: "*una especie de herramienta universal, un IDE abierto y extensible para todo y nada en particular"* (19)*.*

1.8 Metodología de desarrollo. OpenUp

OpenUp es una metodología de desarrollo de *software* basada en código abierto, diseñada para pequeños equipos de desarrollo, donde la mayoría de los elementos que la integran están declarados para fomentar el intercambio de información entre los mismos. Está caracterizada por ser dirigida por casos de uso, centrada en la arquitectura y se basa en el desarrollo iterativo, ágil e incremental, lo que permite mantener un enfoque centrado en el cliente así como detectar y mitigar errores a tiempo. OpenUp estructura el ciclo de vida de un proyecto en cuatro fases: concepción, elaboración, construcción y transición. Es una metodología mínima, completa y extensible, que disminuye las probabilidades de fracaso e incrementa las probabilidades de éxito de un proyecto e involucra un conjunto mínimo de prácticas que ayudan a los equipos de trabajo a ser más efectivos en el proceso de desarrollo de *software* (20).

1.9 Lenguaje de modelado. UML

El UML (del inglés Unified Modeling Language) es un lenguaje para modelar sistemas orientados a objetos, que permite a los creadores de *software* generar diseños que capturen sus ideas de una forma fácil de comprender para comunicarlas a otras personas. Un modelo UML describe lo que

supuestamente hará un sistema, pero no dice cómo implementar el mismo (21). Este lenguaje puede ser usado para modelar a través de sus diagramas distintos tipos de sistemas: sistemas de *software*, de *hardware* y organizaciones del mundo real. Minimizar los costos, mejorar la calidad de las aplicaciones y reducir sustancialmente el tiempo en el proceso de desarrollo de *software*, son algunos de los beneficios de su correcto empleo.

1.10 Herramienta CASE. Visual Paradigm para UML

Las herramientas CASE (del inglés Computer Aided Software Engineering) o Ingeniería de *Software* Asistida por Computadora en español, se define como un conjunto de programas y ayudas que brindan asistencia a los analistas, ingenieros de *software* y desarrolladores, durante todos los pasos del ciclo de vida de desarrollo de un *software* con el fin de aumentar la productividad, mejorar la calidad y reducir el costo de las mismas en términos de tiempo y dinero.

Visual Paradigm para UML es una herramienta CASE para el desarrollo de aplicaciones que utilizan el lenguaje de modelado UML, ha sido concebida para soportar el ciclo de vida completo del proceso de desarrollo del *software* a través de la representación de todo tipo de diagramas. Fue diseñada para una amplia gama de usuarios interesados en la construcción de sistemas de *software* de forma fiable a través de la utilización de un enfoque orientado a objetos. La misma propicia un conjunto de ayudas para el desarrollo de programas informáticos, desde la planificación, pasando por el análisis y el diseño, hasta la generación del código fuente de los programas y la documentación.

Esta herramienta permite aumentar la calidad del *software*, a través de la mejora de la productividad en el desarrollo y mantenimiento del mismo, así como su reutilización, portabilidad y estandarización de la documentación, además del uso de las distintas metodologías propias de la Ingeniería del Software (22).

1.11 Sistema Gestor de Bases de Datos. PostgreSQL

Un sistema gestor de bases de datos o DBMS (siglas del inglés Data Base Management System), es una colección de programas que permiten crear y mantener bases de datos para responder a las necesidades de una institución (23). Existen varios DBMS entre los que se encuentran: MySQL, MAGIC, WindowBase, y PostgreSQL.

PostgreSQL es un sistema de gestión de bases de datos objeto-relacional, distribuido bajo licencia BSD (siglas del inglés Berkeley Software Distribution) y con su código fuente disponible libremente. Es un sistema de gestión de bases de datos de código abierto que funciona muy bien con grandes cantidades de datos y con una alta concurrencia de usuarios accediendo a la vez al sistema.

PostgreSQL utiliza un modelo cliente/servidor y usa multiprocesos para garantizar la estabilidad del sistema (24). Está implementado para que se ejecute en varios sistemas operativos y permite utilizar numerosos tipos de datos, así como definir los propios**.** Se caracteriza por tener una buena escalabilidad, ya que es capaz de ajustarse al número de ordenadores y a la cantidad de memoria disponible, además cuenta con un robusto sistema de seguridad mediante la gestión de usuarios, grupos de usuarios y contraseñas (25).

1.12 Conclusiones parciales del capítulo

La confiabilidad de las estimaciones filogenéticas depende en gran medida de los criterios que sean empleados para conocer la historia evolutiva de determinada especie. Por esta razón en este capítulo se analizaron los principales criterios para realizar una inferencia filogenética, seleccionando entre todos los de máxima verosimilitud y bayesiano, a través del empleo de los programas PhyML y MrBayes respectivamente para el desarrollo del servicio web de análisis filogenético.

Se seleccionó el lenguaje de programación Java y el entorno de desarrollo integrado Eclipse Helios 3.6 para implementar el servicio web de análisis de inferencia filogenética. La plataforma J2EE posibilitará el uso de los portlets como componente de interfaz de usuario y Liferay Portlets 6.0.5 como gestor de contenido y contenedor de portlets. Se seleccionó Hibernate 3.3.1 como herramienta de mapeo objeto – relacional (ORM), Spring Framework 3.0.4 como marco de trabajo y Apache Axis2 1.6.1 para la implementación del servicio web. Se definió el uso de un clúster de 8 nodos del departamento de Bioinformática para agilizar el proceso de obtención de respuestas en el servicio de análisis filogenético. Se utilizará el gestor de bases de datos PotgreSQL 9.1 y como herramienta de modelado Visual Paradigm 8.0 utilizando el lenguaje UML. La metodología OpenUp guiará todo el proceso de desarrollo del sistema.

2

CARACTERÍSTICAS DEL SISTEMA

En el presente capítulo se realiza un análisis de la solución propuesta, para ello se representan a partir del modelo de dominio los principales objetos presentes en el dominio del problema; para logar un mayor entendimiento del sistema, se determinan las capacidades o funciones que debe cumplir (requisitos funcionales) y las propiedades o cualidades que el producto debe tener (requisitos no funcionales). También se describen los procesos en forma de casos de uso que dan cumplimiento a los requisitos funcionales.

2.1 Modelo de Dominio

Un modelo del dominio es una representación de las clases conceptuales del mundo real, es la descomposición de un dominio de interés en clases conceptuales individuales u objetos. A continuación se presenta el modelo de dominio correspondiente al módulo de Inferencia Filogenética de la Plataforma de Servicios Bioinformáticos.

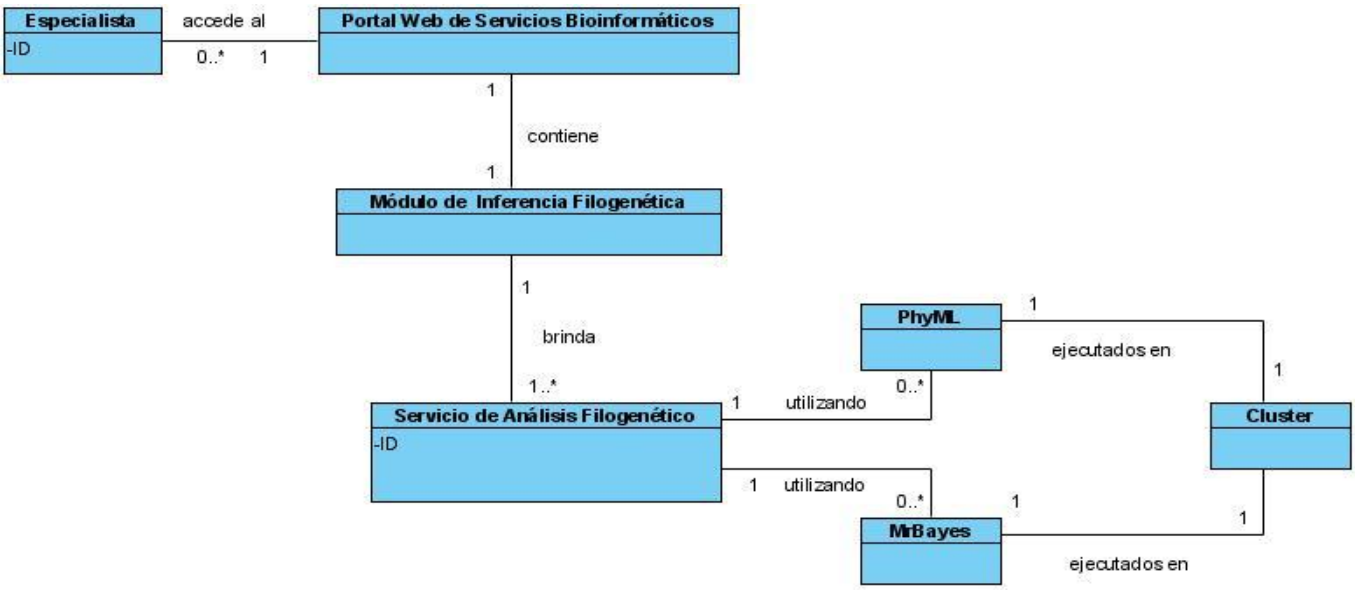


Figura 2. Modelo de Dominio del servicio de inferencia filogenética.

Un modelo de domino, podría considerarse como un diccionario visual, donde resulta fácil entender los distintos elementos importantes del dominio y sus relaciones, es una representación de las cosas del mundo real del dominio de interés, no de componentes *software* (26).

Para confeccionar un correcto modelo de dominio es indispensable identificar los diferentes conceptos u objetos presentes en el dominio del problema. A continuación se muestran y describen los objetos del dominio determinados para el módulo de Inferencia Filogenética.

- **Especialista:** Es la persona que interactuará con el sistema para realizar estudios en el campo de la Bioinformática.
- **Portal de Servicios Bioinformáticos:** Mostrará todos los servicios que engloba la Plataforma de Servicios Bioinformáticos.
- **Módulo de Inferencia Filogenética**: Módulo que contendrá todos los servicios para realizar análisis de inferencia filogenética.
- **Servicio de Análisis Filogenético:** Posibilitará a los especialistas realizar análisis filogenético utilizando los programas PhyML y MrBayes.
- **PhyML:** Programa que permite realizar análisis filogenéticos de secuencias.
- **MrBayes:** Programa que permite realizar análisis filogenéticos de secuencias.
- **Clúster:** Recurso de alto poder de procesamiento empleado para realizar los análisis filogenéticos.

2.2 Requerimientos de software

Los requisitos de *software* son las condiciones o capacidades que debe cumplir o poseer un sistema para sufragar las necesidades del cliente. A continuación se muestran según su tipo, los requisitos funcionales y no funcionales de la solución propuesta.

2.2.1 Requisitos Funcionales

Los requisitos funcionales **(RF)** denotan las funcionalidades y servicios que el sistema debe proporcionar, es decir no son más que las capacidades o condiciones que el sistema debe cumplir.

RF1: Realizar análisis de inferencia filogenética.
RF2: Administrar resultado de análisis filogenético.
RF2.1: Eliminar ficheros de repuesta del análisis filogenético.
RF2.2: Listar resultado del análisis filogenético.
RF3: Descargar resultado del análisis filogenético.

