

Universidad de las Ciencias Informáticas

Facultad 6



**Título: Módulo básico de la Plataforma de Servicios
Bioinformáticos.**

**Trabajo de Diploma para optar por el título de
Ingeniero Informático.**

Autor (es): Andry Daniel Díaz León

Tutor (es): MSc. Orlando Martínez Pérez

MSc. Alina Agramonte Delgado

La Habana, mayo de 2012

“Año 54 de la Revolución”



Ludwig van Beethoven (Bonn, Alemania, 16 de diciembre de 1770 – Viena, Austria, 26 de marzo de 1827) fue un compositor, director de orquesta y pianista alemán. Su legado musical se extendió desde el período clásico hasta el romanticismo musical. “El Coral”, su novena y última sinfonía, es el Himno de la Unión Europea y la única obra musical en toda la historia declarada Patrimonio de la Humanidad por la UNESCO. A pesar de los problemas familiares y auditivos que tuvo Beethoven, compuso obras musicales consideradas por muchos artistas, la cúspide de la música clásica.

“¡Actúa en vez de suplicar. Sacrifícate sin esperanza de gloria ni recompensa! Si quieres conocer los milagros, hazlos tú antes. Sólo así podrá cumplirse tu peculiar destino.”

Beethoven, L. van

Declaración de autoría

Yo Andry Daniel Díaz León declaro ser el autor del trabajo de diploma con título: Módulo básico de la Plataforma de Servicios Bioinformáticos 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 el presente a los ____ días del mes de _____ del año 2012.

Firma del autor

Andry Daniel Díaz León

Firma del tutor

Alina Agramonte Delgado

Firma del tutor

Orlando Martínez Pérez

Agradecimientos

Gracias a todos mi amigos, a mi barrio, al piquete de Calle 5, hemos compartido momentos que perdurarán en la memoria. Gracias por compartir lo que tienen sin interés alguno, por su autenticidad, por su respeto, por estar siempre ahí en el momento y lugar adecuado, gracias a todos mis compañeros, en especial a: Julio César Romero y Manuel Alejandro.

Jamás olvidaré a ese conjunto de personas con las que interactué en mi primer año de esta carrera, es imposible nombrarlas a todas, pero existe un distintivo que nos aglutina en un todo, aunque tristemente nos fuimos separando por motivos de la vida, el grupo 6105 siempre será uno. Gracias al único grupo al que he pertenecido.

Quiero agradecer a toda mi familia. Por ser promotores de mis buenos sentimientos, eso se lo agradezco a todos ustedes, por educarme y mostrarme el concepto de familia, ese núcleo inquebrantable de amor, de unidad, de paz y tranquilidad.

A lo largo de mi carrera he tenido muchos maestros, pero sólo uno se distinguió por ser independiente del método, supo enseñar, aprender y educar. Por sus enseñanzas, por su singular dedicación, por su inspiración, gracias a Raimundo Bermudez, mi maestro predilecto.

Quiero agradecerles a dos personas por brindarme la posibilidad de compartir su espacio, por convertirlo en algo mas que un proyecto, causa admiración ver el gran talento; modesto, libre de petulancia y privado de presunción que los caracteriza. Por impulsar mi autoconfianza, por su atención en todo momento, por su amistad. Su forma de ser constituyó fuente de responsabilidad, compromiso y entrega. Gracias a Alina y a Orlando, mis tutores.

Existe una persona con la que he convivido mucho tiempo, su contacto me ha permitido aprender mucho de mí y aún mucho más de ella. Gracias por todo este tiempo, gracias por ser un puente de amistad a otras personas increíbles, las que me place haber conocido, personas como tú y tu familia son difíciles de encontrar. Gracias a Yanet y familia.

No quiero dejar fuera a dos seres queridos, ellos fueron vitales en mi desarrollo desde la niñez hasta la adolescencia, cuando repentinamente abandonaron este mundo. Quiero agradecerles tantas cosas que serían insuficiente cien cuartillas. Pero el carácter y el temperamento que sembraron en mi, han sido pilares invaluable en este resultado, se que están orgullosos de mi, gracias a mis abuelos Dagoberto y Domingo.

En este largo camino he tenido el placer de conocer muchas personas, mas ha sido un orgullo el conocer a esta, excepcional, de geniales virtudes humanas, con ejemplo trascendental, causa más regocijo saber que compartimos la misma sangre. Gracias por tus consejos, por tu sacrificio y

entrega, gracias por ser manantial inagotable de ese legado lleno de paciencia, inspiración e ilusión, gracias sobre todas las cosas, por ser mi hermano. Gracias Dalier.

Es difícil mantenerse firme cuando escuchas mil voces diciendo NO y casi no te escuchas diciendo SÍ. Pero hay personas que persisten y ponen fe en sus principios. Desde pequeño cuando solías llevarme a tus sesiones de trabajo, fui testigo de esta situación. Gracias por demostrarme que nada es imposible en este mundo cuando de veras te lo propones, no importan los obstáculos, sólo basta creer en ello. Gracias Delvis, mi papá.

No encuentro palabras ni idioma para agradecerte a ti, pero lo intento. Gracias por estar ahí, en el frío y el calor, en las sombras y en la luz, en la tormenta y en la tranquilidad. Por enseñarme, por educarme, por crearme. No han sido, sino tus valores, el núcleo de todo lo que veo, lo que toco, lo que pienso, lo que digo, lo que escucho y lo que hago. Cuando desconcertado quiero todas las respuestas, recuerdo tu máxima predilecta: no existe pregunta alguna que el amor no pueda contestar. Gracias por todo, gracias por existir, gracias Migdalia, mi mamá.

Dedicatoria

A toda mi familia, que siempre han esperado lo mejor de mí, especialmente a mi hermano Dalier, a mi mamá Migdalia y a mi papá Delvis.

A la Revolución Cubana, sin la cual no hubiera sido posible este acontecimiento, en especial a su máximo líder Fidel Castro Ruz.

A mis abuelos Dagoberto y Domingo, aunque hoy, me es imposible su compañía y estén más presente que nunca.

Resumen

Las Tecnologías de la Información y las Comunicaciones (TICs) son un factor de vital importancia en la transformación de la sociedad, especialmente para áreas del conocimiento como la Bioinformática. La presente investigación se enmarca en la concepción e implementación de un módulo de herramientas básicas para el análisis de secuencias moleculares que forma parte de la Plataforma de Servicios Bioinformáticos del centro DATEC de la Universidad de las Ciencias Informáticas. El mismo pretende brindar acceso a dichas herramientas a los investigadores de universidades y centros del país que promueven y desarrollan la Bioinformática. Con este fin se realiza un análisis de los principales sitios web que posibilitan el acceso a estas aplicaciones a través de Internet y se definieron las tecnologías y programas para desarrollar el sistema. El proceso estuvo guiado por la metodología de desarrollo de software OpenUP y se utilizó como lenguaje de programación Java. Finalmente se obtuvo un portlet que contiene una suite de aplicaciones para la edición y manipulación de secuencias moleculares que permite generar secuencias aleatorias, complementarlas e invertirlas, predecir segmentos de transmembrana y marcos abiertos de lectura, así como realizar alineamientos múltiples de secuencias. El sistema se integra a la Plataforma de Servicios Bioinformáticos de la Universidad de las Ciencias Informáticas (UCI) y puede ser reutilizado por otras aplicaciones indistintamente de la tecnología en que estas han sido elaboradas.

Palabras claves: Bioinformática, análisis de secuencias moleculares, Plataforma de Servicios Bioinformáticos, módulo.

Índice de contenido

Índice de figuras.....	IV
Índice de tablas.....	V
Introducción.....	1
Capítulo 1. Fundamentación teórica.....	5
1.1 Evolución de la World Wide Web.....	5
1.1.1 Recursos bioinformáticos en la Web.....	6
1.1.2 Servicios web en la gestión de la información biológica.....	8
1.1.3 Arquitectura Orientada a Servicios.....	9
1.2 Técnicas y lenguajes de programación para el desarrollo de aplicaciones web.....	10
1.2.1 Programación Orientada a Objeto y Orientada a Aspecto.....	11
1.2.2 Lenguajes y plataformas de desarrollo. Java 2 Enterprise Edition.....	12
1.2.3 Algunos componentes de la plataforma Java 2 Enterprise Edition.....	14
1.3 Contenedores de Portlets en la plataforma J2EE.....	16
1.3.1 Liferay Portal.....	17
1.4 Marcos de trabajo.....	17
1.4.1 Marcos de trabajos de mapeo relacional. Hibernate.....	18
1.4.2 Marcos de trabajos para aplicaciones web. Spring Framework.....	18
1.4.3 Marcos de trabajos JavaScript. JQuery y DWR.....	19
1.4.4 Marcos de trabajos para crear servicios web. Apache Axis 2.....	19
1.5 Sistemas gestores de base de datos.....	20
1.5.1 PostgreSQL.....	21
1.6 Entornos de desarrollo integrado.....	21
1.6.1 Eclipse.....	21
1.7 Metodología de desarrollo OpenUP y lenguaje de modelado UML.....	21

1.8 Herramientas CASE. Visual Paradigm.....	22
1.9 Conclusiones del capítulo.....	22
Capítulo 2. Requisitos.....	25
2.1 Modelo conceptual.....	25
2.2 Requisitos del sistema.....	26
2.2.1 Requisitos funcionales.....	27
2.2.2 Requisitos no funcionales.....	28
2.3 Casos de uso del sistema.....	29
2.4 Descripción de los casos de uso.....	30
2.5 Conclusiones del capítulo.....	31
Capítulo 3. Diseño.....	33
3.1 Arquitectura de software.....	33
3.1.1 Estilos y patrones arquitectónicos.....	33
3.1.2 Patrones de diseño.....	34
3.2 Diagramas de interacción.....	36
3.2.1 Diagramas de secuencia y colaboración.....	37
3.3 Modelo del diseño.....	39
3.4 Conclusiones del capítulo.....	41
Capítulo 4. Implementación y pruebas.....	43
4.1 Modelo de implementación.....	43
4.1.1 Diagrama de componentes.....	43
4.1.2 Modelo de despliegue.....	45
4.2 Pruebas de software.....	46
4.2.1 Pruebas de caja blanca.....	46

4.2.2 Pruebas de caja negra.....	47
4.3 Aplicación de pruebas de caja blanca.....	48
4.4 Casos de pruebas de caja negra.....	49
4.5 Conclusiones del capítulo.....	52
Conclusiones generales.....	53
Recomendaciones.....	55
Bibliografía.....	57
Anexos.....	61

Índice de figuras

Figura 1: Componentes básicos de una arquitectura SOA.....	9
Figura 2: Modelo cliente servidor.....	10
Figura 3: Programación Orientada a Aspecto.....	12
Figura 4: Anatomía de un portlet.....	16
Figura 5: Modelo conceptual del sistema.....	25
Figura 6: Diagrama de casos de uso del sistema.....	29
Figura 7: El patrón Modelo Vista Controlador.....	36
Figura 8: Diagrama de secuencia del CU Realizar alineamiento de secuencias.....	38
Figura 9: Diagrama de colaboración del CU Realizar alineamiento de secuencias.....	38
Figura 10: Diagrama de clases del diseño para el CU Realizar alineamiento de secuencias.....	39
Figura 11: Diagrama de clases del diseño para el paquete Objeto de Acceso a Datos.....	40
Figura 12: Diagrama de clases del diseño para el paquete Cliente de Servicios Web.....	41
Figura 13: Diagrama de componentes del CU Realizar alineamiento de secuencias.....	44
Figura 14: Modelo de despliegue del módulo básico.....	46
Figura 15: Función que genera secuencias de nucleótidos aleatoriamente.....	49
Figura 16: Camino básico de la función “makenucseq”.....	49
Figura 17: Diagrama del diseño para el CU Mostrar MAL de secuencias nucleotídicas.....	61
Figura 18: Diagrama de secuencia del CU Mostrar MAL de secuencias nucleotídicas.....	62
Figura 19: Diagrama de colaboración del CU Mostrar MAL de secuencias nucleotídicas.....	62
Figura 20: Diagrama de componentes del CU Mostrar MAL de secuencia nucleotídica.....	63
Figura 21: Diagrama de componentes del paquete Archivos del lado del cliente.....	63
Figura 22: Secuencia de proteína válida.....	64

Índice de tablas

Tabla 1: Descripción de los conceptos del dominio del módulo básico.....	26
Tabla 2: Descripción del CU Realizar alineamiento de secuencias.....	30
Tabla 3: Los patrones GRASP y sus funciones.....	35
Tabla 4: Descripción de los componentes del CU Realizar alineamiento de secuencias.....	44
Tabla 5: Variables para el caso de prueba Alinear con Clustalw.....	50
Tabla 6: Caso de prueba 1: Alinear secuencia con Clustalw.....	50
Tabla 7: Variables para el caso de prueba Generar secuencia nucleotídica aleatoriamente.....	51
Tabla 8: Caso de prueba 2: Generar secuencia nucleotídica aleatoriamente.....	51
Tabla 9: Descripción del CU Administrar alineamiento.....	65
Tabla 10: Descripción del CU Visualizar moléculas en tres dimensiones.....	66
Tabla 11: Descripción del CU Visualizar marcos abiertos de lectura.....	66
Tabla 12: Descripción del CU Visualizar propiedades químico-físicas de secuencias proteicas.....	67
Tabla 13: Descripción del CU Visualizar segmentos de transmembrana de proteínas.....	68
Tabla 14: Descripción del CU Generar secuencias aleatoriamente.....	68
Tabla 15: Descripción del CU Mostrar características básicas de una secuencia.....	69
Tabla 16: Descripción del CU Editar secuencia.....	70

Introducción

A lo largo de la historia de la humanidad se ha evidenciado cómo la ciencia ha constituido un factor esencial que permitió al hombre evolucionar a la sociedad actual, una sociedad en la que los avances tecnológicos marcan nuevos paradigmas en todas sus ramas. Los últimos avances científicos-técnicos y la gran cantidad de información existente en la actualidad han propiciado el surgimiento de una nueva era, la era de la información y el conocimiento. En este contexto, en el que la información crece a velocidades increíbles y se hace ineficiente su gestión, las Tecnologías de la Información y las Comunicaciones (TICs) han jugado un papel imprescindible.

Las TICs son aplicables en casi todos los sectores de la sociedad reportando continuas transformaciones en el modelo económico, social y cultural e incidiendo en la mayoría de los aspectos de la vida cotidiana como: la gestión económica y empresarial, las comunicaciones, la prestación de servicios, la salud y la educación, reportando en cada uno de ellos eficiencia en la gestión de la información y mejoras en la calidad de vida. La ciencia se ha apoyado en ellas para aplicar los conocimientos y aprovecharlos en el progreso de la humanidad. Ciencias como la Física, Química y Biología encuentran en las TICs la posibilidad de incrementar, mejorar y compartir su gama de conocimientos, y por ende contribuir al desarrollo de las mismas.

Con el objetivo de informatizar la sociedad, la Revolución Cubana ha apostado por el avance en el sector de la informática. El Comandante en Jefe Fidel Castro Ruz crea la Universidad de la Ciencias Informáticas con el fin de crear profesionales en este campo que a largo plazo permitieran aprovechar los beneficios económicos que la industria del software puede reportar al país. La UCI, a diferencia de la mayoría de los centros del país, cuenta con un número significativo de recursos computacionales y tecnológicos, como por ejemplo, ordenadores para realizar cálculos de gran magnitud, acceso a Internet para descargar y utilizar herramientas en línea, además de varias herramientas desarrolladas por la propia institución con el objetivo de satisfacer necesidades en varios sectores de la sociedad.