2.2.2 Requisitos no funcionales

Los requisitos no funcionales **(RNF)** son propiedades o cualidades que el producto debe tener, entiéndase por propiedades las características que hacen al producto atractivo, usable, rápido o confiable. Regularmente están vinculados a requisitos funcionales, es decir una vez se conozca lo que el sistema debe hacer, se puede determinar cómo ha de comportarse, qué cualidades debe tener.

Los RNF son importantes para que clientes y usuarios puedan valorar las características no funcionales del producto, por tanto en muchos casos son fundamentales en el éxito del producto. A continuación se listan los RNF particulares del módulo de Inferencia Filogenética.

Rendimiento

El sistema garantizará la rapidez y eficiencia del sistema empleando un clúster de 8 nodos el cual acortará considerablemente el tiempo de respuesta del servicio de inferencia filogenética.

Usabilidad

El servicio de análisis filogenético tendrá un ambiente sencillo y fácil de usar, para que pueda ser utilizado por aquellos usuarios que posean conocimientos básicos en el campo de la bioinformática y en el uso de aplicaciones web.

Apariencia o interfaz externa

El sistema contendrá una interfaz agradable, de forma tal que el especialista se sienta a gusto en él y pueda explotar al máximo sus funcionalidades para agilizar su trabajo.

Software

En el clúster deberá estar instalado:

- MrBayes 3.2.1: Permite realizar análisis filogenéticos utilizando los métodos bayesianos.
- PhyML 3.0: Permite realizar análisis filogenéticos utilizando los métodos de máxima verosimilitud.

En la PC del cliente deberá estar instalado:

- Herramienta para descomprimir archivos en formato ZIP.

Restricciones en el diseño y la implementación

- Se utilizará el estándar de codificación de Java.
- El acceso a la base de datos se garantizará a través del patrón Objeto de Acceso a Datos.
- Se empleará Hibernate para el manejo de las bases de datos.

2.3 Casos de Uso del Sistema

Los diagramas de casos de uso (CU) documentan el comportamiento de un sistema desde el punto de vista del usuario, representando las funciones que un sistema puede ejecutar. Su ventaja principal es la facilidad para interpretarlos, lo que hace que sean especialmente útiles en la comunicación con el cliente. El modelo de CU del sistema de inferencia filogenética se muestra en la Figura 3.

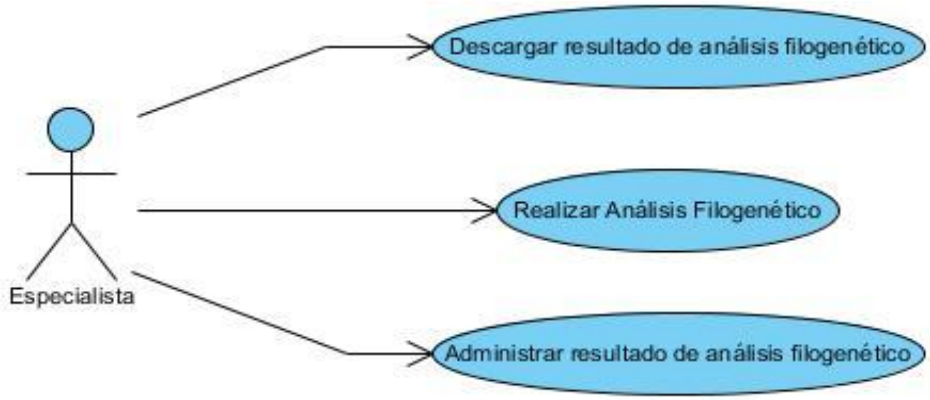


Figura 3. Diagrama de Casos de Uso del sistema.

2.4 Descripción de Casos de Uso

La especificación de los casos de uso hace referencia a la descripción de cada una de las partes antes definidas para lograr un mayor entendimiento del funcionamiento de las mismas. Muestran detalladamente cómo funciona el sistema ante una acción ejercida por el actor, en este caso el especialista. A continuación se describirá el CU “Realizar Análisis Filogenético”, por ser el más significativo desde el punto de vista arquitectónico. Para consultar las restantes descripciones de casos de uso remitirse al anexo 1.

Tabla 1. Descripción del CU “Realizar Análisis Filogenético”.

Objetivo	Este caso de uso posibilita realizar análisis filogenético a secuencias moleculares.
Actores	Especialista
Resumen	El caso de uso se inicia cuando el especialista selecciona la opción “Realizar análisis de inferencia filogenética”. Termina cuando el sistema muestra las ejecuciones realizadas por el especialista.
Complejidad	Alta.
Prioridad	Crítico.
Precondiciones	El especialista debe estar autenticado en el sistema.
Poscondiciones	El sistema realiza al análisis filogenético con el programa seleccionado y muestra las ejecuciones realizadas.

Flujo Normal de Eventos		
	Actor	**Sistema**
1.	Entra al sistema y selecciona la opción “Realizar análisis de inferencia filogenética” del módulo Inferencia Filogenética.	
2.		Muestra una página donde se encuentran los diferentes programas para realizar el análisis filogenético. • PhyML • MrBayes
3.	Escoge el programa por el cual quiere realizar el análisis filogenético.	
4.		Si escoge la opción PhyML ver la sección “Análisis filogenético con PhyML”. Si escoge la opción MrBayes ver la sección “Análisis filogenético con MrBayes”.
5.		Termina así el caso de uso “Realizar Análisis Filogenético”.
Sección 1: “Análisis filogenético con PhyML”.		
Flujo básico “Análisis filogenético con PhyML”.		
	Actor	**Sistema**
1.		Muestra un formulario donde se recogen todos los parámetros necesarios para realizar el análisis.
	Llena los campos Sequence file name, data type, sequential format, data sets, bootstrap, model name, frequencies, ts/tv_ratio, proportion_ invariable, gamma, parameter optimization, num_rand_starts.	
2.		Valida que los datos entrados sean correctos. Llama a la funcionalidad `RealizarAnálisis FilogenéticoConPhyMLEnCluster` del servicio web, el cual envía los datos al clúster para realizar dicho análisis. Guarda en la base de datos el análisis filogenético y muestra una página donde se listan todas la ejecuciones realizadas por el especialista.
Flujos alternos		
Evento 3. “Datos incorrectos”		
	Actor	**Sistema**

1	El especialista entra datos incorrectos	
2		3.1 Muestra un mensaje de error y señala los campos incorrectos. El sistema retorna al paso 1 de la sección “Análisis filogenético con PhyML”.
Sección 2: “Análisis filogenético con MrBayes”.		
Flujo básico Análisis filogenético con MrBayes”.		
	Actor	**Sistema**
1		Muestra un formulario que permite entrar los ficheros necesarios para realizar el análisis.
2	Agrega los ficheros correspondientes en cada campo • fichero de comandos • alineamiento de secuencias	
3		Valida que los datos entrados sean correctos. Llama a la funcionalidad `RealizarAnálisis FilogenéticoConMrBayesEnCluster` del servicio web, el cual envía los datos al clúster para realizar dicho análisis. Guarda en la base de datos el análisis filogenético realizado y muestra una página donde se muestran todas la ejecuciones realizadas por el especialista.
Flujos alternos		
Evento 3. “Datos incorrectos”		
1	El especialista entra datos incorrectos	
2		3.1 Muestra un mensaje de error y señala los campos incorrectos. El sistema retorna al paso 1 de la sección “Análisis filogenético con PhyML”.
Requisitos funcionales	RF1	

2.5 Conclusiones parciales

Durante el desarrollo de este capítulo se identificaron a partir del modelo de dominio las clases conceptuales significativas que intervienen en el desarrollo del módulo de Inferencia Filogenética. Se determinaron las restricciones con las cuales debe cumplir el mismo, así como las funciones que brindará para que los especialistas realicen sus respectivos estudios científicos. Se construyó el diagrama de casos de uso y se realizó una descripción detallada del CU ”Realizar Análisis Filogenético” por ser el de mayor impacto en la solución del problema.

3

DISEÑO DEL SISTEMA

Para tener una idea más completa sobre las funcionalidades que brindará el sistema, en este capítulo se procederá a realizar el diseño del mismo, en el cual se especificará la arquitectura que tendrá la aplicación y se confeccionarán los diagramas de interacción y de clases del diseño necesarios para garantizar su correcta implementación, estableciendo de este modo las bases para el progreso de las posteriores fases de desarrollo del *software*.

3.1 Arquitectura del software

Hoy en día la implementación de sistemas complejos exige a los desarrolladores de *software* diseñar muy cuidadosamente la arquitectura bajo la cual funcionarán sus sistemas, ya que las decisiones que se tomen en cuanto a este tema pueden tener gran influencia a lo largo de todo el ciclo de vida de la aplicación. El término “arquitectura de *software*”, actualmente consta de muchas definiciones, pero en su sentido más amplio se puede decir que abarca lo relativo a la estructura del sistema, su organización en subsistemas y la relación entre ellos (27).

3.1.1 Estilos y Patrones arquitectónicos

Definir una correcta arquitectura de *software* conlleva a la selección de los estilos y patrones arquitectónicos, los cuales ayudan a definir la composición y el comportamiento del *software*, la combinación adecuada de ambos permite alcanzar los requerimientos de calidad esperados. El módulo de Inferencia Filogenética presenta los patrones arquitectónicos Modelo Vista Controlador (MVC), Arquitectura Orientada a Servicios (SOA) y Cliente-Servidor.

Arquitectura Orientada a Servicios

El módulo de Inferencia Filogenética Molecular emplea como patrón una arquitectura orientada a servicios (SOA por sus siglas en inglés). Dicha arquitectura se caracteriza por utilizar el protocolo SOAP, WSDL (siglas en inglés de Web Services Description Language) como lenguaje para la descripción de los servicios y UDDI (siglas en inglés de Universal Description Discovery and Integration) para la publicación o registro de los mismos. La figura que se muestra a continuación representa de forma más clara la estructura de SOA.

Figura 4. Estructura de la arquitectura SOA.

SOA propone tres componentes básicos, cada uno con funcionalidades específicas:

- **Cliente del servicio web:** Es quien solicita la ejecución del servicio web y por tanto el que lo consume.
- **Proveedor de servicio:** Es el encargado de implementarlo y ofrecerlo a los clientes.
- **Registro del servicio:** Es un repositorio donde se almacenan las descripciones de los servicios para que así los clientes puedan buscar el servicio web que mejor se adapte a sus necesidades.

Modelo Vista Controlador

El desarrollo de los portlets fue basado en el patrón arquitectónico Modelo Vista Controlador (MVC), el cual separa los datos, la interfaz de usuario, y la lógica de la aplicación en tres componentes distintos:

- **Modelo:** Esta es la representación específica de los datos con los cuales el sistema opera.
- **Vista:** Este presenta el modelo en un formato adecuado para interactuar, usualmente la interfaz de usuario.
- **Controlador:** Este responde a eventos, usualmente acciones del usuario e invoca cambios en el modelo y probablemente en las vistas.

Arquitectura Cliente-Servidor

La arquitectura cliente-servidor define dos tipos de entidades diferenciadas que se responsabilizan de acciones diferentes: cliente y servidor.

- **Cliente:** Inicia el diálogo mediante el envío de peticiones.
- **Servidor:** Presta el servicio y responde las peticiones recibidas.

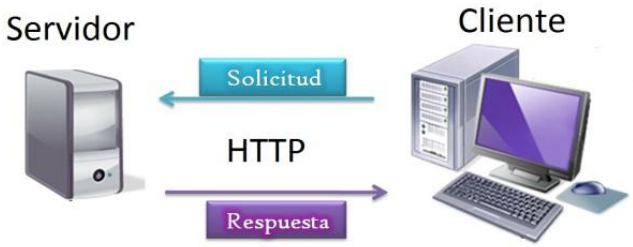


Figura 5. Arquitectura Cliente-Servidor.

Como se muestra en la figura anterior en esta arquitectura el cliente envía un mensaje solicitando un determinado servicio a través de un navegador web, el servidor web atiende las peticiones que el cliente realiza y responde en formato HTML (siglas en inglés de Hyper Text Markup Language); el navegador muestra al usuario los elementos que componen la página.

3.1.2 Patrones de diseño

En la terminología de objetos, el patrón es una descripción de un problema y su solución, que recibe un nombre y que puede emplearse en otros contextos; en teoría, indica la manera de utilizarlo en circunstancias diversas. Muchos patrones ofrecen orientación sobre como asignar las responsabilidades a los objetos ante determinada categoría de problemas (28).

Dentro de los patrones de diseño se encuentran los patrones de asignación de responsabilidades o patrones GRASP, los cuales describen los principios fundamentales de la asignación de responsabilidades a objetos, expresados en forma de patrones. A continuación se explican los patrones utilizados en la confección del sistema.

Alta Cohesión

La cohesión es una medida de cuan relacionadas y enfocadas están las responsabilidades de una clase. Una clase con alta cohesión tiene responsabilidades moderadas en un área y colabora con las otras para realizar las tareas, son fáciles de dar mantenimiento, entender y reutilizar (28).

Bajo Acoplamiento

El acoplamiento es una medida de la fuerza con que las clases están conectadas a otras clases. Una clase de bajo o débil acoplamiento no depende de muchas otras (28).

El sistema posee un bajo acoplamiento entre las vistas y los controladores, ya que estos no conocen el controlador encargado de procesar su información. El controlador-frontal o Distpacher se encarga de mantener una alta cohesión entre vistas y controladores delegando funcionalidades en dependencia de la petición generada por el usuario.

Experto

El patrón experto asigna una responsabilidad al experto en información: la clase que cuenta con la información necesaria para cumplir la responsabilidad. Si se implementa de forma adecuada, los sistemas tienden a ser más fáciles de entender, mantener y ampliar, y se presenta la oportunidad de reutilizar los componentes en futuras aplicaciones (28). Este patrón se empleará en la solución propuesta mediante el Distpacher, ya que Spring trata de aglutinar toda la responsabilidad e información en el mismo.

Creador

El patrón creador explica que clase es la encargada de crear objetos, en determinados escenarios de ejecución. Se encarga de asignarle a la clase B la responsabilidad de crear una instancia de la clase A. B es un creador de los objetos A (28). Este patrón es aplicado por el Distpacher, pues tomando en cuenta lo mencionado anteriormente en el patrón Experto, esta tiene toda la información necesaria para crear las instancias de los objetos mediante el *Bean Factory* y pasar las dependencias mediante la inversión de control, manifestándose así el patrón inyección de dependencia que implementa el marco de trabajo Spring.

Controlador

El patrón controlador asigna una responsabilidad de recibir o manejar un mensaje de evento del sistema (28). Este se evidencia en cada uno de los controladores que se encargan de procesar una petición en particular. Esta tarea le es asignada a dichos controladores por el controlador-frontal, representando un sistema global.

Los patrones GoF[3] también forman parte de los patrones de diseño y proponen soluciones para diferentes clases de problemas en el desarrollo del *software*. Algunos de los patrones GoF utilizados en el sistema se explican brevemente a continuación.

Fachada

El patrón fachada define una clase con una interfaz común a un grupo de componente o a un conjunto heterogéneo de interfaces. Los elementos heterogéneos pueden ser las clases de un paquete, un conjunto de funciones, un esquema o un subsistema (28).

En la solución ofrecida será definida una interfaz común para interactuar con la base de datos, asignándole la responsabilidad de ejecutar todas las consultas, esta interfaz queda representada en la clase `DAOImplementado.java` donde se refleja dicho patrón.

Agente Remoto

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

El patrón agente remoto de la pandilla de los cuatro sugiere crear una clase de *software* local que represente el servicio externo y asignarle la responsabilidad de contactar el servicio real (28). Este patrón se aplica mediante la clase `PortTypeProxy.java` la cual permite que los controladores puedan acceder a las funcionalidades que brinda el servicio web.

3.2 Diagramas de Interacción

Con el propósito de lograr una eficiente implementación de las funcionalidades del sistema se confeccionan los diagramas de interacción, que no son más que diagramas que describen cómo grupos de objetos colaboran para conseguir algún fin, a partir de los cuales se pueden identificar las clases de *software* que intervienen en la solución y sus métodos.

UML define dos tipos de estos diagramas, los de colaboración e interacción. Los diagramas de colaboración describen las interacciones entre los objetos en un formato de red o grafo, mientras que los de secuencia lo hacen a partir de una especie de cerca o muro, siendo ambos uno de los principales artefactos que se generan en el análisis y el diseño orientado a objeto (28). Las figuras que se muestran a continuación muestran los diagramas de interacción del caso de uso más significativo en el desarrollo de la aplicación. En el anexo 2 se encuentran los diagramas de interacción de los restantes casos de uso.

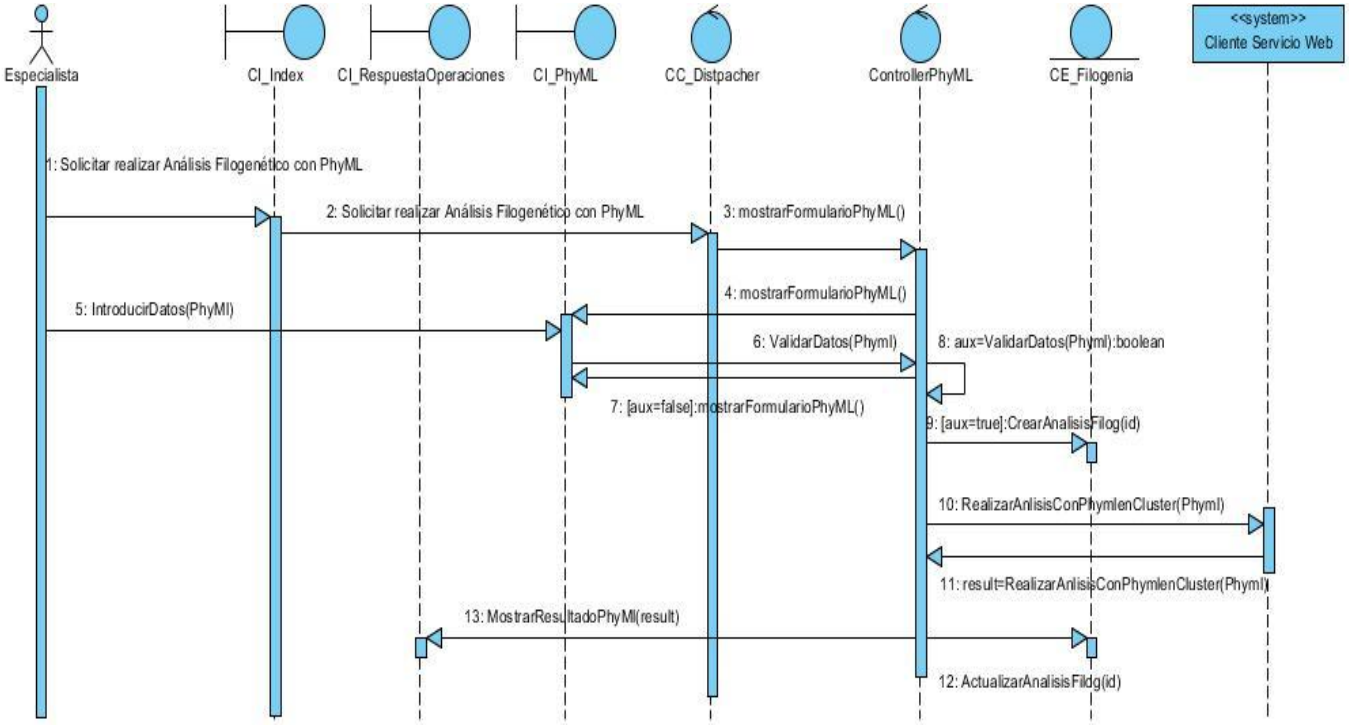


Figura 6. Diagrama de secuencia correspondiente al CU "Realizar Análisis Filogenético", sesión "Análisis filogenético con PhyML"**.**

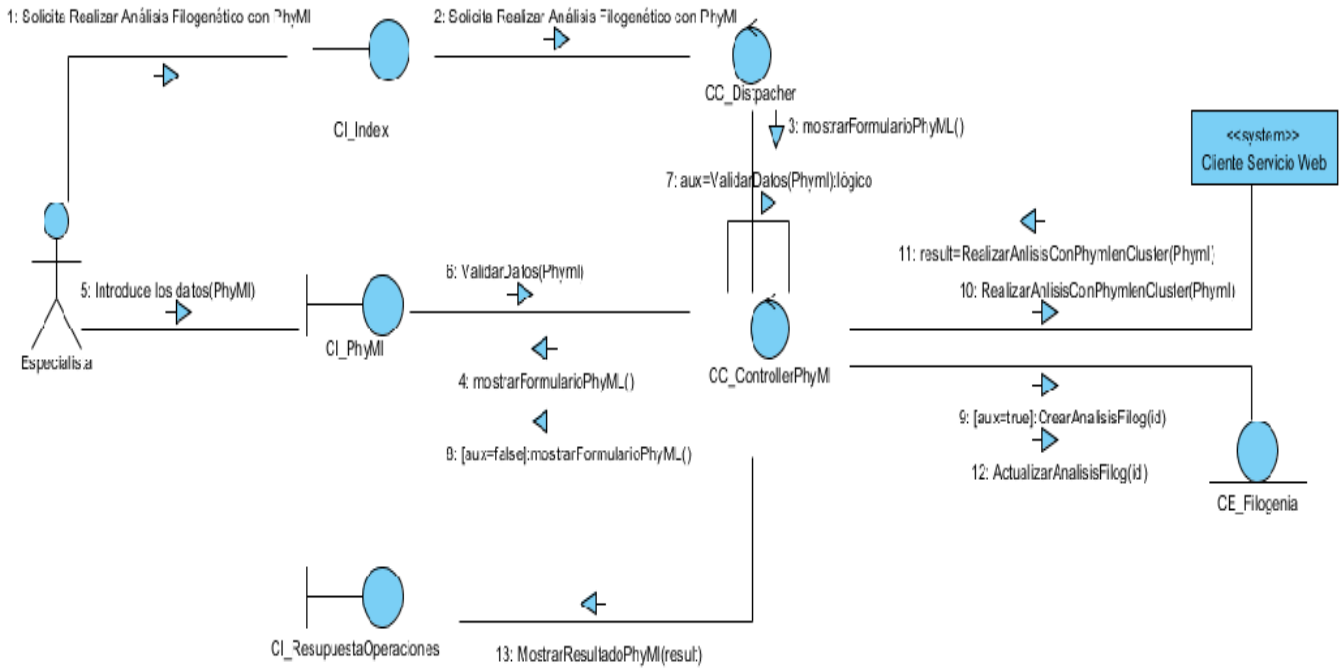


Figura 7. Diagrama de colaboración correspondiente al CU “Realizar Análisis Filogenético”, sesión *“*Análisis filogenético con PhyML”.