Teniendo en cuenta la importancia que representa para el país el desarrollo científico en áreas del conocimiento emergentes como la Bioinformática, se crea en la UCI el Departamento de Bioinformática, el cual forma parte de las líneas de producción del centro DATEC de la Facultad 6. En este departamento se cuenta con varias herramientas enfocadas al procesamiento de datos

biológicos como BioSyS para la simulación de sistemas biológicos a través de ecuaciones diferenciales, T-Arenal, una plataforma para realizar cálculos de manera distribuida aprovechando los recursos de cómputo de la universidad y acortando por ende el tiempo de ejecución de una tarea y siRNADesign para el diseño de sondas que silencian genes patógenos humanos. Todas ellas deben integrarse en la Plataforma de Servicios Bioinformáticos de la UCI que a través de un portal web permitirá el acceso a las herramientas y bases de datos existentes; así como la formulación, composición y ejecución de problemas para la visualización y anotación de los resultados. Esta plataforma actualmente se encuentra en desarrollo y pretende brindar servicios a los especialistas de universidades y centros del país que trabajan en el área de la Bioinformática.

Por su parte la Bioinformática está considerada como una ciencia emergente donde convergen diferentes áreas del conocimiento tales como la Biología, la Química, la Física, la Matemática y la Informática. A pesar de ser un ciencia relativamente nueva, ha sido reconocida como un campo crucial para el desarrollo científico, especialmente para países en desarrollo como Cuba[1]. En la actualidad el procesamiento de la información biológica es cada vez más complejo, necesitando un alto nivel de cómputo, debido a la complejidad creciente de los algoritmos y métodos estadísticos. La meta de esta disciplina científica es permitir el descubrimiento de nuevas ideas, así como ofrecer una perspectiva global a partir de la cual se puedan discernir nuevos principios y paradigmas. En los comienzos de la revolución genómica, la Bioinformática se restringía prácticamente a la creación y mantenimiento de bases de datos para almacenar información biológica, generalmente en la forma de secuencias nucleotídicas y aminoacídicas. El desarrollo de este tipo de bases de datos implicaba, también, el diseño de interfaces complejas que permitieran el acceso a la información, la adición de nuevos datos y la revisión de los ya existentes.

Entre las líneas de investigación más importantes de la Bioinformática se destaca el análisis de secuencias moleculares, dentro de la cual se incluye: la detección de relaciones filogenéticas, la ingeniería genética usando el mapeo de sitios de restricción, la determinación de la estructura genética mediante la predicción de intrones/exones y la inferencia de secuencias codificantes de proteínas por medio del análisis de los marcos abiertos de lectura.

Una tarea básica dentro del análisis de secuencia es la realización de alineamientos de secuencias. Este es el procedimiento de comparar dos o más secuencias para buscar una serie de caracteres individuales o patrones de caracteres que estén en el mismo orden o posición dentro de las mismas. Los alineamientos de secuencias son extremadamente útil para obtener información evolutiva, estructural y funcional de las secuencias biológicas. Una de las mayores dificultades

encontradas a la hora de utilizar herramientas para el análisis de secuencias es la existencia de diferentes formatos. Estos difieren en la presencia de caracteres y palabras que indican los variados tipos de información que puede ser encontrada dentro de un fichero. Entre los formatos más utilizados se encuentran: GenBank, EMBL, Fasta y GDG, es por ello que la conversión de formatos es de gran utilidad dentro de este tipo de análisis.

Uno de los objetivos fundamentales de la bioinformática está en comprender la relación entre una secuencia amino-acídica y la estructura tridimensional de la proteína que codifica[2], en este sentido los visualizadores moleculares juegan un papel fundamental ya que permiten representar estructuras proteicas en el espacio a partir de las coordenadas tridimensionales de sus aminoácidos a través de archivos PDB (siglas en inglés de Protein Data Bank).

Antes de que una proteína sea sintetizada, su secuencia debe ser traducida desde el ADN. La traducción de ADN a proteína no es conceptualmente difícil. Sólo se requiere de una secuencia de ADN, un código genético y un programa que lea una secuencia nucleotídica y de como salida una secuencia proteica. Existen seis maneras posibles de traducir una secuencia de ADN, a estas se le denominan pautas o marcos abiertos de lectura (ORF, siglas en inglés de Open Reading Frame).

En la actualidad existe gran diversidad de recursos bioinformáticos, tales como: bases de datos, servidores, herramientas de software y algoritmos que involucran un gran número de datos con formatos heterogéneos. La mayoría de las herramientas y servicios bioinformáticos para realizar análisis de secuencia se encuentran en disímiles servidores internacionales solo accesibles a través de la red de redes, por lo tanto, el hecho de estar integradas a un portal nacional representaría una ventaja debido al acceso restringido a Internet de la mayoría de los centros de investigación y universidades del país.

Es por ello que el **problema a resolver** de la presente investigación es: ¿Cómo garantizar a los especialistas del país el manejo de herramientas básicas para el análisis de secuencias moleculares a través de la Plataforma de Servicios Bioinformáticos de la Universidad de las Ciencias Informáticas?

Determinando como **objeto de estudio**: los servidores y aplicaciones bioinformáticas para el procesamiento de datos biológicos y como **campo de acción**: el análisis computacional de secuencias moleculares.

El **objetivo general** del presente trabajo es desarrollar el módulo básico de la Plataforma de Servicios Bioinformáticos de la UCI, del cual se derivan los siguientes **objetivos específicos**:

- Definir los requisitos funcionales del módulo básico.
- Diseñar el módulo básico.
- Implementar el módulo básico.
- Validar el módulo.

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

- Análisis del estado del arte de las principales metodologías de desarrollo de software, herramientas y tecnologías para la implementación del sistema.
- Identificación de los requisitos funcionales del módulo básico.
- Selección de los patrones de diseño a utilizar en el módulo básico.
- Implementación de los requisitos funcionales identificados.
- Selección de los métodos y técnicas para probar los servicios implementados.
- Aplicación de las pruebas a los servicios implementados.

La estructura del documento se resume en los siguientes acápite:

Capítulo 1. Fundamentación teórica: Abarca temas relacionados con la evolución de la World Wide Web y las principales tecnologías usadas en su confección. Se evalúan los principales sitios que proveen recursos bioinformáticos en Internet y se fundamenta la selección de las tecnologías usadas en la realización del sistema.

Capítulo 2. Requisitos: Trata aspectos relacionados con el análisis del sistema. Se describirán las características del módulo básico y se identifican y describen las funcionalidades que debe poseer el mismo.

Capítulo 3. Diseño: Describe el diseño de la solución propuesta. Brinda una descripción de los estilos y patrones arquitectónicos más utilizados y la aplicación de los mismos para la confección de los diagramas de interacción de los objetos y el modelo del diseño del sistema.

Capítulo 4. Implementación y pruebas: Relaciona los contenidos de los capítulos anteriores y da paso a la realización del modelo de implementación para desarrollar el sistema propuesto. Además recoge los modelos de prueba que verifican los requisitos de calidad de la solución.

Capítulo 1. Fundamentación teórica

Durante las últimas décadas el mundo se ha enfrentado a continuos cambios en la esfera científico-técnica, la era del conocimiento está transformando los sistemas económicos, políticos y sociales. Las TICs e Internet como plataforma internacional de intercambio de información juegan un papel determinante en la actualidad. Este capítulo engloba conceptos básicos relacionados con la evolución de Internet, las ventajas de la Web y las principales tecnologías que se utilizan para el desarrollo de la misma. Se enfatiza también en el empleo de las TICs para el procesamiento, gestión y análisis de datos biológicos y la explotación de estas tecnologías para diseminar e intercambiar el conocimiento biológico. Además se analizan las principales herramientas para el desarrollo de aplicaciones web y se fundamenta la selección de estas para el desarrollo de la solución propuesta.

1.1 Evolución de la World Wide Web

Para poder utilizar los ordenadores y todas las funcionalidades que estos nos brindan fue necesario una revolución de software. En un principio el software se distribuía en soporte digital (discos, memorias, etc.) y luego se instalaba en el ordenador para poder usarlo, pero dada la variedad de sistemas operativos existentes muchas de estas aplicaciones no se podían ejecutar en todos los ambientes.

Con el desarrollo de Internet, los negocios, el comercio y en general el flujo de información, el panorama cambió, posibilitando al usuario poder utilizar aplicaciones y sistemas de toda índole para satisfacer sus necesidades. Las primeras prestaciones que se brindaron en Internet fueron el correo electrónico y los boletines de noticias, sin embargo el auge de la red y el desarrollo de nuevas tecnologías y protocolos provocó el desarrollo de nuevas funcionalidades, hoy en día podemos disfrutar de foros de discusión, mensajería instantánea, distribución de archivos, buscadores de contenido, música y televisión en línea, sindicación de contenido, entre muchas otras prestaciones.

La necesidad de agrupar estas funciones en un todo, dio origen a aplicaciones de mayor nivel, estas últimas integraban todos los servicios de manera tal que el usuario pudiera utilizarlos desde un solo sitio virtual. Esto permitió la creación de sistemas web complejos, unos destinados al usuario y otros destinados a otras aplicaciones, de esta manera las aplicaciones web se clasifican en dos tipos teniendo en cuenta el usuario final[3]:

- **Orientadas a la presentación:** las aplicaciones web orientadas a la presentación generan páginas web interactivas de forma dinámica que contienen varios tipos de lenguaje de marcado (HTML, XML, etc.) y contenido dinámico (imágenes, texto, videos, entre otros).
- **Orientadas a servicios:** una aplicación web orientada a servicios implementa determinadas funciones a las que se accede por un punto de enlace que posibilita a otras aplicaciones utilizarlas (comúnmente aplicaciones web orientadas a la presentación).

Las aplicaciones web orientadas a servicios no están destinadas al usuario final, ocurriendo lo contrario en el caso de las aplicaciones orientadas a la presentación, que poseen una interfaz mediante la cual se le brinda al usuario las funcionalidades y servicios. Dentro de este tipo de aplicaciones se encuentran: las redes sociales, tiendas virtuales, aplicaciones de correo, bitácoras personales, sitios de carácter informativo y los portales interactivos.

Internet es la herramienta más potente de las TICs en la era de la información y el conocimiento, y cuando se habla del procesamiento, gestión y análisis de la información biológica se hace imposible ignorarlo, ya que como plataforma de intercambio de información es extremadamente útil y se convierte en imprescindible para ciencias como la Bioinformática.

1.1.1 Recursos bioinformáticos en la Web

La reconocida revista *Nucleic Acids Research* de la Universidad de Oxford, dedica cada año dos artículos a actualizar el listado de las bases de datos y servidores de aplicaciones disponibles. En su número correspondiente al mes de julio de 2011, reportan un total de 1250 servidores de herramientas, 144 recursos y 480 bases de datos del Directorio de Vínculos Bioinformáticos[4]. En su número más reciente publicado en enero del presente año, dedicado a las bases de datos, informan la exclusión de 20 bases de datos obsoletas y la inclusión de otras 30, reportando un total de 1380 bases de datos organizadas en 14 categorías y 41 subcategorías[5].

Existen varios portales web de recursos bioinformáticos que proveen a través de sus páginas numerosos servicios. Entre los sitios en línea más populares a nivel mundial se encuentran: el Centro Nacional para la Información Biotecnológica (NCBI, por sus siglas en inglés) el cual está

disponible en <http://www.ncbi.nlm.nih.gov/>, el Expasy (acrónimo de Expert Protein Analysis System) disponible en <http://expasy.org/>, el Instituto Europeo de Bioinformática (EBI, por sus siglas en inglés) disponible en <http://www.ebi.ac.uk> y el Banco de Datos de ADN de Japón (DDBJ por sus siglas en inglés) disponible en <http://www.ddbj.nig.ac.jp/>.

El NCBI es una división de la Biblioteca Nacional de Medicina del Instituto Nacional de Salud de los Estados Unidos que posee sistemas automatizados para el almacenamiento y análisis de datos biológicos, bioquímicos y genéticos[6]. Contiene una base de datos pública de secuencias nucleotídicas llamada GenBank y brinda accesos a PubMed, un buscador de literatura biomédica sobre la base de datos Medline, una de las más grandes en este sentido. Además permite la consulta de un catálogo de enfermedades genéticas humanas a través de la base de datos OMIM. Todos estos recursos pueden ser accedidos libremente y para ello se dispone de un motor de búsqueda llamado Entrez. El NCBI ofrece además algunos servicios bioinformáticos en línea para el análisis de secuencias de ADN, ARN y proteínas.

Otro de los portales de Bioinformática más utilizado por la comunidad científica es el del EBI, un instituto académico localizado en Cambridge, Reino Unido, que forma parte del Laboratorio de Biología Molecular de Europa (EMBL de sus siglas en inglés, European Molecular Biology Laboratory) cuyo objetivo es brindar a la comunidad científica el acceso a datos y recursos bioinformáticos[7]. Entre sus principales servicios se encuentran el alineamiento múltiple de secuencias, el análisis funcional de proteínas, de genomas completos, de estructuras moleculares y búsquedas de secuencias por similitud.

Por su parte el Expasy, es un portal de recursos bioinformáticos especializado en el análisis de las proteínas, brinda acceso a SwissProt y TrEMBL las dos bases de datos principales de secuencias proteicas a nivel mundial así como a un conjunto de herramientas en línea para realizar estudios sobre las proteínas[8]. Dentro de sus servicios más comunes se encuentran: la generación de modelos por homología, el análisis y búsqueda de similitud entre secuencias y la traducción de secuencias nucleotídicas. Además contiene catálogos de información y paquetes de herramientas de propósito general.

Por último el Banco de Datos de ADN de Japón, desde sus comienzos ha funcionado como una base de datos internacional de secuencias nucleotídicas que intercambia información con el NCBI y el EMBL a través del INSDC (International Nucleotide Sequence Database Collaboration). El mismo está asesorado por el Centro de Información Biológica (CIB) del Instituto Nacional de

Genetistas de Japón y tiene su sede en la ciudad de Mishima. Contiene además una serie de herramientas y servicios que permiten realizar análisis sobre la información biológica.

De manera general estos sitios disponen de servicios para editar y manipular secuencias tales como: inversión, extracción de regiones de interés, interconversión de formatos, predicción de segmentos de transmembrana y de marcos abiertos de lectura, traducción a proteína, entre otros. Todos ellos están disponibles como aplicaciones independientes, aunque también existen paquetes de herramientas análogas tales como el EMBOSS[9]. Específicamente para los alineamientos de secuencias comúnmente se utilizan: Clustalw[10], Muscle[11], T-Coffee[12] y Fasta, cada uno basados en algoritmos diversos para alinear secuencias, diferenciándose en el balance calidad-rendimiento.

1.1.2 Servicios web en la gestión de la información biológica

La web se ha convertido en una plataforma de desarrollo y colaboración en el área de la Bioinformática, y se verifica en la gran variedad de funcionalidades que se brindan a través de los principales sitios especializados en este sector. Las limitaciones de las herramientas en línea basadas en HTML han dado pie a otro tipo de soluciones en este área, tal es el caso de las aplicaciones basadas en XML (acrónimo del inglés eXtensible Markup Language), un lenguaje extensible y estándar que permite el intercambio de información entre aplicaciones. Una de las tendencias actuales, dada la dificultad de integrar las funcionalidades de diversas aplicaciones y la gran cantidad de datos que estas intercambian, es la creación de web services (servicios web, en español). Estos permiten la interacción computadora-computadora para realizar operaciones de cómputos que procesen gran cantidad de información, y además brindan la posibilidad de crear aplicaciones más complejas que reutilicen las funcionalidades de otras, logrando de esta manera gran interoperabilidad entre ellas e independencia de las plataformas en la que están construidas.