Como evidencian los diagramas de interacción, para realizar un análisis filogenético utilizando PhyML el especialista accede desde la página principal (CI_Index) a la página de PhyML (CI_PhyML), para introducir los parámetros necesarios para realizar el análisis con dicha herramienta. El controlador frontal (CC_Distpacher) actúa como intermediario para definir el controlador que le dará respuesta a la petición efectuada, en este caso el controlador (CC_controllerPhyML). Una vez ejecutado el servicio web y almacenado los datos se muestran las operaciones realizadas en la página de respuesta (CI_ResultadoOperaciones).

3.3 Modelo de Diseño

Para lograr la especificación de las clases del *software* y de las interfaces de la aplicación es importante la confección de los diagramas de clases de diseño, el cual contiene las definiciones de las entidades del *software* y no conceptos del mundo real. A continuación se muestra el diagrama de diseño para el caso de uso “Realizar Análisis Filogenético”. En el anexo 3 se encuentran los diagramas de diseño de los restantes casos de uso.

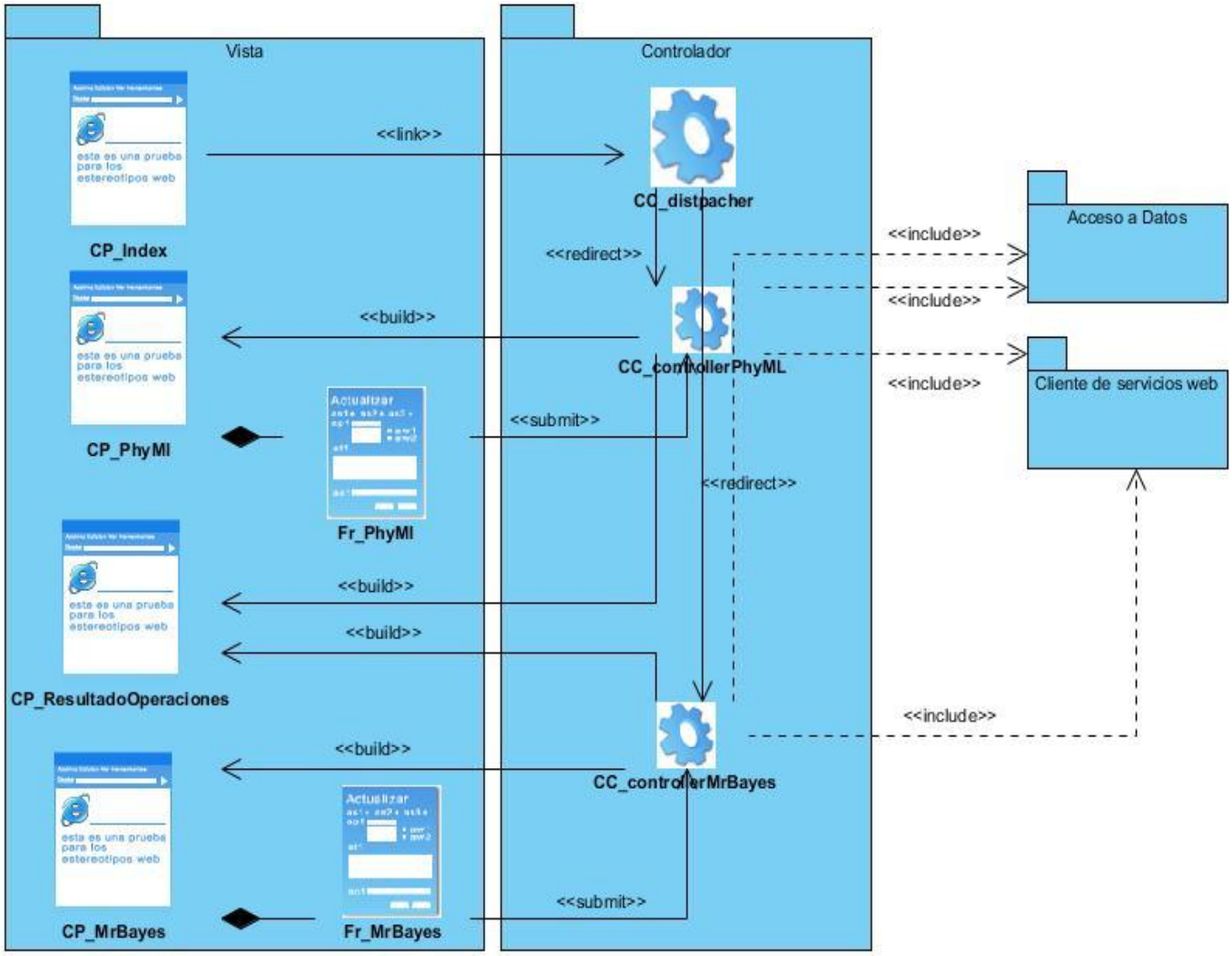


Figura 8. Diagrama de clases del diseño para el CU “Realizar Análisis Filogenético”.

En correspondencia con las capas del patrón arquitectónico MVC, se identifican tres paquetes funcionales: Vista, donde se encuentran las páginas JSP que se muestran con la respuesta del sistema; en el Controlador se gestionan todas las peticiones hechas por los usuarios, y por último el Objeto de Acceso a Datos.

En el paquete Acceso a Datos (Figura 9), se ubican todas las clases necesarias para la gestión de la base de datos, función que realiza a partir del empleo de la interfaz `DAO.java`, la clase `DAOImplementado.java` implementa todos los métodos especificados en la clase interfaz y utiliza las clases `EspecialistaHome.java`, `OperacionHome.java` y `ArchivoHome.java` para acceder a las tablas del modelo, representado el acceso a datos para los demás componentes del sistema.

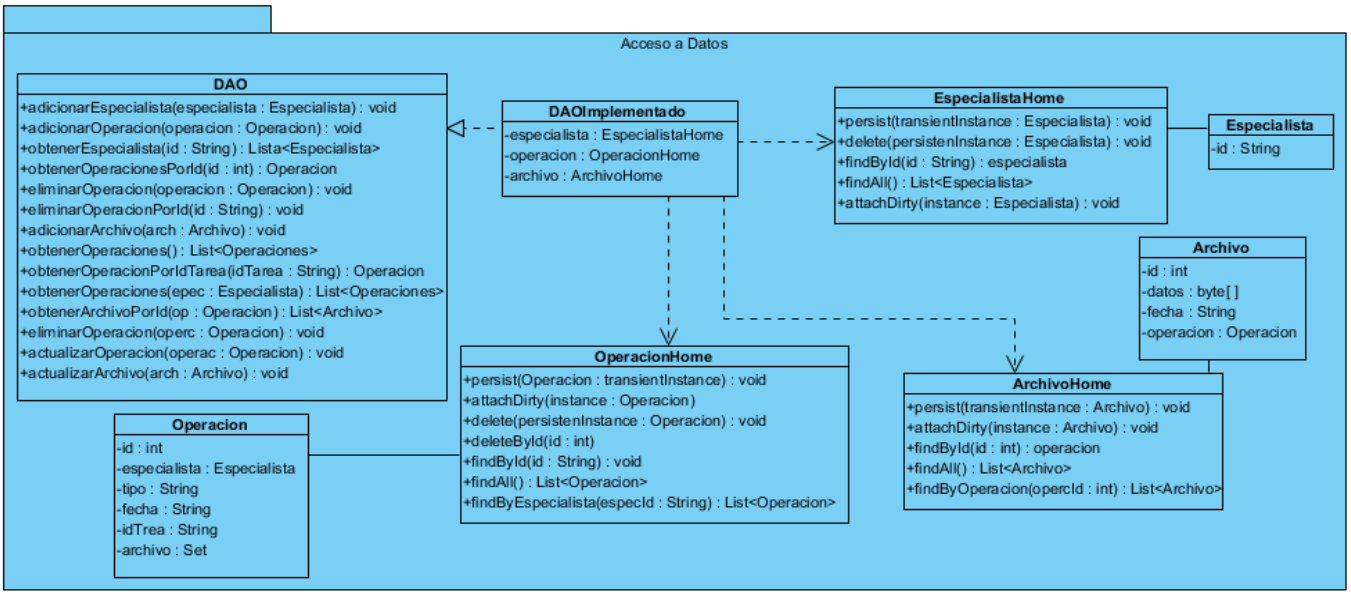


Figura 9. Paquete de Acceso a Datos del diagrama de clases del diseño

El paquete Cliente de Servicio Web contiene las clases necesarias para lograr la interacción con las funcionalidades del servicio web, con la clase `BiosoftWSPortType.java` se declaran las funciones que invocan a los servicios, mientras que la clase `BiosoftWSPortTypeProxy.java` se encarga de definirlas, en este caso requiere de la clase `PhyMl.java`, la cual contiene la información necesaria para cumplir con lo especificado en la sesión “Análisis filogenético con PhyML” del CU “Realizar Análisis Filogenético”. En el paquete servicio web es donde se encuentran todos los servicios que serán utilizados, los cuales se auxilian de una biblioteca para acceder a las funcionalidades del clúster.

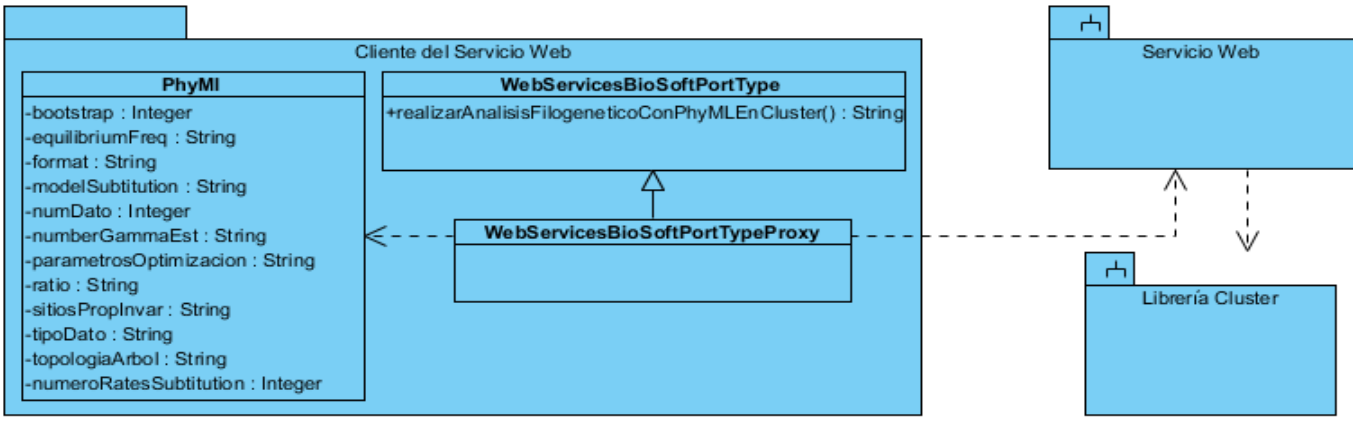


Figura 10. Paquete correspondiente al cliente del servicio web.

3.4 Modelo de Datos

Para lograr almacenar y gestionar la información del servicio de Inferencia Filogenética Molecular fue

necesario el diseño de una base de datos, lo cual se logró a partir de la confección de modelo de datos que a continuación se muestra:

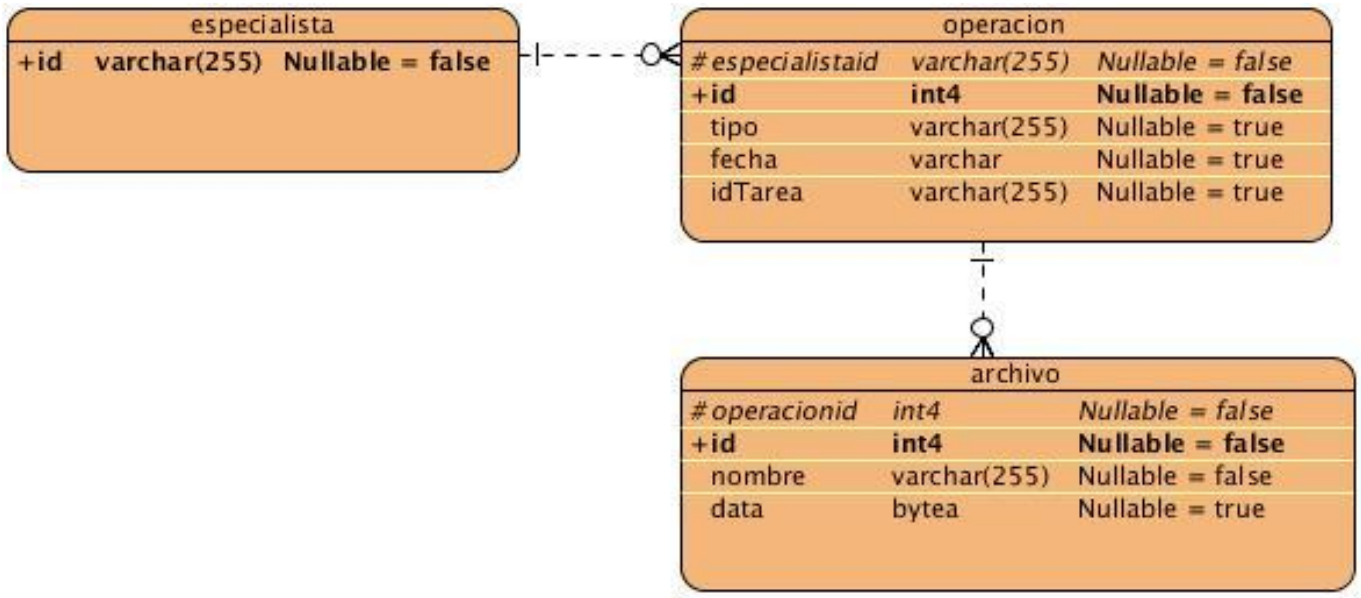


Figura 11. Modelo de Datos del servicio de Análisis Filogenético Molecular.

Este modelo de datos está compuesto por tres tablas, donde cada una de ellas contiene su propio identificador. La tabla especialista guarda la información necesaria de los especialistas que acceden al sistema, la tabla archivo que es la encargada de guardar los archivos que generan como respuesta los diferentes análisis que se realizan en el sistema y la tabla operación se describe detalladamente a continuación.

Tabla 2. Descripción de la tabla operación del modelo de datos.

Nombre: Operación		
Descripción: Almacena la información de las operaciones que realiza cada especialista en el sistema		
Atributo	**Tipo**	**Descripción**
especialistaid	Varchar(255)	Identificador del especialista
id	int	Identificador de la operación
tipo	Varchar(255)	Tipo de operación
fechaCreacion	Varchar(255)	Fecha de creación de la operación
id_tarea	Varchar(255)	Identificador de la tarea que está en ejecución

3.5 Conclusiones parciales

En el presente capítulo fue definida la arquitectura del módulo de Inferencia Filogenética Molecular como uno de los aspectos básicos de la etapa de diseño de *software*, que permite garantizar la calidad y funcionalidad del sistema, además se confeccionaron los diagramas de diseño e interacción correspondientes a esta fase. Con el propósito de indicar la mejor forma de modelar los objetos del sistema se identificaron los patrones diseño: creador, controlador, alta cohesión, bajo acoplamiento de la familia GRASP y los patrones fachada y agente remoto de los GoF. Se confeccionó el modelo de datos para entender mejor las relaciones entre las clases persistentes de la base de datos.

4

IMPLEMENTACIÓN Y PRUEBA

Basado en los resultados obtenidos en la etapa de diseño, en este capítulo se procede a la fase de implementación del sistema, donde se muestra el modelo de despliegue del sistema y los diagramas de componentes del requisito más importante. Posteriormente se le aplican las pruebas a la aplicación, partiendo de la confección y descripción de los casos de prueba.

4.1 Modelo de implementación

El modelo de implementación describe la estructura general del *software* con vistas a su construcción, ejecución e instalación, es el encargado de describir cómo los elementos de diseño se implementan en componentes, tales como ficheros de código fuente, ejecutables y biblioteca. Dicho modelo abarca el diagrama despliegue para describir la distribución física del sistema y los diagramas de componentes encargados de mostrar cuáles son las diferentes partes del *software*.

4.1.1 Diagramas de Componentes

Los diagramas de componentes representan el empaquetamiento físico de los elementos del modelo de diseño. Describen la descomposición física del sistema de *software*, a efectos de construcción y funcionamiento, dicha descomposición se realiza en términos de componentes y de relaciones entre los mismos (29).

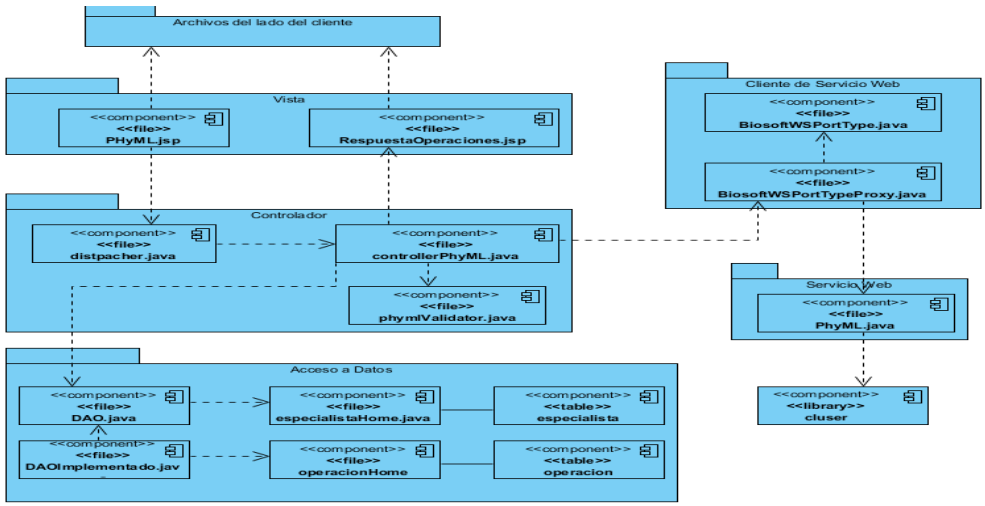


Figura 12. Diagrama de componentes correspondiente al CU *"*Realizar Análisis Filogenético*,* sesión *"*Análisis filogenético con PhyML*".*

A continuación se muestra una breve descripción de los componentes expuestos en el diagrama anterior. Para consultar los diagramas de componentes de los restantes casos de uso remitirse al anexo 4.

Tabla 3. Descripción de los componentes del CU "Realizar Análisis Filogenético", sesión “Análisis filogenético con PhyML”.

Componente	Función
Archivos del lado del cliente	Contiene los estilos CSS y los archivos JavaScript que se ejecutan en el navegador del cliente.
PhyMl.jsp	Página JSP que mostrará al usuario el formulario que permite realizar el análisis filogenético con PhyML.
RespuestaOperaciones.jsp	Página JSP que muestra todas las ejecuciones.
Distpacher.java	Clase Java encargada de gestionar todas las peticiones de los usuarios y asignarlas a su controlador correspondiente.
controllerphyML.java	Clase Java que procesa y responde a la petición de mostrar de realizar análisis con PhyML.
phymlValidator.java	Clase Java que valida cada uno de los parámetros de PhyML.
DAO. java	Interfaz del lenguaje Java que declara los métodos necesarios para acceder a la base de datos.
DAOImplementado.java	Clase Java que se encarga de implementar los métodos que permiten el acceso a la base de datos.
especialistaHome.java	Clase responsable de manipular la tabla especialista.
operacionHome.java	Clase responsable de manipular la tabla operación.
especialista	Archivo responsable de asociar al especialista con la base de datos.
operacion	Archivo responsable de asociar la operación con la base de datos.
BiosoftWSPortType.java	Clase interfaz que declara todas las operaciones que invocan los servicios web.
BiosoftWSPortTypeProxy.java	Clase que implementa las operaciones de la interfaz BiosoftWSPortType.java.
PHyML.java	Clase Java que contiene las funcionalidades necesarias para ejecutar el programa PhyML.

Cluster.jar	Biblioteca que permite acceder a las funcionalidades del clúster, como enviar una tarea, autenticarse y obtener información.

4.1.2 Modelo de despliegue

Los modelos de despliegue representan la disposición física de los distintos nodos que entran en la composición de un sistema y el reparto de los programas ejecutables sobre estos nodos, donde cada nodo representa un recurso de computación, que generalmente tiene algo de memoria y, a menudo, capacidad de procesamiento (29). En fin estos diagramas son utilizados para describir los componentes de *hardware*, donde los componentes de *software* se han instalado.