Internet ofrece acceso a variados recursos y servicios web para la gestión y análisis de datos biológicos, todos ellos disponibles a través de los portales antes mencionados ubicados en países desarrollados como Estados Unidos, Reino Unido de Gran Bretaña, Alemania y Japón[13].

El EBI, por ejemplo contiene una serie de funciones implementadas como servicios web[14], entre ellos están: la búsqueda de información en bases de datos, la utilización de herramientas de análisis, búsqueda por similitud en secuencias, realización de análisis filogenéticos y alineamientos múltiples de secuencias. Por su parte el NCBI provee también servicios web[15], entre las principales funcionalidades que este brinda se encuentran la extracción de datos de Pubmed,

OMIM, Gene y Sequences. El DDBJ brinda también servicios web[16], que permiten realizar búsqueda y recuperación de archivos PDB (siglas en inglés de Protein Data Bank), búsqueda de secuencias nucleotídicas, alineamiento múltiple de secuencias y análisis de genomas completos.

1.1.3 Arquitectura Orientada a Servicios

A pesar de las grandes ventajas que poseen los servicios web tales como la usabilidad y la reutilización combinadas en la interoperabilidad, que es la característica más importante que proveen los mismos; estos presentan ciertas limitaciones y desventajas, como bajo nivel de abstracción, intercambio de grandes cantidades de datos sobre protocolos simples como HTTP, críticos problemas de seguridad y bajo rendimiento. Para aprovechar el potencial de los servicios web estos deben ser diseñados cuidadosamente bajo los conceptos y principios de la orientación a servicios propuestos por la Arquitectura Orientada a Servicios (SOA por sus siglas en inglés).

SOA mantiene la filosofía de los servicios web, pero provee además una serie de conceptos que garantizan un mejor diseño de aplicaciones en cuanto a seguridad, administración de servicios, confiabilidad, separación en unidades lógicas de procesamiento y calidad de los servicios brindados[17]. Los componentes básicos de SOA se representan en la Figura 1.

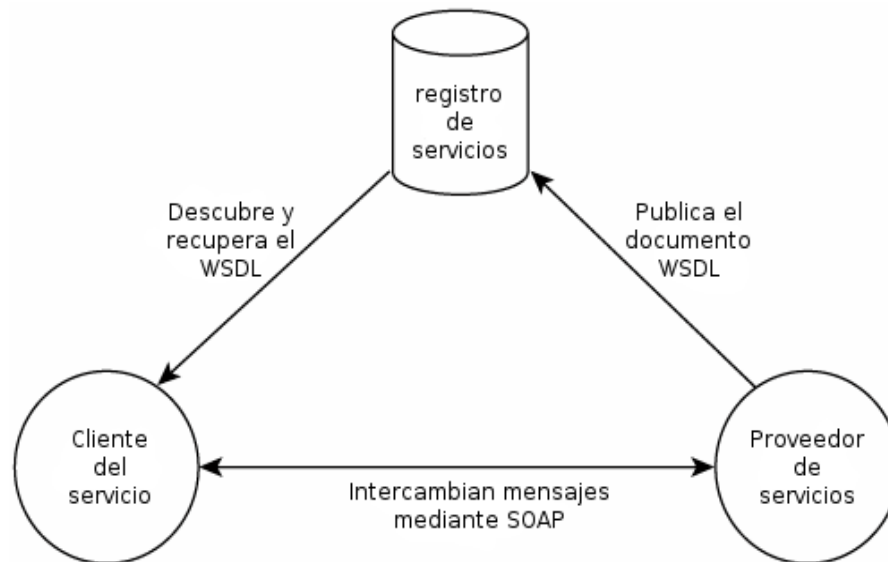


Figura 1: Componentes básicos de una arquitectura SOA. El proveedor de servicios utiliza el registro UDDI para publicar las funcionalidades, este registro proporciona a los clientes la información necesaria para invocar las funciones implementadas como servicios web.

SOA organiza las funcionalidades en servicios web que pueden ser consumidos por aplicaciones clientes, las cuales intercambian información con el proveedor de servicios mediante el Protocolo Simple de Acceso a Objeto (SOAP por sus siglas en inglés). El proveedor publica los

servicios en un registro denominado UDDI (siglas en inglés de Universal Description, Discovery and Integration), el cual especifica mediante documentos descriptores de servicios web (WSDL por sus siglas en inglés) las funciones y los parámetros necesarios para poder utilizar el servicio web. Para la implementación del módulo básico se optó por el uso de una Arquitectura Orientada a Servicios siguiendo los principios de orientación a servicios que esta propone.

1.2 Técnicas y lenguajes de programación para el desarrollo de aplicaciones web

El éxito espectacular de la web se basa en dos puntales fundamentales: el protocolo HTTP y el lenguaje HTML. Uno permite una implementación simple y sencilla de un sistema de comunicaciones, mientras el otro proporciona un mecanismo de composición de páginas enlazadas de forma fácil y altamente eficiente[18]. Ambos son los componentes fundamentales del modelo cliente-servidor (Figura 2).

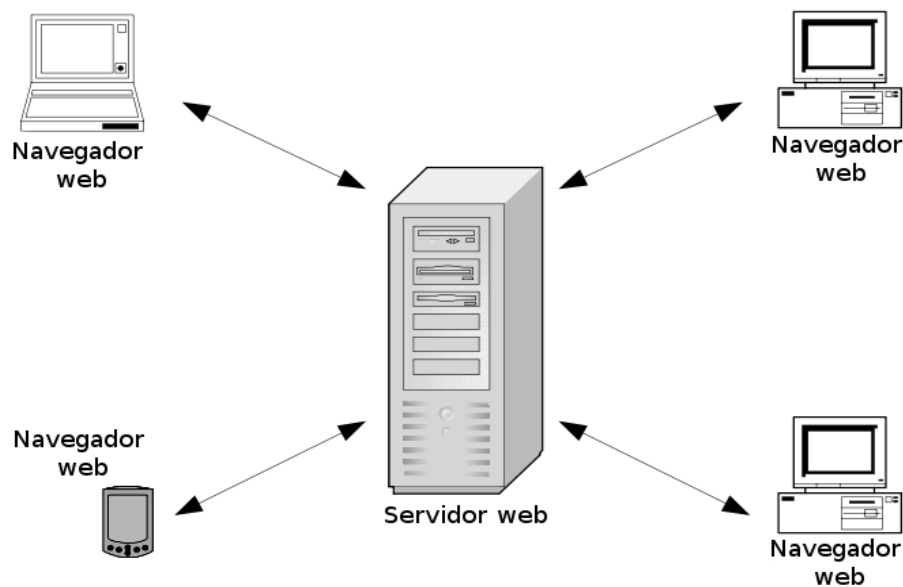


Figura 2: Modelo cliente servidor. El servidor web atiende las peticiones que el usuario realiza a través de un navegador web, el servidor responde en formato HTML y el navegador muestra al usuario los elementos que componen la página.

El protocolo de transferencia de hipertexto o HTTP (Hyper Text Transfer Protocol por sus siglas en inglés) es la base de la red de redes, mediante este los usuarios haciendo uso de un navegador web pueden realizar peticiones a un servidor. El lenguaje de marcado de hipertexto o HTML (Hyper Text Markup Language en inglés) es el lenguaje que se usa para mostrar los componentes de una página web, es interpretado por el navegador que convierte las sentencias de hipertexto en ricas páginas web.

1.2.1 Programación Orientada a Objeto y Orientada a Aspecto

Entre las técnicas más comunes para desarrollar aplicaciones se encuentra el paradigma de Programación Orientado a Objetos (OOP, siglas en inglés de Object Oriented Programming), una técnica que usa objetos y sus interacciones para diseñar aplicaciones y programas de computadora. Este paradigma proporciona una forma particular de programar, más cercana a la manera de cómo se expresan los objetos en la vida real.

La OOP divide la implementación de sistemas en componentes denominados objetos, los cuales encapsulan sus datos y procedimientos a otros objetos. Los objetos son instancias de una clase, una colección de objetos similares que combinan estado, comportamiento e identidad. Un sistema orientado a objetos reutiliza funcionalidades comunes mediante técnicas como la herencia y la delegación de responsabilidades, pero en determinadas situaciones es necesario tratar aspectos del sistema en el que estas técnicas pueden traer consecuencias negativas como alto acoplamiento y baja reusabilidad.

Los conceptos aglutinados dentro de la Programación Orientada a Aspectos (AOP, siglas en inglés de Aspect Oriented Programming), tratan de resolver problemas en el desarrollo de software en los que la OOP no se puede aplicar o resulta contradictorio su aplicación.

La AOP solventa estas dificultades separando las funciones comunes en aspectos (Figura 3). Un aspecto es una unidad modular y funcional diseminada en otras unidades funcionales del sistema[19]. Los aspectos son implementados con otros conceptos como punto de unión, punto de cruce y consejo. Un punto de unión es una parte en la ejecución de la aplicación (autenticarse, generar reporte, etc.) en donde un aspecto puede ser aplicado, puede ser la llamada a un método, el lanzamiento de una excepción o incluso cuando un campo haya sido modificado.

Es imposible aplicar AOP en todas las partes de una aplicación, la tarea de definir dónde aplicar un aspecto es del punto de cruce, que no es más que la unión de uno o más puntos de unión. Por su parte, el consejo, es quién define el trabajo de un aspecto, y responde a la pregunta de cuándo aplicarlo: antes de invocar un método, después de que este retorne o cuando lance alguna excepción.

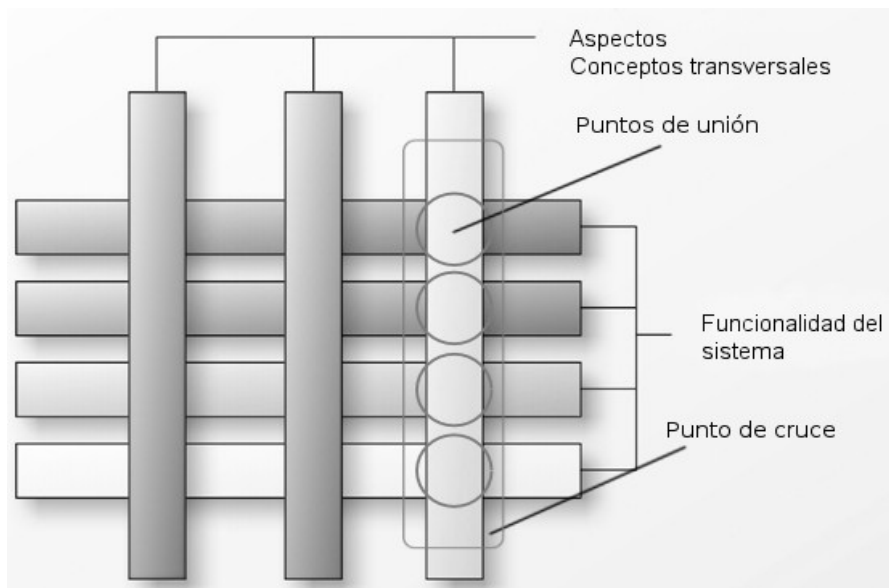


Figura 3: Programación Orientada a Aspecto. Las funcionalidades de una aplicación en ocasiones comparten el mismo código, lo que se denomina aspecto en AOP. Los puntos de unión es la intersección del mismo código en diferentes funciones del sistema.

La AOP no constituye un nuevo paradigma, sin embargo ofrece un complemento a la OOP que permite reutilizar más código y una forma más eficaz de implementar funcionalidades transversales en una aplicación.

1.2.2 Lenguajes y plataformas de desarrollo. Java 2 Enterprise Edition

Actualmente existen muchos lenguajes de programación para desarrollar aplicaciones web, dependiendo del lado en el que se ejecutan, se pueden clasificar en lenguajes del lado del cliente o lenguajes del lado del servidor, siendo estos últimos los más importantes en el desarrollo de aplicaciones, entre los más utilizados se pueden citar PHP, Python, C# y Java[20].

PHP o Preprocesador de Hipertexto, es un lenguaje de código abierto, originalmente diseñado para ser usado en el desarrollo de aplicaciones web. PHP creció rápidamente hasta convertirse en el lenguaje que es hoy, adquiriendo el nombre de PHP, para representar sus habilidades de procesar páginas web antes de ser mostradas[21].

PHP no necesita ser compilado, y el código está embebido en las páginas HTML por lo que el proceso de cargar las páginas es bastante corto. Se puede obtener sin necesidad de pagar ninguna licencia, es multiplataforma, posee marcos de trabajo bien conocidos como Symfony, ZendFramework y PHPNuke y también cuenta con Sistemas de Manejo de Contenidos (CMS por sus siglas en inglés) muy populares como Drupal, Joomla y WordPress.

Otro de los lenguajes muy populares es Python, con él no es necesario declarar el tipo de dato que va a contener una determinada variable, sino que su tipo se determinará en tiempo de ejecución según el tipo de valor que se le asigne. El intérprete de Python es multiplataforma ya que está disponible en multitud de plataformas (UNIX, Windows, Mac OS, etc.) Está orientado a objetos y cuenta con varios marcos de trabajo que facilitan el desarrollo de las aplicaciones web, entre ellos están Django, TurboGears y Pylons[22].

La compañía Microsoft provee una plataforma de desarrollo web denominada ASP.NET. Una tecnología para brindar interacción con datos manejados por aplicaciones web sobre Internet e Intranets. Abarca un gran número de lenguajes de programación entre los que se encuentran C, C# y C++. Incluye varios controles prefabricados, como cajas de texto, botones, imágenes, y datos cuadrículados, que pueden ser ensamblados, configurados y manipulados con código para crear páginas HTML procesadas por los navegadores web[23].

ASP presenta una arquitectura donde separa conceptos de aplicaciones mediante la adopción del patrón Modelo Vista Controlador (MVC). Provee una suite de desarrollo madura, que incluye una serie de librerías para hacer el desarrollo fácil y rápido. Presenta un marco de trabajo, con múltiples funcionalidades para autenticar, gestionar perfiles e internacionalizar el contenido. Además cuenta con un Entorno de Desarrollo Integrado (IDE, siglas del inglés Integrated Development Environment) muy completo como Visual Studio. ASP.NET está licenciado bajo Ms-PL, una Iniciativa de Código Abierto (OSI, siglas de Open Source Initiative) aprobada bajo licencia de código abierto. También se pueden encontrar gran variedad de libros, folletos, tutoriales y artículos acerca de ASP.NET, esta plataforma cuenta con gran aceptación internacional y dispone de documentación en varios idiomas[24].

Probablemente el lenguaje de programación más utilizado para la creación de software es Java. La JRS-58 (Java Specification Request 58), más conocida como Plataforma Java 2 Enterprise Edition (J2EE), es la extensión de Java para el desarrollo web, representa la evolución de la plataforma de desarrollo del lado del servidor hacia una especificación madura y sofisticada[25].