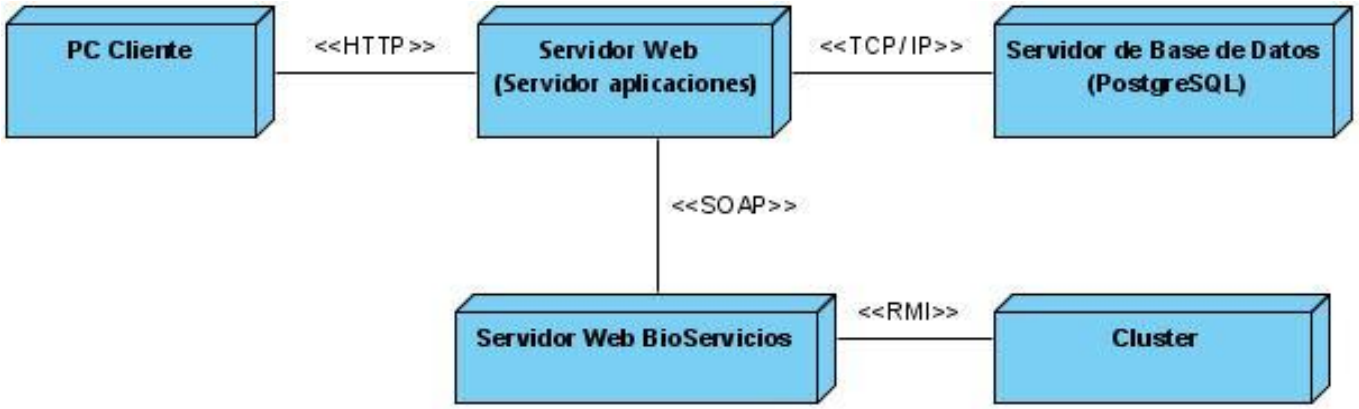


Figura 13. Diagrama de despliegue del servicio Inferencia Filogenética Molecular.
Se representan los recursos necesarios para el despliegue del sistema.

Para el módulo de Inferencia Filogenética se requiere al menos de una computadora (PC) con un navegador web, que permita realizar peticiones a través del protocolo de red HTTP al servidor de aplicaciones. Los datos persistentes del sistema serán almacenados en el servidor de bases de datos, al que se puede acceder utilizando el protocolo TCP/IP, por último las funcionalidades están accesibles en el servidor de servicios web a través de SOAP, el cual a través del Método de Invocación Remota (RMI, por sus siglas en inglés) se conecta con el clúster para ejecutar las tareas de alto procesamiento que exigen los programas PhyML y MrBayes.

4.2 Interfaces de la aplicación

El módulo de Inferencia Filogenética se compone por varias vistas que le permiten al especialista interactuar fácilmente con la aplicación. A continuación se muestran la interfaz principal de dicho módulo la cual posibilita realizar los análisis filogenéticos empleando los programas PhyML y MrBayes, además de poder consultar las ejecuciones realizadas para eliminarlas o descargarlas en caso que desee.

Figura 14. Interfaz principal del módulo de Inferencia Filogenética Molecular

El sistema permite además realizar análisis filogenéticos empleando el programa PhyML, la (Figura 15) muestra el formulario encargado de recopilar los parámetros necesarios para realizar el análisis empleando este programa.

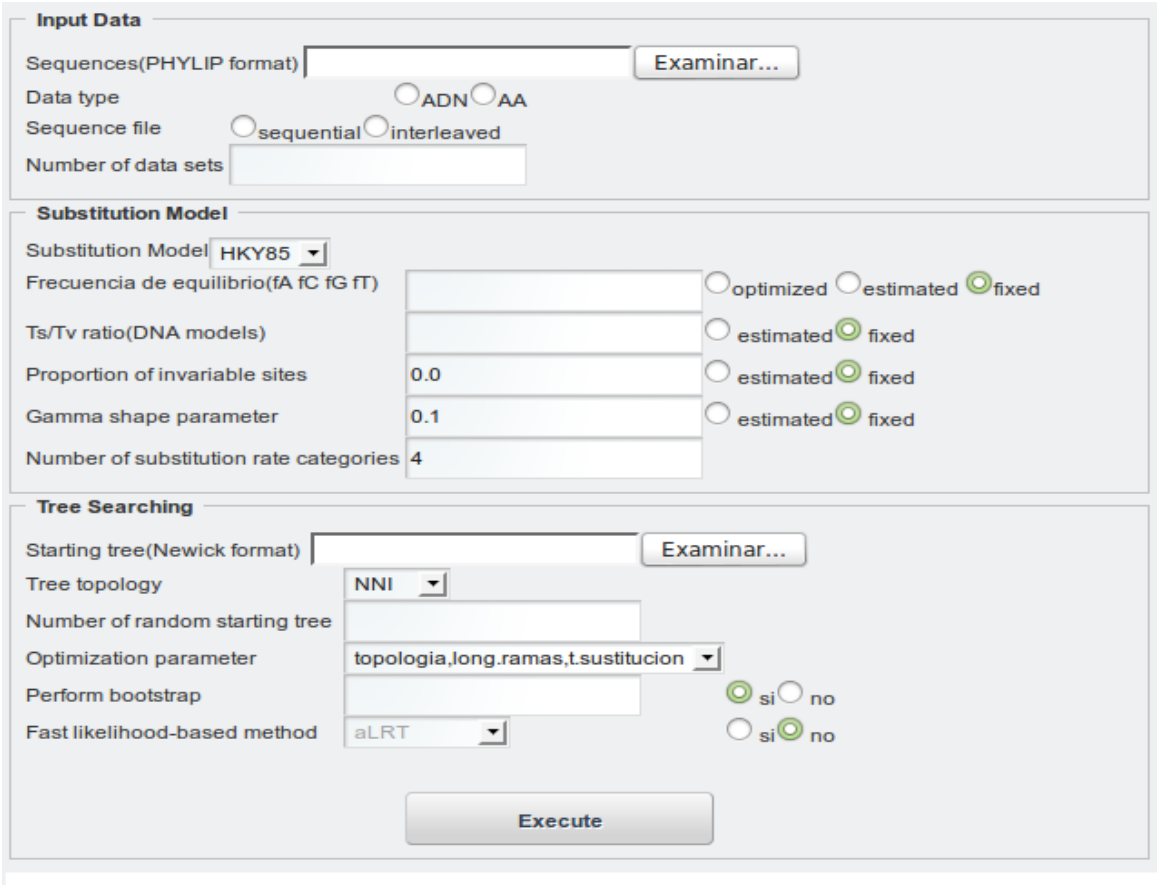


Figura 15. Interfaz para realizar análisis con PhyML.
Formulario para realizar análisis que recoge los datos necesarios filogenéticos con PhyML**.**

Para realizar análisis filogenéticos empleando el programa MrBayes se utiliza el siguiente formulario, en el cual se introducen los archivos para los campos de alineamiento de secuencia y archivo de comando.

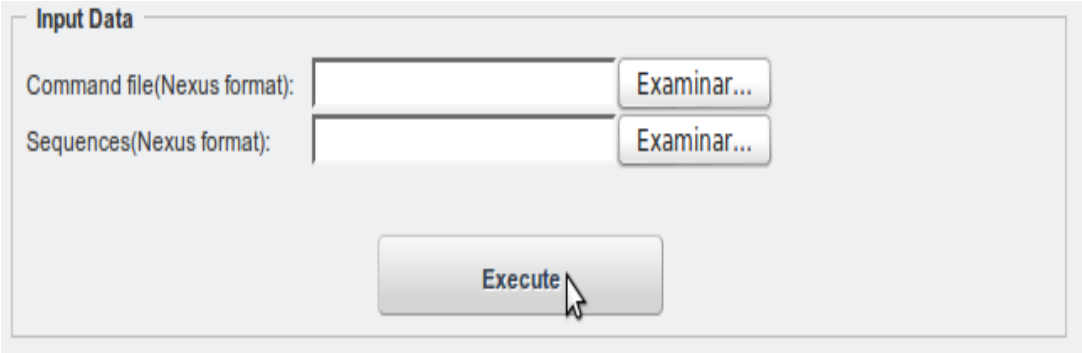


Figura 16. Interfaz para realizar análisis con MrBayes.
Formulario que recoge los datos necesarios para realizar análisis filogenéticos con MrBayes

Una vez realizadas las estimaciones filogenéticas por los programas PhyML o MrBayes se puede observar a través de esta interfaz todas las ejecuciones realizadas, brindando además la posibilidad de descargarlas o eliminarlas.

No	Tipo	Identificador Tarea	Fecha	Opciones
1	MrBayes	FC5BD06CA08B091C	2013-05-24	
2	MrBayes	CB7618312BAB90C3	2013-05-24	
3	MrBayes	6786483714D3F480	2013-05-24	
4	MrBayes	9372E7DA2C4059FC	2013-05-24	
5	PhyMl	3A4C8664A0C4629F	2013-05-24	
6	PhyMl	B2E24A1645B57A98	2013-05-24	

Figura 17. Interfaz de respuesta.
Lista todas las ejecuciones realizadas por el especialista.

4.3 Pruebas de software

Las pruebas es una actividad en la cual un sistema o componente es ejecutado bajo unas condiciones o requerimientos específicos, los resultados son observados y registrados, y una evaluación es hecha de algún aspecto del sistema o componente. Su principal objetivo es evaluar o valorar la calidad del producto a través de:

- Encontrar y documentar los defectos que puedan afectar la calidad del *software*.

- Validar que el *software* trabaje como fue diseñado.
- Validar y probar los requisitos que debe cumplir el *software*.
- Validar que los requisitos fueron implementados correctamente.

La prueba de *software* es un elemento crítico para la garantía de la calidad del *software* y representa una revisión final de las especificaciones del diseño y de la codificación (30). Las pruebas de caja negra constituyen unos de los métodos fundamentales de pruebas de *software*.

4.3.1 Pruebas de caja negra

La prueba de Caja Negra se centra principalmente en los requisitos funcionales del *software*. Estas pruebas permiten obtener un conjunto de condiciones de entrada que ejerciten completamente todos los requisitos funcionales de un programa. Estas pruebas se llevan a cabo sobre la interfaz del *software*, por lo que los casos de prueba pretenden demostrar que las funciones del *software* son operativas, que la entrada se acepta de forma adecuada y que se produce una salida correcta, así como que la integridad de la información externa se mantiene (30).

Para preparar los casos de pruebas hacen falta un número de datos que ayuden a la ejecución de estos casos y que permitan que el sistema se ejecute en todas sus variantes, pueden ser datos válidos o inválidos para el programa según si lo que se desea es hallar un error o probar una funcionalidad. Dentro de las técnicas que existen para desarrollar las pruebas de caja negra se encuentra la de Partición de Equivalencia, la cual divide el campo de entrada en clases de datos que tienden a ejercitar determinadas funciones del *software*.

Dentro del método de Caja Negra la técnica de la Partición de Equivalencia es una de las más efectivas, pues permite examinar los valores válidos e inválidos de las entradas existentes en el *software.* Siguiendo las directrices que propone dicha técnica se diseñó un caso de prueba para el caso de uso Realizar Análisis Filogenético, donde se seleccionaron cinco variables a medir, las cuales se describen a continuación:

Tabla 4. Variables para el caso de prueba análisis filogenético con PhyML.