J2EE es considerada la mejor alternativa a la tecnología ASP.NET de Microsoft[20]. Las aplicaciones desarrolladas bajo esta plataforma pueden ser desplegadas en cualquier sistema operativo. J2EE utiliza el lenguaje de programación Java, el cual está licenciado bajo licencia GNU GPL, no siendo así con las librerías necesarias para ejecutar el código, conocido como Paquete de Desarrollo de Java (JDK siglas en inglés de Java Development Kit), sin embargo se dispone de

una versión completamente libre conocida como OpenJDK (<http://openjdk.java.net/>). J2EE es una unidad de software funcional que está ensamblada dentro de las aplicaciones Java con sus clases relacionadas y archivos que lo comunican con otros componentes, dispone de varias herramientas de código abierto como IDEs, servidores, marcos de trabajo y APIs (siglas en inglés de Applications Programming Interface). Posibilita una serie de librerías que facilitan el trabajo de los desarrolladores, tales como Java Database Connectivity (JDBC), Enterprise Java Beans (EJB), Java Server Pages (JSP), Java Tag Library (JSTL) y Java Message Service (JMS). Cuenta con varios marcos de trabajo que facilitan el desarrollo de las aplicaciones, tales como Spring, Struts y Java Server Faces. Tiene además varios CMS con muchas funcionalidades agregadas, entre ellos se pueden mencionar Liferay Portal, Apache Jeetspeed y JBoss Portal. Tanto para la plataforma J2EE y para el lenguaje Java en general existen múltiples bibliografías en varios idiomas y se han creado comunidades que promueven la utilización de esta tecnología.

Teniendo en cuenta lo anteriormente planteado se decide utilizar la plataforma J2EE para el desarrollo de la solución, destacando como ventaja sustancial la reutilización de código que ella permite, ya que el Departamento de Bioinformática de la UCI cuenta con varias herramientas implementadas en Java.

1.2.3 Algunos componentes de la plataforma Java 2 Enterprise Edition

Para el correcto funcionamiento de la plataforma J2EE se necesitan una serie de componentes, entre los que vale destacar los contenedores web. Un contenedor J2EE es un período de ejecución para gestionar los componentes de la aplicación desarrollados según las especificaciones del API y destinados a proporcionar acceso a los APIs de J2EE[25]. Un contenedor web tiene la capacidad de recibir peticiones desde un navegador y generar una respuesta, llamando a componentes de aplicaciones como por ejemplo los servlets, las páginas JSP y los componentes EJB.

La primera generación de tecnologías para desarrollar aplicaciones web en Java fueron los servlets. Un Servlet es una clase del lenguaje de programación Java que es usada para extender la habilidad de los servidores de albergar aplicaciones que son accesibles bajo el modelo de programación petición-respuesta[3]. Implementa la interfaz `HttpServlet` del API Java Servlet. Esta interfaz define un conjunto de métodos para proporcionar respuestas dinámicas a solicitudes HTTP. Un servidor de aplicación compatible con J2EE proporciona a los servlets un entorno en el que residir y administrar las solicitudes entrantes[26].

El principal problema que presenta el uso de servlets es que la página HTML debe ser escrita en el código de la aplicación utilizando excesivas sentencias de salida con el método *"println"*. De esta forma se generaba mucho código innecesario y se hacía muy complicado crear estéticas interfaces de usuario. Además trae consigo dependencia entre el código HTML y el código de la aplicación en Java, lo que constituye esfuerzo adicional por parte de los programadores. Este conjunto de deficiencias marcaron el surgimiento de nuevas tecnologías como las páginas JSP.

La tecnología Java Server Pages, permite crear contenido tanto dinámico como estático de forma fácil, separando el código de la interfaz de usuario de la lógica de negocio. JSP provee un lenguaje para escribir documentos simples, brinda una serie de expresiones que mejoran la interacción con el servidor que viajan dentro de los objetos *"request"* y *"reponse"*, además de mecanismos para definir extensiones del lenguaje. Una página JSP es un documento que contiene dos tipos de texto: datos estáticos, que pueden ser expresados en cualquier formato de texto (como son HTML, WML, y XML), y elementos JSP, que construyen contenido dinámico[3].

JSP es un gran paso de avance, sin embargo muchas veces se necesita incluir código Java dentro de la página para realizar acciones específicas, como iterar sobre una lista de valores, establecer una condición o conectarse a una base de datos. Teniendo en cuenta que este tipo de operaciones puede repetirse en varias páginas JSP, es necesario repetir el código en cada una de ellas. Para solventar esta desventaja se desarrolló una tecnología denominada JSTL.

La biblioteca de etiquetas estándares de JSP (JSTL siglas en inglés de JSP Standar Tag Library) encapsula funcionalidades comunes para muchas aplicaciones JSP. En vez de mezclar numerosas operaciones en tu aplicación JSP, JSTL permite una simple y estándar lista de etiquetas. Esta generalización permite desarrollar y desplegar aplicaciones JSP sobre cualquier contenedor web[3]. Las JSTL permiten iterar sobre listas de elementos, insertar condicionales, trabajar con formularios y además brinda la posibilidad de definir nuevas etiquetas personalizadas para funciones específicas.

Dentro de J2EE existen componentes que permiten combinar varias potencialidades de la plataforma, incluyendo el desarrollo de interfaces de usuario, ejemplo de ello lo constituyen los portlets. Un Portlet (Figura 4), es un componente de interfaz de usuario reutilizable, que provee una vista de la información del sistema. Estos componentes producen fragmentos de lenguaje de marcado, que los dispone para ser agregados dentro de páginas en portales[27].

Para que un portlet pueda mostrarse, debe existir un contenedor de portlets, que cumpla con las JSR (siglas en inglés de Java Specification Request) 168 y 286. Un contenedor de portlets es el entorno de ejecución de un portlet, este maneja su ciclo de vida, sus características, como deben lucir y manipula las peticiones desde el portal[27].

Una de las potencialidades de los portlets es su capacidad de transportarse sin necesidad de cambiar su código, los portlets se compilan en un Archivo de Aplicación Web (WAR siglas en inglés de Web Application Archive), el cual puede ser desplegado en otro contenedor de portlets que sean compatibles con las JSR 168 y 286, para su desarrollo se utilizan Servlets, páginas JSP y las JSTL.



Figura 4: Anatomía de un portlet. Estos componentes permiten personalizar la forma en que se muestran, las decoraciones a menudo son opciones de navegación (ayuda, edición o vista) o estados (minimizado, maximizado o normal). El contenido de un portlet es también una página web ya que produce código HTML.

Otro aspecto importante es la gran interactividad de estos componentes, ya que pueden ser minimizados, maximizados o restaurados como una ventana tradicional, incluso pueden cerrarse para después ser mostrados nuevamente. Además ofrecen diferentes modos de navegación como ayuda, vista y edición.

1.3 Contenedores de Portlets en la plataforma J2EE

En la plataforma J2EE existen varios contenedores de portlets que son capaces de manejar el ciclo de vida de un Portlet, entre los más utilizados están Apache Jetspeed, JBoss Portal y Liferay Portal. El primero es una plataforma de información basada en un portal, escrito completamente bajo código abierto[28], el segundo proporciona un entorno de código abierto, para alojar y servir aplicaciones en una interfaz web de portal, es además un CMS con múltiples funciones[29], mientras el último, goza de prestigio a nivel internacional y entre los casos de éxitos que utilizan este portal se pueden mencionar empresas como: AutoZone (<http://www.autozone.com/>), Movistar (<http://www.movistar.es/on/>) y T-Mobile (<http://www.t-mobile.cz/>).

1.3.1 Liferay Portal

Liferay Portal es un portal de gestión de contenidos de código abierto implementado en Java y su contenido está basado en portlets. Incluye muchas aplicaciones tales como mensajería instantánea, foros y biblioteca de documentos. Soporta múltiples bases de datos como: PostgreSQL, MySQL, Oracle, SQL Server y Sybase, además se puede desplegar en muchos servidores como Jetty, JBoss, Sun GlassFish y Apache Tomcat, es multiplataforma ya que puede ejecutarse en cualquier sistema operativo[30]. Liferay es un proyecto de código abierto que usa licencia LGPL, esto implica que Liferay Portal puede ser vinculado con software que no sea de código abierto y con fines lucrativos[31]. El portal provee la personalización mediante plantillas y temas que pueden ser instalados a partir de un archivo WAR.

Liferay Portal fue nombrado el mejor producto en portal horizontal en el año 2011, en la publicación anual a las mejores soluciones Gartner's Magic Quadrant de la empresa estadounidense de consultoría e investigación de las TICs: Gartner, además fue nombrado el CMS más popular escrito en Java por Water & Stone en el 2010 y seleccionado por InfoWorld, el mejor portal de código abierto en el mercado[32]. Actualmente cuenta con una amplia gama de herramientas y complementos que se pueden adherir a entornos de desarrollos integrados de la plataforma Java como Eclipse y NetBeans.

1.4 Marcos de trabajo

El desarrollo de aplicaciones promueve la reutilización de código, pues muchas de estas tienen funcionalidades iguales o muy parecidas a las de otra. El objetivo es no reinventar la rueda y aprovechar funciones generales o que ya están implementadas, a menudo estas funciones son agrupadas en librerías o marcos de trabajo.

Un marco de trabajo (framework, en inglés) es una pieza de software estructural. Se dice estructural porque estructura es el objetivo de un framework que especifica todo requerimiento funcional. Un marco de trabajo trata de hacer generalizaciones acerca de las tareas comunes e intenta proveer una plataforma donde las aplicaciones pueden ser rápidamente construidas[33].

En la plataforma J2EE existen marcos de trabajos para diversos propósitos, dígase manipular bases de datos, desarrollar aplicaciones web, visualizar datos y gestionar peticiones AJAX (acrónimo del inglés Asynchronous JavaScript And XML).

1.4.1 Marcos de trabajos de mapeo relacional. Hibernate

Una herramienta de mapeo relacional u ORM (del inglés Object Relational Mapping) es un marco de trabajo para gestionar bases de datos que brinda una mayor abstracción del modelo de base de datos, lo cual permite manejar la información sin tener conocimiento de un determinado lenguaje SQL. Existen muchas herramientas ORM para Java, entre las más populares está Hibernate. Esta provee un puente entre la base de datos y la aplicación para almacenar objetos persistentes, posibilita al desarrollador escribir menos código y gestionar con mucha facilidad el modelo de datos[34].

Hibernate posee un lenguaje de consultas propio llamado HQL (siglas del inglés Hibernate Query Language). Para utilizar HQL no es necesario tener conocimientos de SQL, además provee un complemento para Eclipse que permite realizar consultas a la base de datos y posibilita mapear las clases con completamiento de código, además puede generar el código Java a partir del modelo de base de datos y viceversa.

1.4.2 Marcos de trabajos para aplicaciones web. Spring Framework

La plataforma J2EE cuenta con marcos de trabajo muy populares para desarrollar aplicaciones web, entre ellos se pueden mencionar Struts, Java Server Faces (JSF) y Spring Framework.

Struts es un marco de trabajo para aplicaciones J2EE que implementa el patrón de diseño Modelo Vista Controlador[33]. Es de código abierto y está liberado bajo la licencia Apache 2.0. Proporciona un conjunto de etiquetas JSP que cooperan con el controlador para ser usadas en la capa de visualización, en la que puede usarse tecnologías como JSP, FreeMarker y Velocity[35]. Puede integrarse con otros marcos de trabajo como Hibernate y Spring.

Por su parte JSF es un marco de trabajo estándar en Java para construir interfaces de usuarios para aplicaciones web. Simplifica el desarrollo de las interfaces que es a menudo una de las más difíciles y tediosas tareas del desarrollo en la web[36]. JSF provee un MVC donde las vistas conocen al controlador y en caso de un cambio en los mismos, se hace necesario modificar el código de las páginas de la capa de visualización. Por otro lado la construcción de las vistas es tediosa por la gran cantidad de líneas de código que hay que escribir.

Uno de los marcos de trabajo más populares en la comunidad de Java es Spring Framework, este incluye varias librerías que ayudan a solventar problemas comunes en el desarrollo de aplicaciones. Spring soporta la Programación Orientada a Aspecto, implementa el patrón MVC, separando la interfaz de usuario, los controladores y la capa de acceso a datos y utiliza el Objeto

de Acceso a Datos (DAO, siglas en inglés de Data Acces Object), que es un patrón de diseño para gestionar las bases de datos[37]. También permite el trabajo con herramientas ORM como Hibernate y con motores de plantillas para interfaces de usuarios como Tiles y Velocity, además se puede utilizar en conjunto con otros marcos de trabajo como JSF y Struts.

1.4.3 Marcos de trabajos JavaScript. JQuery y DWR

JavaScript es un lenguaje de programación que es interpretado por el navegador web, se utiliza fundamentalmente para crear interfaces de usuarios dinámicas, con JavaScript se pueden entre otras cosas, manejar los eventos del ratón, crear efectos visuales y manipular los diversos elementos que componen una página web. Actualmente se pueden encontrar muchos marcos de trabajo para JavaScript, entre los más populares están Moo Tools, Dojo Toolkit, YUI (siglas del inglés Yahoo User Interface) y JQuery, y otros de propósitos específicos como Direct Web Remoting (DWR por sus siglas en inglés).

Una librería con gran aceptación a nivel internacional es JQuery, es una biblioteca de JavaScript rápida y concisa que simplifica el desarrollo de documentos HTML, el manejo de eventos, crear animaciones e interactuar vía AJAX con el servidor para crear aplicaciones web rápidamente[38].

Teniendo en cuenta las ventajas que brinda la utilización de la técnica AJAX en las aplicaciones web, se consideró la utilización de Direct Web Remoting (DWR), un marco de trabajo de código abierto especializado en AJAX, perteneciente a Getahead (<http://getahead.ltd.uk/>). DWR brinda un enfoque interesante para AJAX, ya que permite el tratamiento de las clases de Java que se ejecutan en el servidor como si fueran locales[39], permitiendo invocar métodos del servidor desde el navegador e incrementando el rendimiento de la aplicación. DWR se puede integrar con Spring y otros marcos de trabajos para procesar peticiones asíncronas al servidor, está basado en los servlets de Java, aunque se explique aquí dentro de los marcos de trabajo JavaScript.

1.4.4 Marcos de trabajos para crear servicios web. Apache Axis 2

Un servicio web es una aplicación de software que puede ser remotamente accesible usando diferentes lenguajes basados en XML y está identificado por una URL (siglas del inglés Uniform Resources Locator)[40].

La gran ventaja de los servicios web es que independizan a las aplicaciones de forma tecnológica, ya que las funciones que realizan pueden ser llamadas desde cualquier lenguaje como PHP, Python o Perl. Existen varias bibliotecas para la implementación de servicios web como Metro, Apache CXF y Apache Axis 2.

De los marcos de trabajo antes mencionados, el más utilizado es Apache Axis2, este fue desarrollado por la Fundación Apache (<http://www.apache.org/>) con el objetivo de facilitar la implementación de servicios web, soporta SOAP 1.1 y 1.2, es eficiente, modular y extensible, esta propiedad permite soportar las especificaciones WS-^{*1}, que congrega la mayoría de las especificaciones para desarrollar servicios web, Axis2 es el motor de servicios web que soporta el mayor número de especificaciones WS-^{*}[41].

1.5 Sistemas gestores de base de datos

Un sistema gestor de bases de datos o RDBMS (siglas del inglés Relational DataBase Management System), es un programa capaz de almacenar una enorme cantidad de datos de gran variedad y de distribuirlos para cubrir las necesidades de una organización[42]. Existen RDBMS muy populares, entre los que sobresalen: MySQL, Oracle y PostgreSQL.

MySQL es un RDBMS que dispone de muchas de las funciones que exigen los desarrolladores. Se desarrolla y actualiza de forma rápida, se ejecuta en la inmensa mayoría de sistemas operativos y en la mayor parte de los casos, los datos se pueden transferir de un sistema a otro sin dificultad. Resulta fácil de utilizar y de administrar. Las herramientas de MySQL son potentes y flexibles, sin sacrificar su capacidad de uso. MySQL pertenece a la compañía Oracle Corporation (<http://www.oracle.com/index.html>).