No	Nombre de campo	Clasificación	Valor Nulo	Descripción
1	Número random	Campo de texto	Si	Número mayor que 0
2	Boostrap	Campo de texto	Si	Número igual o mayor que 0
3	Número de datos	Campo de texto	Si	Número mayor que 0
4	Sitios de proporción invariable	Campo de texto	Si	Número mayor que 0

5	Radio	Campo de texto	Si	Número mayor que 0

Tabla 5. Caso de Prueba: Análisis filogenético con PhyML.

Escenario	Descripción	V 1	V2	V3	V4	V5	Respta del sistema	Flujo central
EC 1.1 Introducir los datos correctamente	Se llenan los campos correctamente y se realiza el análisis filogenético	V[4]: 100	V: 50	V: 4	V: 0.3	V: 0.1	Realiza el análisis filogenético	1.Realizar Análisis/ PhyML 2.Llenan los datos 3.Enviar
EC 1.2 Introducir valores en blanco	Se desea realizar análisis filogenético dejando campos en blanco	I[5]: ( )	V: 50	V: 4	V: 0.3	V: 0.1	Realiza el análisis filogenético	1.Realizar Análisis/ PhyML 2.Llenan los datos 3.Enviar
		V: 100	I: ( )	V: 4	V: 0.3	V: 0.1		
		V: 100	V: 50	I: ( )	V: 0.3	V: 0.1		
		V: 100	V: 50	V: 4	NA[6]: ( )	V: 0.1		
		V: 100	V: 50	V: 4	V: 0.3	I: ( )		
EC 1.3 Introducir valores incorrectos	Se desea realizar el análisis introduciendo valores incorrectos	I: -5	V: 50	V: 4	V: 0.3	V: 0.1	Se muestra un mensaje de error	1.Realizar Análisis/ PhyML 2.Llenan los datos 3.Enviar
		V: 100	I: - 10	V: 4	V: 0.3	V: 0.1		
		V: 100	V: 50	I: - 1	V: 0.3	V: 0.1		
		V: 100	V: 50	V: 4	I: 3	V: 0.1		
		V: 100	V: 50	V: 4	I: -2	V: 0.1		
		V: 100	V: 50	V: 4	V: 0.3	I: -8		

[4] Entrada válida
[5] Entrada inválida
6 No aplica

4.4 Pruebas de Speed-Up

Las pruebas de *Speed-Up* son aplicadas para comprobar la ganancia de velocidad de la aplicación una vez utilizado el clúster de computadora como recurso de alto procesamiento. El *Speed-Up* de *p* procesadores (*Sp*) es el cociente entre el tiempo de ejecución de un programa secuencial (*TS*) y el tiempo de ejecución de la versión paralela de dicho programa en *p* procesadores (*TP)*.

$$Sp = \frac{Ts}{Tp}$$

En el mejor de los casos el tiempo de ejecución de un programa en paralelo con *p* procesadores será *p* veces inferior al de su ejecución en un solo procesador, teniendo todos los procesadores igual potencia de cálculo (31). Es por ello que el valor máximo que puede alcanzar el *Speed-Up* es p, razón por la cual los resultados suelen compararse con la recta y = x.

A continuación se mostrarán una serie de tablas y gráficas que plasmarán los resultados de dichas pruebas al ejecutar los programas PhyML y MrBayes en el clúster.

PhyML

Para realizar las pruebas con el programa PhyML fue empleado un fichero de 100 secuencias de nucleótidos con una extensión de 600 caracteres, especificándole para cada iteración el modelo evolutivo HKY85, una búsqueda de topología por SPR y un *boostrap* igual a 10.

Se realizaron cinco iteraciones de las pruebas y se calcularon los promedios de tiempo para cada uno de los nodos, los resultados obtenidos se muestran a continuación:

Tabla 6. Speed-Up para PhyML

Cantidad de nodos	Tiempo	Speed-Up
1	00:50:44	1
3	00:31:11	1.626
5	00:20:20	2.495
8	00:23:42	2.140

Como se muestra en la Tabla 6 el valor del *speed-up* obtenido para ocho nodos es menor que el obtenido para cinco, por lo que se puede atribuir que cinco nodos son suficientes para realizar un análisis filogenético con el fichero y los parámetros antes descritos como entrada. En la (Figura 18) se grafican los datos antes expuestos para una mejor comprensión.

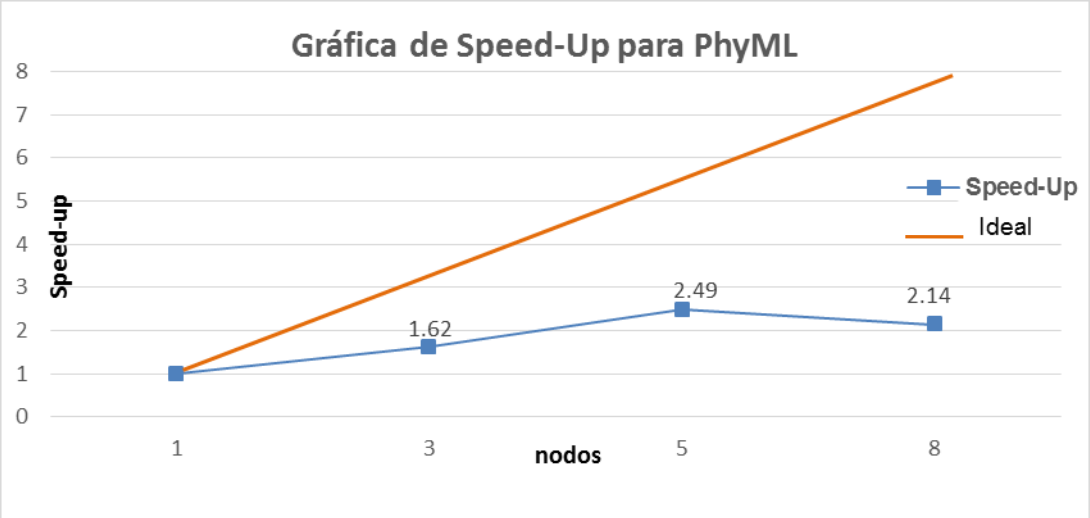


Figura 18. Gráfico de Speed-Up para PhyML

MrBayes

Las pruebas de *speed-up* del programa MrBayes fueron aplicadas con las mismas características que las de PhyML. Los ficheros de entrada utilizados tienen la misma estructura en cuanto a información, pero con la particularidad de realizar 2000 generaciones con un modelo evolutivo GTR con distribución gamma. La Tabla 7 muestra los tiempos y el *speed-up* obtenido de dicha prueba.

Tabla 7. Speed-Up para MrBayes

Cantidad de nodos	Tiempo	Speed-Up
1	04:10:28	1
3	01:47:51	2.32
5	01:14:00	3.38
8	00:52:14	4.79

En la tabla anterior se muestra claramente que los valores de *speed-up* van en aumento, de forma tal que a medida que se incrementa la cantidad de nodos se obtiene una mayor ganancia de velocidad. En la (Figura 19) se muestran graficados los valores de speed-up en correspondencia con la recta y=x, que representa lo ideal, o sea que el speed-up sea igual a la cantidad de nodos.

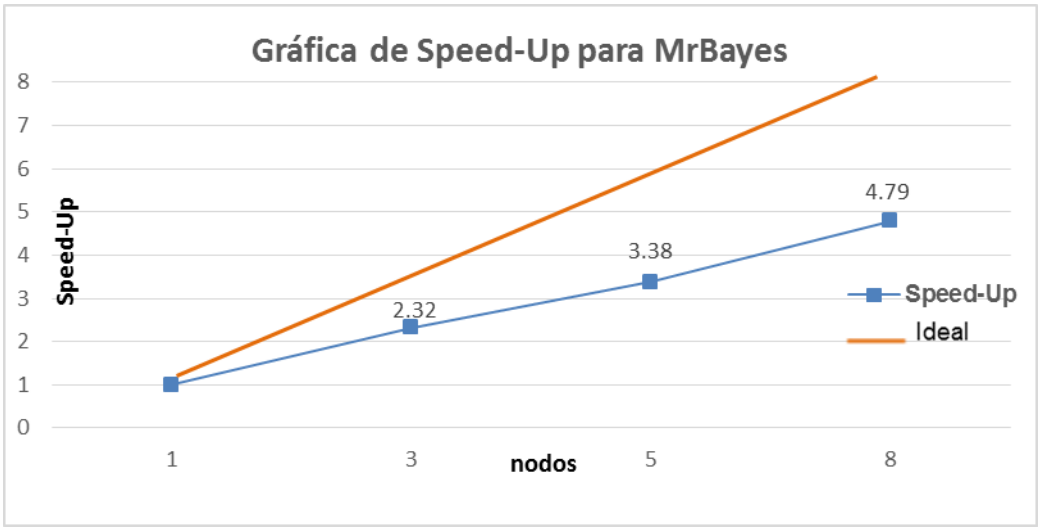


Figura 19. Gráfica de Speed-Up para MrBayes

4.5 Conclusiones Parciales

En este capítulo se culminó la fase de implementación del módulo de Inferencia Filogenética Molecular, con la construcción del diagrama de componentes para el CU “Realizar Análisis Filogenético”, para la sesión “Realizar Análisis con PhyML”, donde se especificaron las relaciones y dependencias entre cada uno de los componentes de la aplicación.

A partir de la elaboración del diagrama de despliegue se identificaron los recursos necesarios para el despliegue y correcto funcionamiento de la solución.

Una vez implementado el sistema se realizaron las pruebas de caja negra aplicando la técnica particiones de equivalencia, a partir de las cuales se detectaron algunos errores de validación, los que fueron erradicados con el objetivo de cumplir satisfactoriamente con los requisitos funcionales.

Las pruebas de speed-up arrojaron como resultado que para PhyML con solo cinco procesadores es suficiente para obtener un aumento de aceleración y para el caso de MrBayes se evidenció la ganancia de aceleración con el aumento de los nodos.

Conclusiones Generales

Los resultados alcanzados con la realización de este trabajo de diploma han permitido arribar a las siguientes conclusiones:

- Se seleccionaron los programas PhyML y MrBayes, que incluyen los métodos de máxima verosimilitud y bayesianos respectivamente, para realizar análisis de inferencia filogenética mediante el empleo de un clúster de computadoras, como recurso para disminuir el tiempo de respuesta.

- El análisis del proceso de desarrollo del módulo de Inferencia Filogenética permitió la identificación de tres requisitos funcionales: Realizar Análisis Filogenético, Administrar y Salvar Resultado, los cuales se tomaron en cuenta para realizar el diseño de las clases del *software*, aplicando los patrones de diseño GRASP y GoF.

- Se implementaron satisfactoriamente los requisitos funcionales identificados, obteniendo un sistema capaz de realizar estimaciones filogenéticas usando los programas PhyML y MrBayes a través de un clúster de computadoras.

- La aplicación de las pruebas de caja negra permitieron la identificación de cuatro no conformidades, las cuales fueron corregidas en la segunda iteración, con el objetivo de garantizar la calidad y buen funcionamiento de la aplicación.

- Las pruebas de Speed-Up arrojaron como resultado, para el set de datos empleado, que el programa PhyML con solo cinco nodos de los ocho del clúster obtiene un aumento de la aceleración, mientras que el programa MrBayes tiene un aumento gradual de aceleración a medida que los nodos se incrementan.