Uno de los RDBMS más completos que existe en la actualidad es Oracle, desarrollado por Oracle Corporation. Oracle organiza eficientemente los datos de una manera relacional, es privativo por lo que hay que pagar para su uso, está basado en RAC (siglas del inglés Real Application Clusters), de modo que el procesamiento es sobre una red de servidores brindando un mayor rendimiento y aprovechamiento de los recursos[43]. Oracle puede ejecutarse en la mayoría de los sistemas operativos existentes como Windows, Unix, Linux y Solaris. Utiliza técnicas de compresión que mejora el tiempo de consulta y reduce el tráfico de datos por la red.

Otro RDBMS con mucha aceptación en la comunidad internacional es PostgreSQL, un sistema de manejo de base de datos relacional basado en Postgres, desarrollado en la Universidad de California de Ciencia de la Computación en Berkeley. PostgreSQL provee muchas características que están disponibles en herramientas comerciales como Oracle, MS SQL y no comerciales como MySQL.

1 Conjunto de especificaciones estándares recomendadas para desarrollar servicios web.

1.5.1 PostgreSQL

PostgreSQL es altamente extensible, se pueden añadir tipos de datos propios, operadores, indexar los tipos, reescribir partes del sistema y agregar tus propios complementos. El diseño de PostgreSQL permite a los hilos de ejecución leer y escribir con la seguridad de que no exista bloqueo en los procedimientos. PostgreSQL está implementado para que se ejecute en varios sistemas operativos. A diferencia de otros sistemas de bases de datos, PostgreSQL es desarrollado por una comunidad, está liberado bajo licencia TPL (siglas del inglés The PostgreSQL License), que es una licencia muy similar a la Berkeley Software Distribution (BSD) de código abierto[44]. El autor del presente documento considera que de los sistemas gestores de base de datos estudiados PostgreSQL es la opción ideal por las características mencionadas.

1.6 Entornos de desarrollo integrado

Un entorno de desarrollo integrado es una herramienta de software dedicada exclusivamente al desarrollo de programas informáticos, brindando una serie de complementos que facilitan el desarrollo ágil de software. Para el desarrollo de aplicaciones en el lenguaje de programación Java se utilizan fundamentalmente dos IDEs: Eclipse y NetBeans, sin descartar a IntelliJ, pero este por ser comercial y de elevado costo no posee un buen respaldo en la comunidad.

1.6.1 Eclipse

Eclipse Foundation (<http://www.eclipse.org/>) es una comunidad de código abierto con proyectos enfocados en proveer una plataforma de desarrollo de marcos de trabajo y herramientas para desarrollar y gestionar los ciclos de vida en el desarrollo de software, la plataforma de desarrollo es denominada Proyecto Eclipse, la última versión del IDE, en el momento que se escribe este documento, es la 3.7 llamada Eclipse Índigo. Eclipse no sólo trae editores de código para Java, se puede utilizar para desarrollar en otros lenguajes como PHP, C y C++. Posibilita la agregación de componentes ya que su arquitectura está basada en complementos. Se optó por usar Eclipse, ya que en la práctica resultó ser más económico en la utilización de los recursos de hardware y por lo tanto se muestra con mayor rendimiento y rapidez que la versión 7.1 de NetBeans.

1.7 Metodología de desarrollo OpenUP y lenguaje de modelado UML

Para desarrollar la solución se utilizó la metodología de desarrollo de software OpenUP. Esta preserva las características fundamentales del Proceso Unificado de Desarrollo (RUP por sus siglas en inglés). OpenUP es un proceso de desarrollo de software de código abierto diseñado para pequeños equipos organizados, quienes quieren tomar una aproximación ágil del desarrollo. Es un proceso iterativo que es mínimo, completo, y extensible. Se valora la colaboración y el

aporte de los stakeholders² sobre los entregables. OpenUP está organizado dentro de cuatro áreas principales de contenido: comunicación y colaboración, intención, solución y administración. Se caracteriza por ser iterativo e incremental, estar centrado en la arquitectura y guiado por los casos de uso.

Además utiliza el Lenguaje Unificado de Modelado (UML por sus siglas en inglés) para el modelado de sistemas, este consiste en un grupo de especificaciones orientadas a objetos, compuestas por distintos diagramas, que representan las etapas del desarrollo de un proyecto de software. Permite la especificación, visualización, construcción y documentación de elementos de la Ingeniería del Software. UML se define como un lenguaje que permite especificar, visualizar y construir los artefactos de sistemas de software. Es un sistema notacional (que, entre otras cosas, incluye el significado de sus notaciones) destinado a los sistemas de modelado que utilizan conceptos orientados a objetos[45].

1.8 Herramientas CASE. Visual Paradigm

Las herramientas CASE (del inglés Computer Aided Software Engineering, que se traduce al español como: Ingeniería de Software Asistida por Computadora), son herramientas con el objetivo de incrementar la productividad y la calidad de los productos de software, mejorar la planificación del proyecto así como reducir el tiempo y coste de su desarrollo. Existen numerosas herramientas CASE, algunas de ellas son: Umbrello, Rational Rose y Visual Paradigm.

Visual Paradigm for UML, está diseñada para desarrollar software con Programación Orientada a Objetos, busca reducir la duración del ciclo de desarrollo brindando ayuda tanto a arquitectos, analistas, diseñadores y desarrolladores. Es una herramienta de modelado visual para todos los tipos de diagramas UML compatible con una amplia gestión de casos de uso que permite el diseño de bases de datos y reúne los esfuerzos más eficaces en el análisis y diseño de sistemas.

1.9 Conclusiones del capítulo

En el presente capítulo se describieron los principales conceptos relacionados con la evolución de Internet y el desarrollo de aplicaciones web, se analizaron las tendencias actuales en la gestión y análisis de datos biológicos en la web, centrando la atención en las aplicaciones disponibles para el análisis de secuencias. Entre los principales portales que brindan ese tipo de servicios, se encuentran: NCBI, EMBL y Expasy.

² Personas afectadas por el desarrollo del software, ya sea positiva o negativamente

Se determinó el uso de servicios web bajo las directrices de una Arquitectura Orientada a Servicios la que se desarrollará bajo el lenguaje de programación Java, para así aprovechar las características de la plataforma J2EE a través de componentes como Servlets, JSP, JSTL y Portlets. Por otra parte se fundamentó la selección de Liferay Portal como portal de aplicaciones y contenedor de Portlet, Eclipse como entorno de desarrollo integrado, Apache Axis 2 para la implementación de los servicios web y como marco de trabajo se optó por Spring Framework.

Además se estableció PostgreSQL como RDBMS, el cual se encargará de almacenar los datos persistentes e Hibernate como herramienta ORM. Para lograr una mayor interactividad en la aplicación se eligió el uso de AJAX, auxiliándose de DWR y JQuery. Finalmente el proceso de desarrollo estará guiado por la metodología de desarrollo de software OpenUP, la cual utiliza UML como lenguaje de modelado y se eligió como herramienta CASE al Visual Paradigm.

Capítulo 2. Requisitos

En este capítulo se describen las características fundamentales del módulo básico y se identifican los resultados esperados del sistema. Para ello se realiza el levantamiento de requisitos funcionales y no funcionales a partir del modelo conceptual construido, basándose en los objetos relevantes del problema, además se identifican los casos de uso que contiene el sistema y se describe cada uno para guiar el proceso de desarrollo de software.

2.1 Modelo conceptual

Un paso esencial en el análisis de construcción de software orientados a objetos es descomponer el problema en conceptos u objetos individuales en un modelo conceptual. El modelo conceptual del módulo básico se muestra en la Figura 5.

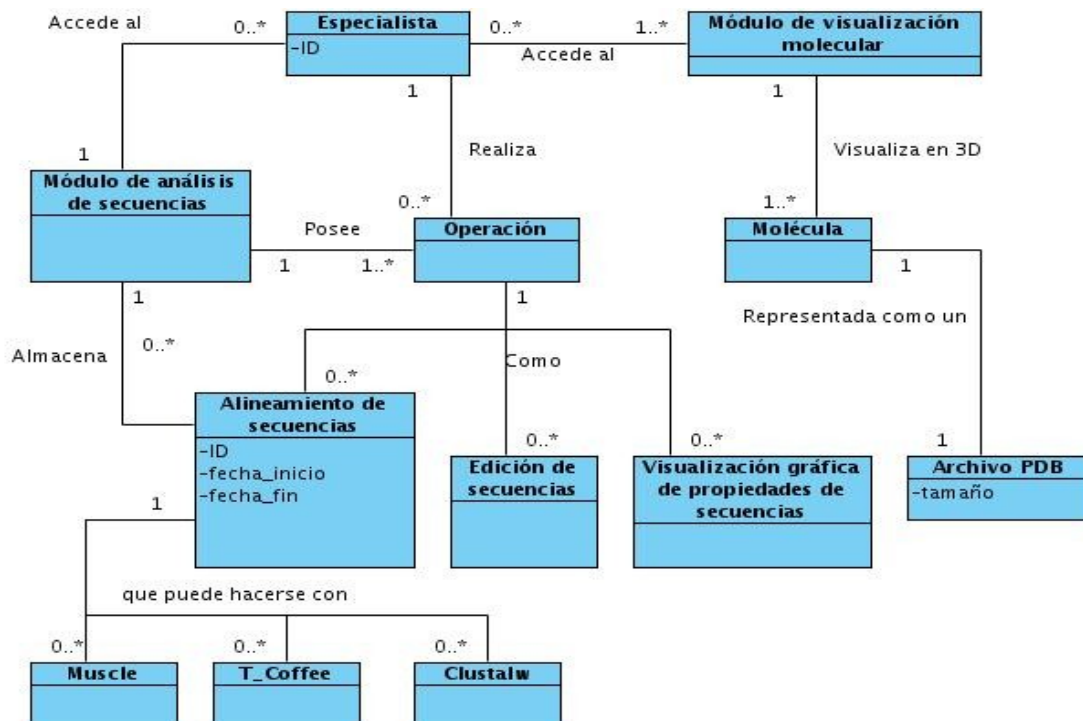


Figura 5: Modelo conceptual del sistema. Representa los conceptos del dominio del mundo real y las relaciones entre ellos. El modelo de dominio no representa objetos de software.

Un modelo conceptual es una representación de conceptos en un dominio del problema. En UML se ilustra con un grupo de diagramas de estructura estática donde no se define ninguna operación. La designación del modelo conceptual ofrece la ventaja de subrayar fuertemente una concentración en los conceptos del dominio, no en las entidades del software. Los conceptos del dominio se identificaron siguiendo las estrategias de obtención de conceptos a partir de una lista de categorías y a través de frases nominales. Los conceptos del dominio del módulo básico se describen a continuación.

Tabla 1: Descripción de los conceptos del dominio del módulo básico.

Concepto del dominio	Descripción
Especialista	Persona con conocimientos básicos en el área de la Bioinformática, capacitada para interactuar con el sistema.
Módulo de visualización molecular	Subsistema del módulo básico que permite al especialista representar moléculas en el espacio tridimensional.
Molécula	Unidad mínima de una sustancia que conserva sus propiedades químicas. Puede estar formada por átomos iguales o diferentes.
Archivo PDB	Formato de archivo utilizado para representar moléculas.
Módulo de análisis de secuencias	Subsistema del módulo básico que permite realizar operaciones para analizar secuencias moleculares
Operación	Acción que ejecuta el especialista a través del sistema.
Alineamiento de secuencias	Operación de alinear una o varias secuencias.
Edición de secuencias	Operación que permite la edición de secuencias como: cortar subsecuencias, complementar secuencias, etc.
Visualización gráfica de propiedades de secuencias	Operación que muestra a través de imágenes diferentes características de las secuencias.
Muscle	Programa o algoritmo que permite alinear secuencias.
T_Coffee	Programa o algoritmo que permite alinear secuencias.
Clustalw	Programa o algoritmo que permite alinear secuencias.

2.2 Requisitos del sistema

La base de comunicación entre los clientes y el equipo de desarrollo reside en los requisitos de software, estos están divididos en dos tipos: los funcionales y los no funcionales.

2.2.1 Requisitos funcionales

Los requisitos funcionales (RF) son capacidades o condiciones que el sistema debe cumplir. Expresan una especificación detallada de las responsabilidades del sistema en cuestión y permiten determinar, de una manera clara, lo que debe hacer el sistema.

El objetivo fundamental de los RF es identificar y documentar lo que se espera del sistema, de una forma clara y comprensible para el cliente y los miembros del equipo de desarrollo. El reto consiste en definirlos de manera inequívoca, de modo que se detecten los riesgos y no se presenten sorpresas al momento de entregar el producto[45]. Para el módulo básico se determinaron 21 RF los cuales se definen a continuación:

RF1: Administrar alineamiento

RF1.1: Consultar resultado del alineamiento

RF1.2: Actualizar alineamiento

RF1.3: Buscar alineamiento

RF1.4: Eliminar alineamiento

RF1.5: Listar alineamientos

RF2: Realizar alineamiento de secuencias

RF3: Visualizar moléculas en tres dimensiones

RF4: Visualizar marcos abiertos de lectura de secuencias nucleotídicas

RF5: Visualizar propiedades químico-físicas de secuencias proteicas

RF6: Visualizar segmentos de transmembrana de secuencias proteicas

RF7: Generar secuencias aleatoriamente

RF8: Mostrar características básicas de una secuencia

RF9: Editar secuencias

RF9.1: Invertir secuencia

RF9.2: Complementar secuencia

RF9.3: Unir secuencias

RF9.4: Ordenar secuencias por tamaño

RF9.5: Extraer subsecuencia

RF10: Convertir formato de secuencia

RF11: Traducir secuencias de ácidos nucleicos

2.2.2 Requisitos no funcionales

Los requisitos no funcionales (RNF) son propiedades o cualidades que el producto debe tener, aunque no formen parte de su función. Existen múltiples categorías para clasificar los requisitos no funcionales, siendo las siguientes las más representativas, aunque no limitan la definición de otras. Para el módulo básico se definieron los siguientes RNF:

Software

- En los ordenadores de los usuarios debe estar instalado:
 - La Máquina Virtual de Java (JVM siglas en inglés de Java Virtual Machine).
 - AL menos un navegador web con los complementos para visualizar applets³ de Java.
- Para el servidor de servicios web se deben instalar los siguientes programas:
 - Clustalw 2.0.12: para realizar alineamientos de secuencias con Clustalw.
 - Muscle 3.7: para realizar alineamientos de secuencias con Muscle.
 - T_Coffee 8.84: para realizar alineamientos de secuencias con T-Coffee.
 - Jemboss 6.1.0-5: una suite de herramientas para procesar datos biológicos.
 - Readseq 1.0: que posibilita la conversión de formatos de secuencias moleculares.

Restricciones en el diseño y la implementación

- Se utilizará el estándar de codificación propuesto por el lenguaje de programación Java.

³ Componentes implementados en Java que pueden mostrarse a través de una página HTML.

- Para las llamadas asíncronas al servidor se utilizará la tecnología AJAX con asistencia de la librería DWR.
- El acceso a la base de datos se garantizará a través del patrón Objeto de Acceso a Datos.
- Para la programación en el lado del cliente se utilizará JavaScript con la librería JQuery 1.4.4, de igual forma se empleará JQueryUI para enriquecer la aplicación.

Usabilidad

- El sistema podrá ser utilizado por usuarios con conocimientos básicos en el área de la Bioinformática.

2.3 Casos de uso del sistema

Para entender los requisitos se necesita, en parte, conocer los procesos del dominio y el ambiente externo, o sea los factores externos que participan en los procesos. Dichos procesos de dominio pueden expresarse en casos de uso (CU), que son descripciones narrativas de los procesos del dominio en un formato estructurado de prosa[45].

Los CU describen las interacciones entre uno o más usuarios y el sistema, con el fin de proporcionar un resultado observable de valor para el actor[46]. El modelo de CU del módulo básico se puede apreciar en la Figura 6.

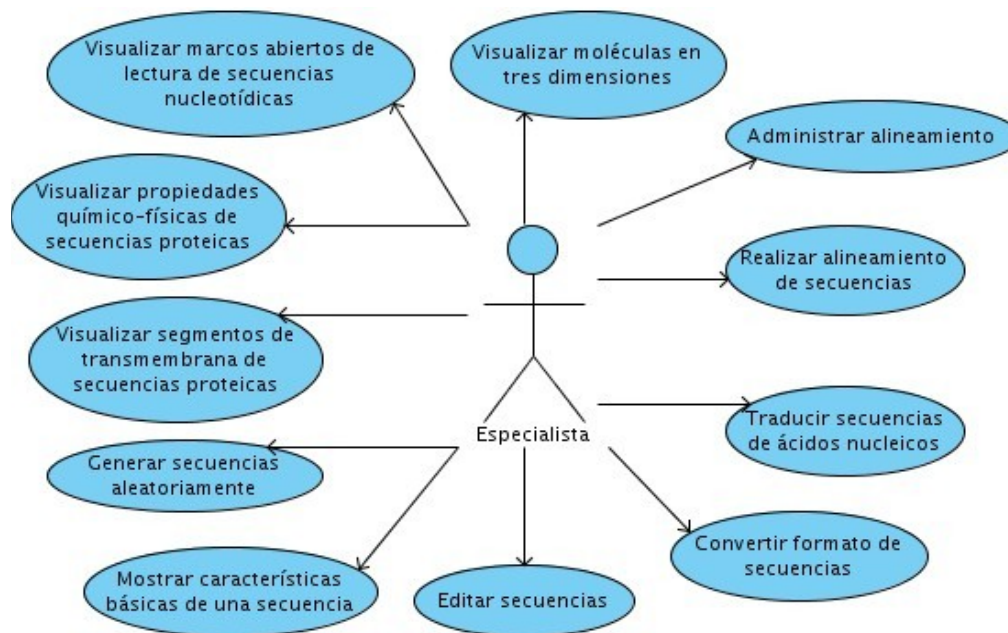


Figura 6: Diagrama de casos de uso del sistema. Los CU pueden agrupar más de un requisito funcional, todos los CU del módulo básico serán inicializados por el especialista en Bioinformática.

2.4 Descripción de los casos de uso

La tarea de especificar los casos de uso busca adquirir un detalle suficiente de estos para entender las necesidades del cliente y proveer una comprensión que permita comenzar el desarrollo del sistema. En este apartado se describirá el CU “Realizar alineamiento de secuencias” por ser el que más impacta en la arquitectura del sistema, para ver las descripciones de los restantes CU se debe consultar el Anexo E.

Tabla 2: Descripción del CU Realizar alineamiento de secuencias.

Actores:	Especialista
Propósito:	Alinear secuencias
Resumen:	El CU se inicia cuando el especialista selecciona la opción “Alineamientos de secuencias” del módulo de análisis de secuencia.
Referencia:	RF2
Precondiciones:	El especialista debe estar autenticado.
Postcondiciones:	El sistema alinea las secuencias con el programa seleccionado y muestra el resultado de la operación.
Flujo Normal de Eventos	
Acción del Actor	Respuesta del sistema
1 El especialista selecciona la opción “Realizar alineamiento de secuencias” del módulo “Análisis de secuencia”.	1.1 El sistema muestra la página con las opciones de alinear por Clustalw, Muscle y T-Coffee.
2 El especialista escoge una opción.	2.1 Si escoge: - Alinear con Clustalw, ver sección Alinear con Clustalw. - Alinear con Muscle, ver sección Alinear con Muscle. - Alinear con T-Coffee, ver sección Alinear con T-Coffee.
Sección Alinear con Clustalw	
	1.1 El sistema muestra el formulario para que el usuario entre los datos.
2 El especialista entra los datos: - sequence, type, quicktree, ktuple, topdiags, pairgap, score, pwgapopen, pwgapext, pwmatrix, pwdnamatrix, helixgap, strandgap, loopgap, terminalgap, helixendin, helixendout, strandendin, strandendout, secstrout, output,	2.1 El sistema valida los datos y guarda en la base de datos el alineamiento creado. 2.2 El sistema llama al servicio web “Alinear con Clustalw”. 2.3 El sistema obtiene el resultado y actualiza el alineamiento en la base de datos. 2.4 El sistema muestra una página con el resultado del alineamiento y la opción “Listar alineamientos”.

outorder, nosectr1, nosectr2, seqnos.	
Sección Alinear con Muscle	
	1.1 El sistema muestra el formulario para que el usuario entre los datos.
2 El especialista entra los datos: - sequence, outfile, maxiters, diags, refine, usetree_nowarn, profile, matrix, matrixgapopen, matrixgapextend, matrixcenter, urlFile.	2.1 El sistema valida los datos y guarda en la base de datos el alineamiento creado. 2.2 El sistema llama al servicio web "Alinear con Muscle". 2.3 El sistema obtiene el resultado y actualiza el alineamiento en la base de datos. 2.4 El sistema muestra una página con el resultado del alineamiento y la opción "Listar alineamientos".
Sección Alinear con T-Coffee	
	1.1 El sistema muestra el formulario para que el usuario entre los datos.
2 El especialista entra los datos: - sequence, outfile, special_mode, in, profile, profile1, profile2.	2.1 El sistema valida los datos y guarda en la base de datos el alineamiento creado. 2.2 El sistema llama al servicio web "Alinear con T-Coffee". 2.3 El sistema obtiene el resultado y actualiza el alineamiento en la base de datos. 2.4 El sistema muestra una página con el resultado del alineamiento y la opción "Listar alineamientos".
Flujo alterno	
1 El especialista entra datos inválidos.	1.1 El sistema muestra los mensajes de error correspondientes, especificando los campos incorrectos. 1.2 El sistema retorna al paso 2 de la sección Alinear con Clustalw.
Prioridad	Crítico

2.5 Conclusiones del capítulo

En este capítulo se definieron los artefactos correspondientes a los requisitos de la metodología de desarrollo OpenUP, y aprovechando su extensibilidad, se definió el modelo conceptual del sistema y se describieron los objetos del dominio del problema. Se definieron 21 requisitos funcionales y 12 requisitos no funcionales. Los RF identificados se agruparon en el modelo de casos de uso del sistema, destacando por su impacto en la arquitectura el CU: "Realizar alineamiento de secuencias", el cual se describe dividido en secciones con el objetivo de lograr una mayor comprensión del mismo.

Capítulo 3. Diseño

Luego de haber comprendido los conceptos y los requisitos del sistema en el capítulo anterior, se procede a diseñar la solución propuesta. Este capítulo brinda un acercamiento a la implementación del sistema, para ello, se construyen los diagramas de interacciones del mismo, así como el modelo de clases del diseño, los cuales proporcionan la base para proceder a la etapa de construcción.

3.1 Arquitectura de software

Un problema frecuente en los proyectos, radica en no comprender la utilidad de preparar los diagramas de interacción, la consideración cuidadosa de la asignación de responsabilidades y las habilidades que todo esto requiere[45].

En todo desarrollo de software, tanto la calidad de diseño de la interacción de los objetos como la asignación de responsabilidades presentan gran variación. Las decisiones poco acertadas en este sentido dan origen a sistemas y componentes frágiles y difíciles de mantener. Una alternativa hábil para disminuir este error se funda en los principios que rigen una buena arquitectura de software.

La arquitectura de software define la estructura de un sistema. Esta estructura se constituye de componentes, módulos o piezas de código que nacen de la noción de abstracción, cumpliendo funciones específicas e interactuando entre sí con un comportamiento definido. Los componentes se organizan de acuerdo a ciertos criterios, que representan decisiones de diseño. En los últimos años la arquitectura de software se ha debatido en dos corrientes fundamentales: los estilos arquitectónicos y los patrones arquitectónicos.

3.1.1 Estilos y patrones arquitectónicos

Un estilo arquitectónico es como una familia de sistemas de software en términos de un patrón de organización estructural, que define un vocabulario de componentes y tipos de conectores y un

conjunto de restricciones de cómo pueden ser combinadas[47]. Para muchos estilos puede existir uno o más modelos que especifiquen cómo determinar las propiedades generales del sistema.

Un patrón arquitectónico es la expresión de un esquema estructural de organización para sistemas de software, que proporciona un conjunto de subsistemas predefinidos, especifica sus responsabilidades e incluye normas y directrices para las relaciones entre ellos[48].

Existen disímiles categorías para los estilos arquitectónicos, que incluyen a su vez varios estilos, como por ejemplo: Llamada y retorno, que aglutina la arquitectura orientada a objetos, arquitecturas basadas en componentes y arquitectura en capas y la categoría Peer to peer que contiene estilos de arquitectura basada en eventos, arquitecturas orientada a servicios y arquitectura basada en recursos. Existen otros estilos como Tuberías y filtros, Código móvil y Arquitectura centrada en datos.

Los términos estilo y patrón arquitectónicos son a menudo confundidos, ya que están ampliamente diseminados. En el libro titulado “*Arquitectura de software en práctica*” de Len Bass, Paul Clements y Rick Kazman describen a un patrón arquitectónico como un conjunto de restricciones en una arquitectura que describe sus elementos y los tipos de relaciones entre ellos, mas adelante especifican que el término estilo arquitectónico ha sido usado también para describir el mismo concepto[49]. Aunque no se precise de una definición estándar para estos términos, sean estilos o patrones arquitectónicos, sus aplicaciones en el desarrollo de software constituyen un elemento elemental.

El módulo básico presenta los patrones arquitectónicos Cliente-Servidor y Arquitectura Orientada a Servicios. El primero se manifiesta en el paradigma petición-respuesta, dado que es una aplicación web a la que se accede por el protocolo HTTP. El segundo está representada en los servicios web implementados que están disponibles a través del protocolo SOAP e intercambian información usando XML.

3.1.2 Patrones de diseño

Al momento de elaborar los diagramas de interacción es importante asignar correctamente las responsabilidades. Para ello, una buena práctica es auxiliarse de los patrones de diseño, entre los que se pueden mencionar los GRASP, más conocidos como patrones de asignación de responsabilidades, los patrones GoF⁴ y el Modelo Vista Controlador (MVC).

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

Los patrones GRASP describen los principios fundamentales de la asignación de responsabilidades a objetos, expresados en forma de patrones[45]. En la siguiente tabla se resumen algunos de los más utilizados.

Tabla 3: Los patrones GRASP y sus funciones.

Patrón GRASP	Función
Experto	Asignar una responsabilidad al experto en información: la clase que cuenta con la información necesaria para cumplir la responsabilidad.
Creador	Explica que clase es la encargada de crear objetos, en determinados escenarios de ejecución.
Bajo acoplamiento	Asignar una responsabilidad para mantener bajo acoplamiento. Explica cómo dar soporte a una dependencia escasa y a un aumento de la reutilización.
Alta Cohesión	Asignar una responsabilidad de modo que la cohesión siga siendo alta. Explica cómo mantener la complejidad dentro de límites manejables.
Controlador	Asignar la responsabilidad del manejo de los eventos de un sistema a una clase que represente un sistema global.

Los patrones GoF complementan a los patrones GRASP y en ocasiones se puede encontrar una contraposición entre este tipo de patrones, e incluso, podría inferirse que algunos patrones GoF son variantes de los patrones GRASP, es por ello que la decisión de utilizar uno u otro debe tomarse con precaución y aplicarse sólo en el ámbito necesario. Entre los patrones GoF más utilizados se pueden citar el de Agente y Agente Remoto, Fachada y Agente Dispositivo y el Comando.

El patrón Agente explica el comportamiento cuando no se desea o no es posible acceder directamente a un componente, para ello este patrón sugiere definir una clase sustituta de software que represente al componente y asignarle la responsabilidad de comunicarse con el componente real. Por su parte el patrón Agente Remoto es un caso especial del patrón Agente[45]. Este define como el sistema debe comunicarse con un componente situado en otro espacio o contexto a través de una clase de software local que represente al componente externo y asignarle la responsabilidad de contactar al componente real.

El patrón Fachada es generalmente utilizado para resolver problemas de integración, y se aplica cuando se requiere una interfaz común de comunicación con un conjunto de interfaces o funciones

de otro subsistema, en este sentido Fachada define una sola clase que unifique las interfaces y le asigna la responsabilidad de colaborar con el subsistema. El patrón Agente Dispositivo es una especificidad de Fachada cuando el subsistema que se quiere integrar es un dispositivo externo.

Otro de los patrones GoF muy utilizado es el Comando, este patrón soluciona el problema de cuando un objeto o sistema recibe varias peticiones o comandos, para ello cada comando define una clase que lo represente y le asigna la responsabilidad de ejecutarse el mismo, de esta forma reduce la responsabilidad del receptor en el manejo de los comandos, aumenta la facilidad con que pueden agregarse otros comandos y ofrece las bases para registrar los comandos, formar colas de espera con ellos y cancelarlos.

Uno de los patrones de diseño generalizado en las aplicaciones actuales es el MVC, aunque en muchos textos este patrón se define como un patrón arquitectónico. El MVC separa el sistema en tres capas fundamentales, una capa de modelo, que representa los datos relacionados con la aplicación, una capa de vista o presentación, encargada de representar los datos del modelo al usuario, mediante la cual se permitirá manipular la información y una capa controladora que procesa las peticiones que vienen desde la capa de presentación y donde está encapsulada la lógica del negocio (Figura 7).

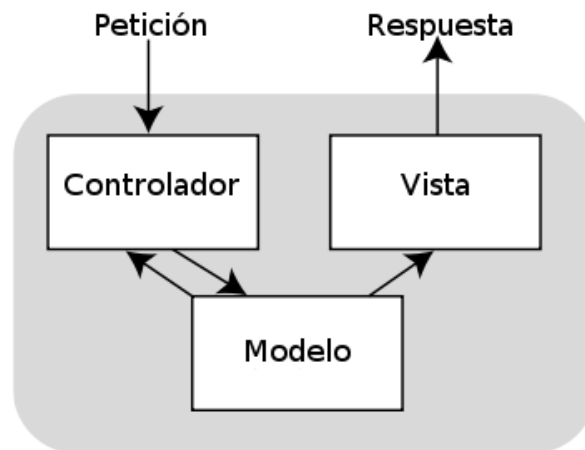


Figura 7: El patrón Modelo Vista Controlador. La utilización del patrón MVC aumenta el desacoplamiento de los componentes del sistema en el que cada capa responde a la llamada de otra manteniendo la cohesión entre ellas.

3.2 Diagramas de interacción

Para lograr una solución eficaz de los requisitos funcionales identificados es importante la elaboración de diagramas de interacción, que muestran gráficamente cómo los objetos se comunican entre sí a fin de cumplir con las funcionalidades.

El lenguaje de modelado UML define dos tipos de diagramas de interacción, ambos sirven para expresar interacciones semejantes o idénticas de mensaje, ellos son: los diagramas de secuencia y los diagramas de colaboración. Los diagramas de secuencia describen las interacciones en una especie de formato de cerca o muro, y se representan en el tiempo mediante líneas temporales en sentido vertical, mientras que los de colaboración describen las interacciones entre los objetos en un formato de grafo o red.