Recomendaciones

Incluir otros servicios que utilicen criterios bayesianos más exhaustivos en cuanto a las posibles relaciones evolutivas entre los organismos.

Insertar un servicio de visualización y manipulación de los árboles filogenéticos arrojados por PhyML y MrBayes.

Brindar la posibilidad en una futura versión de que el usuario pueda especificar, a la hora de realizar el análisis filogenético, el número de nodos a utilizar en el mismo.

Referencias Bibliográficas

1. **Medina, Annia Ferrer, Galez, Marcelo Nazabal y Perez, Lidia Ines Novoa.** Una Aproximación a la genómica. [aut. libro] C.Holmes.

2. **Hormazáabal, Jorge Martínez.** *Función e Importancia de la Bioinformática en el desarrollo de las Ciencias, especialmente en Biotecnología y Medicina Molecular.* Facultad de Ciencias de la universidad de Chile. Chile, Las Palmeras : s.n.

3. **Vinuessa, Pablo.** Introduccion a la Inferencia Filogenética Molecular. *Programa de Ingeniería Genómica.* México : s.n., págs. 1-20.

4. **C.Holmes.** Inferring Molecular Phylogeny. [aut. libro] Holmes C. *Molecular Evolution.* 6, págs. 173-227.

5. **Vinuessa, Pablo.** *Inferencia Bayesiana de Filogenias.* Cuernavaca,México : s.n., junio,2007. págs. 1-27, Curso Fundamental de Inferencia Filogenética Molecular.

6. **Yang, Ziheng.** *Computacional Molecular Evolution OSEE.* London : British Library Cataloguing, marzo 2006. pág. 449.

7. Biowulf at the NIH. [En línea] http://biowulf.nih.gov/apps/mrbayes.html.

8. **Guindon, S.** South of France Bioinformatics platform. [En línea] 2010. http://www.atgc-montpellier.fr/phyml/.

9. **Microsystems, Sun.** [En línea] http://www.java.com/.

10. **Allamaraju, Subrahmanyam, y otros.** *Programación Java Server con J2EE Edition.* pág. 1206.

11. [En línea] [Citado el: 2012 de noviembre de 10.] http://www.sicuma.uma.es/sicuma/independientes/argentina08/Badaracco/j2ee.html./.

12. *Los portlets en la construcción de portales web: OracleWebCenter.* **Pérez, Isis Torres.** 2008, Convención Científica de Ingeniería y Arquitectura, Vol. 14, págs. 1-4.

13. **Alvarez, Fabricio.** [En línea] http://es.scribd.com/doc/66877224/13/Contenedor-de-Portlets.

14. Liferay. [En línea] [Citado el: 2012 de noviembre de 18] http://www.liferay.com/.../14/b9274c27-ef7b-41bb-be3c-0aae3b03aa59.

15. **Gutiérréz, Javier J.** *¿Qué es un framework web?* págs. 1-4.

16. **Apache Axis.** Apache Axis2. [En línea] [Citado el: 2012 de noviembre de 18] http://axis.apache.org/axis2/java/core/.

17. *Diseño y desarrollo de un sistema con un módulo para la administración de los recursos técnicos utilizando un IVR para la empresa Telalca S.A.* **Jaramillo, Ricardo Mena.** Universidad de San Francisco : s.n., 2010.

18. *Clúster de Alto Rendimiento.* **Pérez, Virgilio Cervantes.** Universidad de Guadalajara, México : s.n. págs. 1-27.

19. **Eclipse.** Eclipse Documentation. *Eclipse Documentation.* [En línea] [Citado el: 2013 de enero de 15] http://www.eclipse.org/..

20. [En línea] [Citado el: 2013 de enero de 16.] http://www.eclipse.org/epf/general/OpenUp.pdf..

21. **UCI.** Entorno Virtual de Aprendizaje. [En línea] [Citado el: 2013 de febrero de 18] http://eva.uci.cu/mod/resource/view.php?id=9400&subdir=/Otros_materiales. Aprendiendo_UML_en_24_horas.part2.rar.

22. Visual Paradimg para UML. [En línea] http://www.software.com.ar.

23. *Revista Digital de Innovación y experiencias educativas.* **Peréz, Teresa Garzón.** 30, 2010, págs. 1-15.
24. **PostgreSQL.** [En línea] [Citado el: 2013 de febreo de 2.] http://www.postgresql.org.es/.
25. [En línea] [Citado el: 2013 de febrero de 5.] http://www.aplicacionesempresariales.com/ventajas-y-desventajas-de-postgresql.html.
26. **Larman, Craig.** *UML y Patrones.* México : PRENTICE HALL. 2003. Vol. 2.
27. **Almaira, Adriana Sandra.** *Entorno Virtual de Aprendizaje.* [En línea] marzo de 2007. [Citado el: 2013 de marzo de 25] http://eva.uci.cu/file.php/158/Documentos/Recursos_bibliograficos/Libros_y_articulos_UD_1/Arquitectura_de_Software/Tesina_Arquitectura_de_Soft.pdf.
28. **Larman, Craig.** *UML y Patrones.* México : PRENTICE HALL, 1999.
29. **Falgueras, Benet Camptderrich.** *Ingeniería del software .* Barcelona : UOC, 2003.
30. Entorno Virtual de Aprendizaje. *Entorno Virtual de Aprendizaje.* [En línea] http://eva.uci.cu/mod/resource/view.php?id=8649&subdir=/Comun.
31. **Rasúa, Rafael Arturo Trujillo.** Algoritmos paralelos para la solución de problemas de optimización discretos aplicados a la decodificación de señales. [En línea] marzo de 2009. [Citado el: 2013 de mayo de 26.] www.dsic.upv.es/docs/bib-dig/tesis/etd-05182009.../TesisTrujillo.pdf.

Bibliografía

1. **Medina, Annia Ferrer, Galez, Marcelo Nazabal y Perez, Lidia Ines Novoa.** Una Aproximación a la genómica. [aut. libro] C.Holmes.

2. **Hormazáabal, Jorge Martínez.** *Función e Importancia de la Bioinformática en el desarrollo de las Ciencias, especialmente en Biotecnología y Medicina Molecular.* Facultad de Ciencias de la universidad de Chile. Chile, Las Palmeras : s.n.

3. **Vinuessa, Pablo.** Introduccion a la Inferencia Filogenética Molecular. *Programa de Ingeniería Genómica.* México : s.n., págs. 1-20.

4. **C.Holmes.** Inferring Molecular Phylogeny. [aut. libro] Holmes C. *Molecular Evolution.* 6, págs. 173-227.

5. **Vinuessa, Pablo.** *Inferencia Bayesiana de Filogenias.* Cuernavaca,México : s.n., junio,2007. págs. 1-27, Curso Fundamental de Inferencia Filogenética Molecular.

6. **Yang, Ziheng.** *Computacional Molecular Evolution OSEE.* London : British Library Cataloguing, marzo 2006. pág. 449.

7. Biowulf at the NIH. [En línea] http://biowulf.nih.gov/apps/mrbayes.html.

8. **Guindon, S.** South of France Bioinformatics platform. [En línea] 2010. http://www.atgc-montpellier.fr/phyml/.

9. **Microsystems, Sun.** [En línea] http://www.java.com/.

10. **Allamaraju, Subrahmanyam, y otros.** *Programación Java Server con J2EE Edition.* pág. 1206.

11. [En línea] [Citado el: 2012 de noviembre de 10.] http://www.sicuma.uma.es/sicuma/independientes/argentina08/Badaracco/j2ee.html./.

12. *Los portlets en la construcción de portales web: OracleWebCenter.* **Pérez, Isis Torres.** 2008, Convención Científica de Ingeniería y Arquitectura, Vol. 14, págs. 1-4.

13. **Alvarez, Fabricio.** [En línea] http://es.scribd.com/doc/66877224/13/Contenedor-de-Portlets.

14. Liferay. [En línea] [Citado el: 2012 de noviembre de 18] http://www.liferay.com/.../14/b9274c27-ef7b-41bb-be3c-0aae3b03aa59.

15. **Gutiérréz, Javier J.** *¿Qué es un framework web?* págs. 1-4.

16. **Apache Axis.** Apache Axis2. [En línea] [Citado el: 2012 de noviembre de 18] http://axis.apache.org/axis2/java/core/.

17. *Diseño y desarrollo de un sistema con un módulo para la administración de los recursos técnicos utilizando un IVR para la empresa Telalca S.A.* **Jaramillo, Ricardo Mena.** Universidad de San Francisco : s.n., 2010.

18. *Clúster de Alto Rendimiento.* **Pérez, Virgilio Cervantes.** Universidad de Guadalajara, México : s.n. págs. 1-27.

19. **Eclipse.** Eclipse Documentation. *Eclipse Documentation.* [En línea] [Citado el: 2013 de enero de 15] http://www.eclipse.org/..

20. [En línea] [Citado el: 2013 de enero de 16.] http://www.eclipse.org/epf/general/OpenUp.pdf..

21. **UCI.** Entorno Virtual de Aprendizaje. [En línea] [Citado el: 2013 de febrero de 18] http://eva.uci.cu/mod/resource/view.php?id=9400&subdir=/Otros_materiales.

Aprendiendo_UML_en_24_horas.part2.rar.
22. Visual Paradimg para UML. [En línea] http://www.software.com.ar.
23. *Revista Digital de Innovación y experiencias educativas.* **Peréz, Teresa Garzón.** 30, 2010, págs. 1-15.
24. **PostgreSQL.** [En línea] [Citado el: 2013 de febreo de 2.] http://www.postgresql.org.es/.
25. [En línea] [Citado el: 2013 de febrero de 5.] http://www.aplicacionesempresariales.com/ventajas-y-desventajas-de-postgresql.html.
26. **Larman, Craig.** *UML y Patrones.* México : PRENTICE HALL. 2003. Vol. 2.
27. **Almaira, Adriana Sandra.** *Entorno Virtual de Aprendizaje.* [En línea] marzo de 2007. [Citado el: 2013 de marzo de 25] http://eva.uci.cu/file.php/158/Documentos/Recursos_bibliograficos/Libros_y_articulos_UD_1/Arquitectura_de_Software/Tesina_Arquitectura_de_Soft.pdf.
28. **Larman, Craig.** *UML y Patrones.* México : PRENTICE HALL, 1999.
29. **Falgueras, Benet Camptderrich.** *Ingeniería del software .* Barcelona : UOC, 2003.
30. Entorno Virtual de Aprendizaje. *Entorno Virtual de Aprendizaje.* [En línea] http://eva.uci.cu/mod/resource/view.php?id=8649&subdir=/Comun.
31. **Rasúa, Rafael Arturo Trujillo.** Algoritmos paralelos para la solución de problemas de optimización discretos aplicados a la decodificación de señales. [En línea] marzo de 2009. [Citado el: 2013 de mayo de 26.] www.dsic.upv.es/docs/bib-dig/tesis/etd-05182009.../TesisTrujillo.pdf.

Anexos

Anexo 1: Descripción de Casos de Uso

Tabla 8