Los diagramas de interacción constituyen uno de los artefactos más importantes que se generan en el análisis y en el diseño orientados a objetos[45]. Es a partir de este planteamiento que se definen en las siguientes secciones este tipo de diagramas.

3.2.1 Diagramas de secuencia y colaboración

Los diagramas de secuencia muestran gráficamente los eventos que fluyen desde los actores hacia el sistema. Es conveniente crear un diagrama de secuencia por cada operación del sistema, y enfatizar la secuencia de los eventos en el flujo normal de eventos de los CU. En el presente acápite se tratará el diagrama de secuencia del CU: *“Realizar alineamiento de secuencias”*.

El diagrama de secuencia de la Figura 8 muestra la comunicación de los componentes del software. Se define una Clase Controladora (CC) llamada `ControladorFrontal.java` encargada de gestionar todas las peticiones del usuario y de determinar el controlador que le dará respuesta a dicha petición, en este caso `ControladorClustalw.java`. De esta manera se aplican patrones como el Controlador de GRASP y el Comando de los GoF, también están los patrones Bajo acoplamiento y Alta cohesión.

Por otro lado se identifican una serie de métodos o funciones en las clases controladoras, las cuales deben ser capaz, entre otras cosas, de mostrar formularios y validar los datos que provienen de los mismos.

En la Figura 9 se muestra el diagrama de colaboración del CU *“Realizar alineamiento de secuencias”* para la sección *“Alinear con Clustalw”*.

Los diagramas de colaboración y secuencia pueden sincronizarse dado que representan conceptos similares, aunque los de colaboración poseen mayor poder comunicativo en un menor espacio.

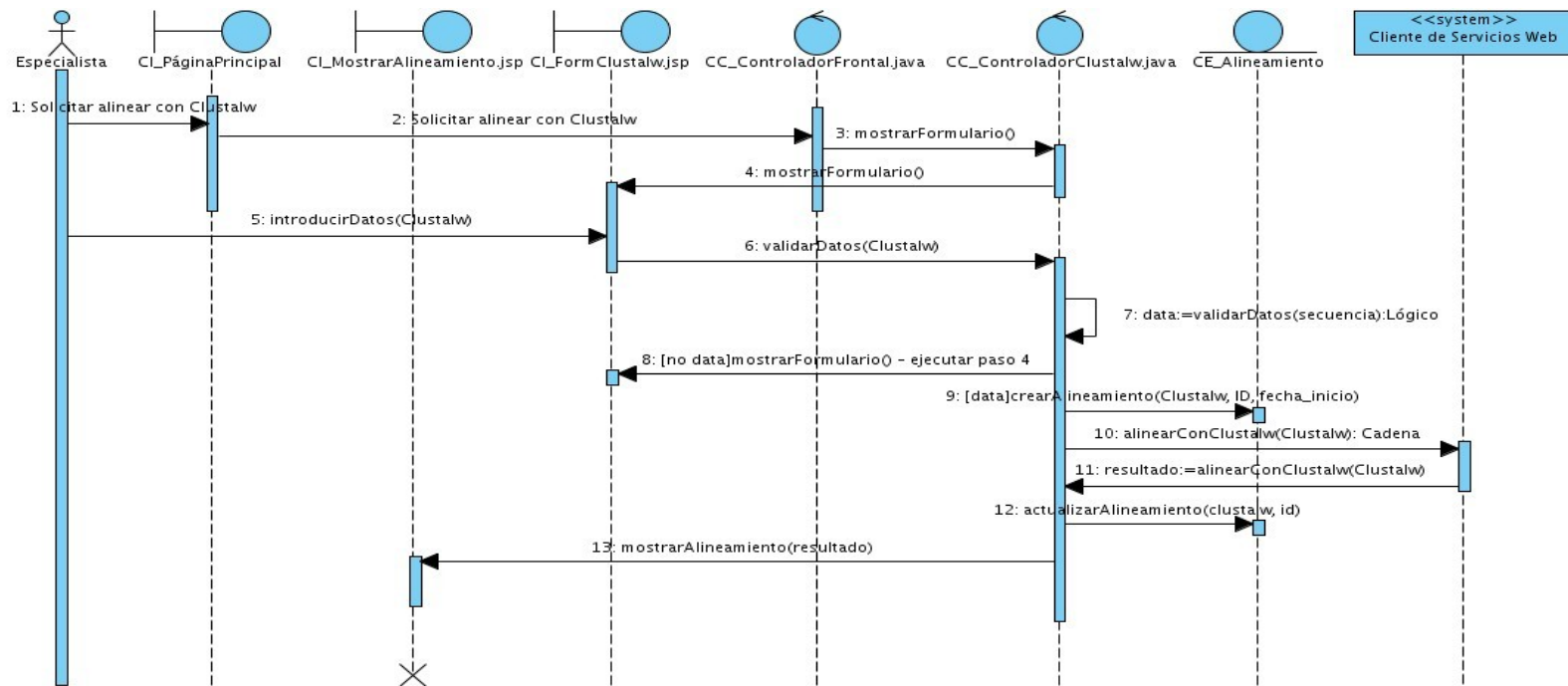


Figura 8: Diagrama de secuencia del CU Realizar alineamiento de secuencias. Las clases interfaces (CI) realizan peticiones a las clases controladores (CC), estas consultan las entidades(CE) de la base de datos y realizan las llamadas a los servicios web.

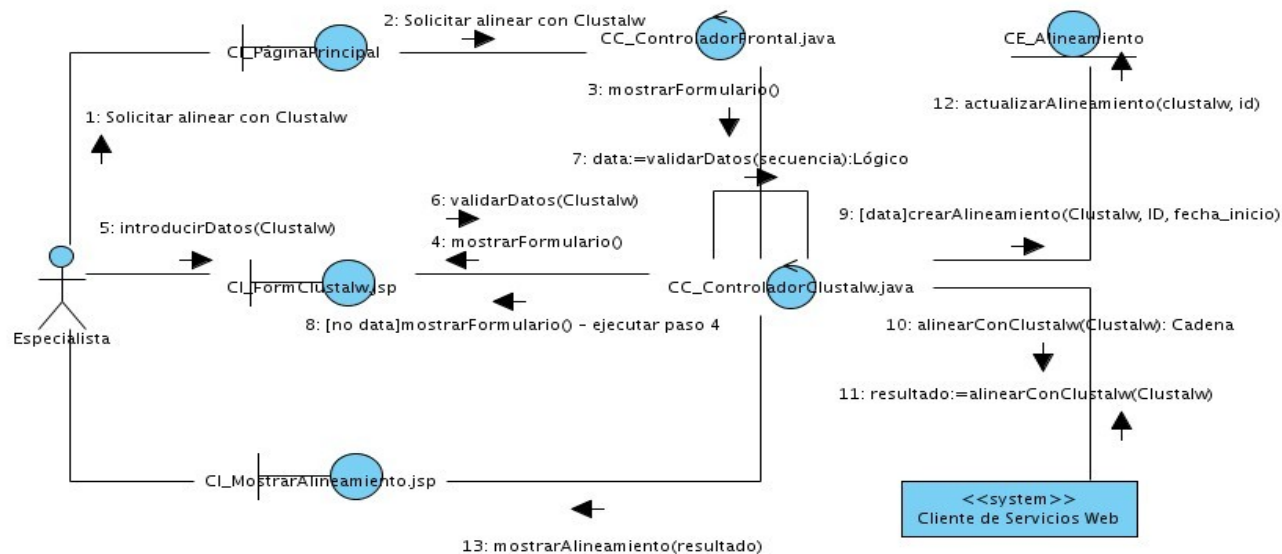


Figura 9: Diagrama de colaboración del CU Realizar alineamiento de secuencias. El mensaje entre los objetos es una trayectoria de conexión que indica una forma de navegación entre ambos objetos. Los mensajes incluyen parámetros que indican la transferencia de información.

3.3 Modelo del diseño

Una vez elaborados los diagramas de interacción, se pueden identificar las clases del software que participarán en la solución del sistema y complementarlas con el diseño. Un diagrama de clases del diseño, a diferencia del modelo conceptual, contiene las definiciones de las entidades del software en vez de conceptos del mundo real.

UML no define un elemento denominado "diagrama de clases del diseño", sino que se sirve de un término más genérico: "diagrama de clases"[45].

En el diagrama de clases del diseño para el CU "Realizar alineamiento de secuencias" (Figura 10), se identifican cuatro paquetes funcionales: Presentación, Lógica de aplicación, Objeto de Acceso a Datos, los que corresponden a las capas del patrón de diseño MVC, y por último el Cliente de Servicios Web. Además se puede apreciar la distribución de los objetos dentro de estos paquetes, separando el sistema en varias capas, aplicando el patrón arquitectura en capas.

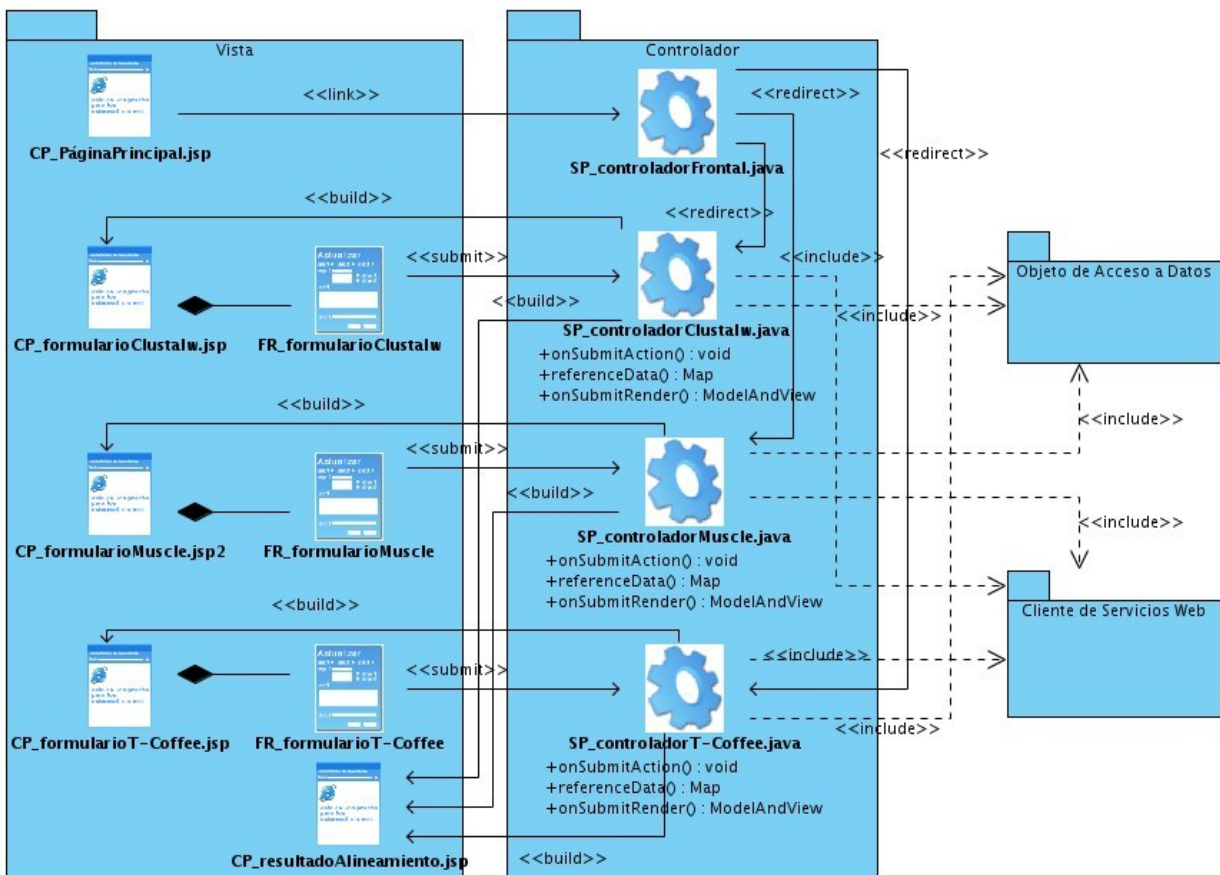


Figura 10: Diagrama de clases del diseño para el CU Realizar alineamiento de secuencias. Se muestran las páginas clientes (CP) en el paquete Vista, las páginas servidoras (SP) en el paquete Controlador y los paquetes encargados de gestionar los servicios web y la base de datos.

En el diagrama de clases del diseño correspondiente al paquete Objeto de Acceso a Datos (Figura 11) se define la clase `DAO.java` que es una interfaz donde se encuentran todos los métodos necesarios para gestionar la base de datos, la clase `DAOImplementado.java` implementa estos métodos y utiliza dos clases para acceder a las tablas del modelo: `UsuarioHome.java` y `AlineamientoHome.java`.

La clase `DAOImplementado.java` representa el acceso a la base de datos para los demás componentes del sistema, jugando el rol de fachada, correspondiente con el patrón de mismo nombre. Esta clase también es la experta en el acceso al modelo, pero para evitar el acoplamiento o la sobrecarga de información, delega las funciones de acceder a las tablas a las clases `UsuarioHome.java` y `AlineamientoHome.java`. De igual forma actúa como un agente ya que la base de datos solo estará disponible a través de él.

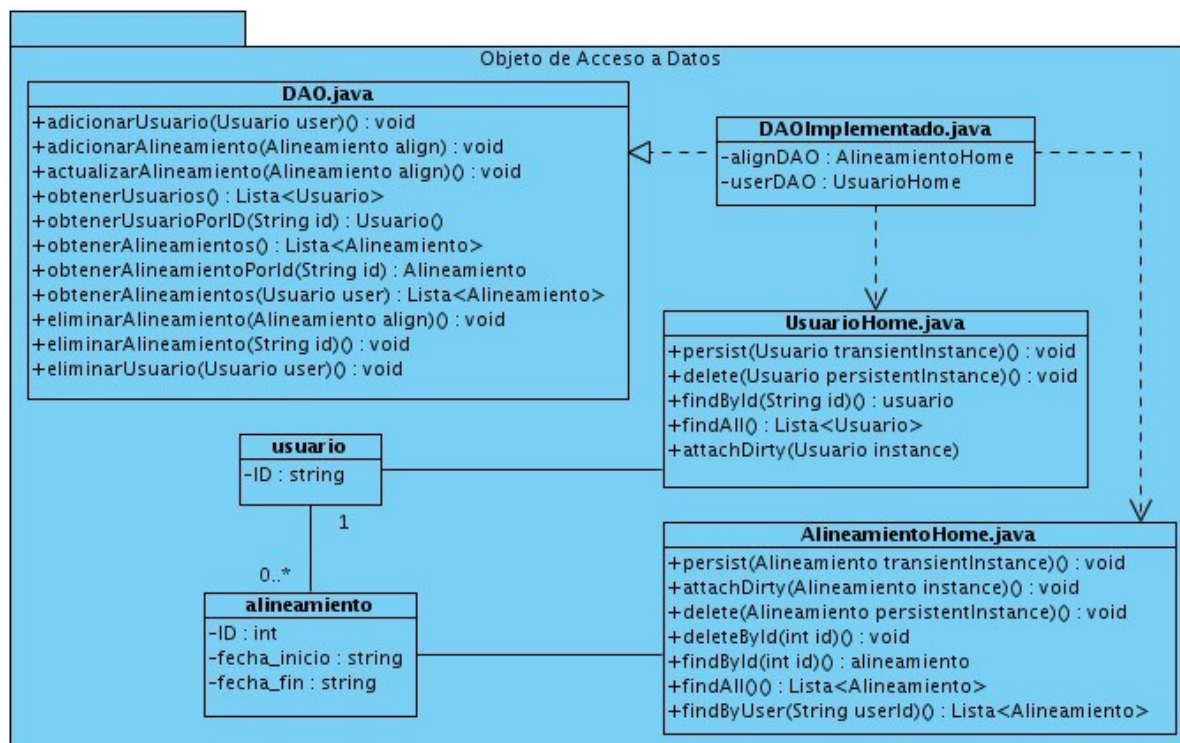


Figura 11: Diagrama de clases del diseño para el paquete Objeto de Acceso a Datos. Muestra los componentes encargados de gestionar la base de datos. La clase `DAOImplementado.java` es el punto de acceso a la base de datos para los demás componentes del sistema.

El diagrama de clase del diseño correspondiente al paquete Cliente de Servicios Web (Figura 12) muestra los componentes encargados de realizar las llamadas a las funciones remotas del sistema.

En este paquete la interfaz `BiosoftWSPortType.java` declara las funciones que invocan los servicios mientras que la clase `BiosoftWSPortTypeProxy.java` define las mismas, para ello utiliza otras clases, las que están en correspondencia con los parámetros necesitados por el servicio en cuestión, en este caso, el servicio web “*alinealClustalw*” requiere un objeto de la clase `Clustalw.java`. El subsistema Servicios Web Biosoft representa los servicios albergados en el servidor de servicios web. En este diagrama se manifiesta el patrón Agente Remoto de la familia GoF a través de la clase `BiosoftWSPortTypeProxy.java`.

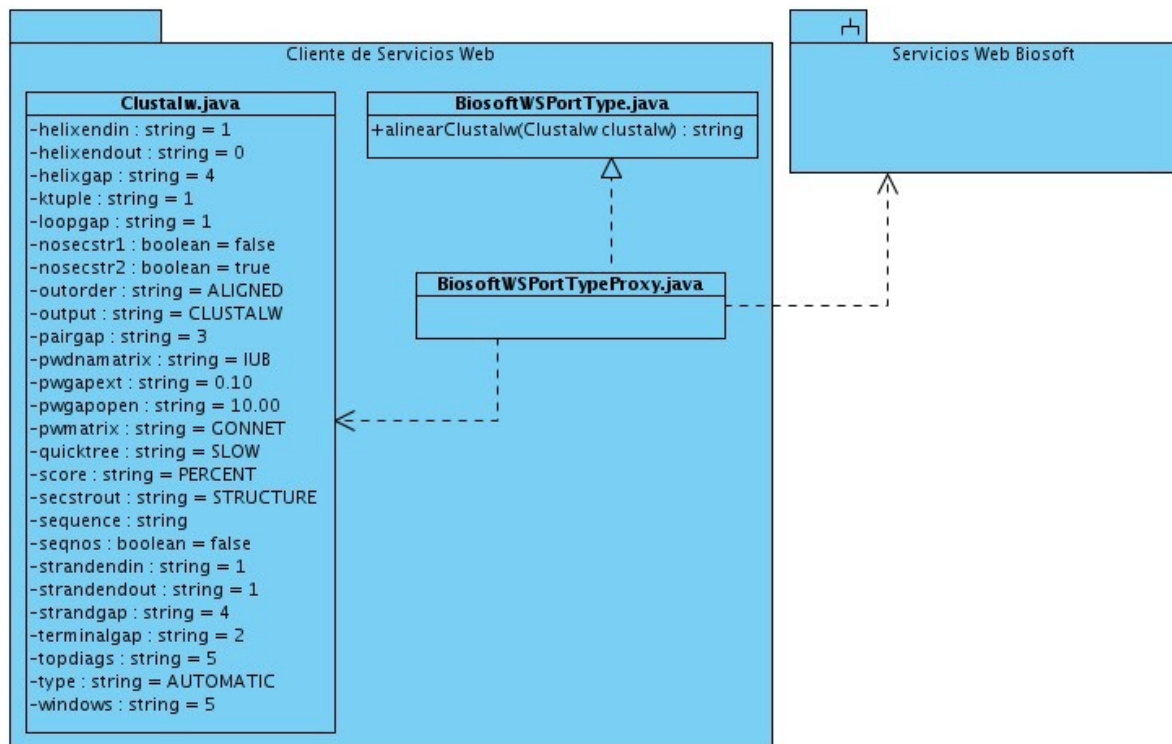


Figura 12: Diagrama de clases del diseño para el paquete Cliente de Servicios Web. Este paquete contiene las clases necesarias para invocar los servicios web que ejecutan las operaciones del sistema.

3.4 Conclusiones del capítulo

En el presente capítulo se definieron los diagramas de interacción del sistema para el CU de mayor impacto en la arquitectura: “*Realizar alineamiento de secuencias*”, específicamente para la sección “*Alinear con Clustalw*”. Para la elaboración de los mismos se analizaron diferentes patrones arquitectónicos y patrones de diseño que posibilitaron definir una mejor arquitectura, entre estos se definieron la arquitectura cliente-servidor y orientada a servicios, así como el MVC, Alta cohesión y Bajo acoplamiento.

Por último se definió el diagrama de clases del diseño para el CU: *“Realizar alineamiento de secuencias”*, en la elaboración del mismo se dividió el sistema en paquetes funcionales correspondientes con la arquitectura en capas, obteniendo una capa para la presentación (Vista), una para la lógica de aplicación (Controladora) y dos capas al mismo nivel: servicios web y modelo de base de datos. Estas dos últimas se representaron como paquetes individuales, los cuales fueron igualmente definidos, mostrando los objetos de software encargados de gestionar el acceso a la base de datos y de invocar las funciones implementadas como servicios web.

Capítulo 4. Implementación y pruebas

Una vez concluido el modelo del diseño, se dispone de los detalles suficientes para proceder a la construcción del sistema, y una vez concluido este, se procede a la verificación del cumplimiento de los requisitos funcionales mediante las pruebas de software. En el presente capítulo se construye el modelo de implementación correspondiente al módulo básico, desglosándolo en los diagramas de componentes de los requisitos más importantes, de la misma forma se determinan los tipos de pruebas a realizar y los casos de pruebas que serán aplicados al sistema.

4.1 Modelo de implementación

El modelo de despliegue se crea durante las últimas actividades del flujo de trabajo de diseño y provee la descripción de la distribución física del sistema. Se utiliza como entrada fundamental en las actividades de diseño e implementación. El diagrama de componentes, por su parte, estructura el modelo de implementación y muestra las relaciones entre los elementos de implementación. Ambos diagramas conforman el modelo de implementación, que describe como los elementos del diseño se implementan en términos de componentes tales como ficheros de código fuente, ejecutables, librerías y documentos.

4.1.1 Diagrama de componentes

El diagrama de componentes se representa como un grafo de componentes de software unido por medio de las relaciones de dependencia. Entre los componentes se encuentran las clases, interfaces, librerías y componentes ejecutables. Normalmente los diagramas de componentes se utilizan para modelar código fuente, versiones ejecutables, bases de datos físicas entre otros. El diagrama de componentes muestra un conjunto de ficheros relacionados entre sí para lograr una completa funcionalidad del sistema.

En la Figura 13 se muestra el diagrama de componentes del CU: *“Realizar alineamiento de secuencias”* para la sección *“Alinear con Clustalw”*. Los restantes diagramas de componentes se encuentran en el Anexo C.

controladorClustalw.java	Clase Java para procesar las peticiones de alineamiento con Clustalw como: validar formulario y mostrar resultado.
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.
clustalw.sh	Programa ejecutable, disponible como servicio web, el cual realiza el alineamiento con el algoritmo Clustalw.
DAO.java	Clase que declara los métodos necesarios para acceder a la base de datos..
DAOImplementado.java	Clase que implementa las funciones que permiten acceder a la base de datos.
UsuarioHome.java	Clase experta en la manipulación de la tabla usuario.
AlineamientoHome.java	Clase experta en la manipulación de la tabla alineamiento.
usuario	Tabla de la base de datos que representa los usuarios del sistema.
alineamiento	Tabla de la base de datos que representa los alineamientos realizados por el sistema.

4.1.2 Modelo de despliegue

Un diagrama de despliegue es un grafo de nodos unidos por conexiones. Un nodo puede contener instancias de componentes de software y objetos. En general un nodo es una unidad de computación de algún tipo, por ejemplo un sensor o un dispositivo. Las instancias de componentes de software pueden estar unidas por relaciones de dependencia, estas relaciones pueden ser múltiples ya que un componente puede tener más de una interfaz de comunicación.

En la Figura 14 se muestra el modelo de despliegue del sistema. Los nodos representados, se conectan mediante soportes bidireccionales que pueden a su vez estereotiparse. Esta vista permite determinar las consecuencias de la distribución y la asignación de los recursos. Las instancias de los nodos pueden contener instancias de ejecución, como instancias de componentes y objetos.

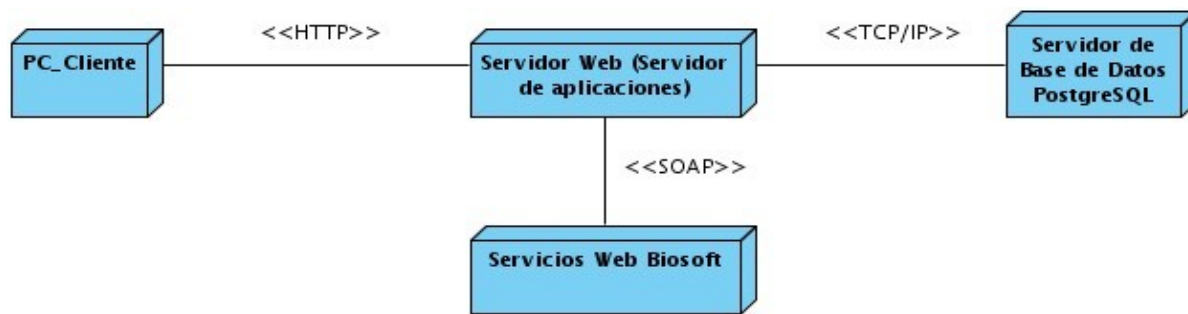


Figura 14: Modelo de despliegue del módulo básico. Se representan los componentes necesarios para realizar el despliegue del sistema y los protocolos de comunicación entre ellos.

Para el módulo básico 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, el cual encapsula la lógica de aplicación. 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.

4.2 Pruebas de software

Cuando un sistema tiene terminado una serie de clases y componentes, el software debe ser probado para descubrir y corregir el máximo de errores posible, antes de su entrega al cliente. El objetivo es diseñar una serie de casos de prueba que tengan una alta probabilidad de encontrar errores. Para ello se aplican una serie de técnicas de pruebas del software. Estas técnicas facilitan una guía sistemática para diseñar pruebas que: comprueben la lógica interna de los componentes del software y verifiquen los dominios de entrada y salida del programa para descubrir errores en la funcionalidad, el comportamiento y rendimiento[50].

Las pruebas definen el grado de aceptación del sistema que pueden tener los clientes del mismo, e incluso determinar si lo aceptan o no. Para ser más eficaces, las pruebas deberían ser realizadas por un equipo independiente, esto garantiza pruebas con alta probabilidad de encontrar errores, que es el objetivo principal de las pruebas. Existen dos tipos de pruebas, las de caja blanca y las de caja negra.

4.2.1 Pruebas de caja blanca

La prueba de caja blanca, denominada a veces prueba de caja de cristal, es un método de diseño de casos de prueba que usa la estructura de control del diseño procedimental para obtener los casos de prueba que serán aplicados[50].

Mediante los métodos de prueba de caja blanca se pueden obtener casos de prueba que garanticen que: se ejercite por lo menos una vez todos los caminos independientes de cada módulo; ejerciten todas las decisiones lógicas en sus vertientes verdaderas y falsas; ejecuten todos los bucles en sus límites y con sus límites operacionales; y ejerciten las estructuras internas de datos para asegurar su validez.

Uno de los métodos de prueba de caja blanca es el camino básico, la cual determina la complejidad ciclomática de una porción de código. La complejidad ciclomática es una métrica del software que proporciona una medición cuantitativa de la complejidad lógica de un programa[50]. Cuando se usa el camino básico, el valor calculado como complejidad ciclomática define el número de caminos independientes del conjunto básico de un programa y nos da un límite superior para el número de pruebas que se deben realizar. Esta complejidad se puede calcular de tres formas:

1. El número de regiones del grafo de flujo coincide con la complejidad ciclomática.
2. La complejidad ciclomática, $V(G)$, de un grafo de flujo G , se define como: $V(G)=A-N+2$. Donde A es el número de aristas del grafo de flujo y N es el número de nodos del mismo.
3. La complejidad ciclomática, $V(G)$, de un grafo de flujo G también se define como $V(G) = P + 1$. Donde P es el número de nodos predicado⁵ contenidos en el grafo de flujo G .

4.2.2 Pruebas de caja negra

Las pruebas de caja negra, también denominadas pruebas de comportamientos, se centran en los requisitos funcionales del software. O sea, la prueba de caja negra permite obtener conjuntos de condiciones de entrada que ejerciten completamente todos los requisitos funcionales de un programa.

La prueba de caja negra no es una alternativa a las técnicas de prueba de caja blanca. Más bien, se trata de un enfoque complementario que intenta descubrir diferentes tipos de errores que los que identifican los métodos de caja blanca[50]. La prueba de caja negra intenta identificar errores de las siguientes categorías: funciones incorrectas o ausentes, errores de interfaz, errores en estructuras de datos o en accesos a bases de datos externas, errores de rendimiento y errores de inicialización y de terminación. Existen varios métodos de caja negra entre los que se pueden mencionar los métodos de prueba basados en grafo, partición equivalente y análisis de valores al límite.

5 Nodos a partir de los que se puede tomar más de un camino.

La partición equivalente es un método de prueba de caja negra que divide el campo de entrada de un programa en clases de datos de los que se pueden derivar casos de prueba. Un caso de prueba ideal descubre de forma inmediata una clase de errores que, de otro modo, requerirían la ejecución de muchos casos antes de detectar el error genérico[50]. La partición equivalente se dirige a la definición de casos de prueba que descubran clases de errores, reduciendo así el número total de casos de prueba que hay que desarrollar.

Las clases de equivalencia se pueden definir de acuerdo con las siguientes directrices:

1. Si una condición de entrada especifica un rango, se define una clase de equivalencia válida y dos no válidas.
2. Si una condición de entrada requiere un valor específico, se define una clase de equivalencia válida y dos no válidas.
3. Si una condición de entrada especifica un miembro de un conjunto, se define una clase de equivalencia válida y una no válida.
4. Si una condición de entrada es lógica, se define una clase de equivalencia válida y una no válida.

Otro de los métodos de caja negra muy utilizado es el de análisis de valores al límite, ya que por razones que no están del todo claras, los errores tienden a darse más en los límites del campo de entrada que en el centro[50]. Por ello se ha desarrollado el análisis de valores límites (AVL) como técnica de prueba. Esta lleva a una elección de casos de prueba que ejerciten los valores límite. El análisis de valores al límite es una técnica de diseño de casos de prueba que complementa a la de partición equivalente.

4.3 Aplicación de pruebas de caja blanca

Como se mencionó en acápites anteriores uno de los métodos de prueba de caja blanca es la del camino simple, que se aplica a fragmentos de código de la aplicación. Se determinó aplicar este método a la función que genera secuencias aleatorias de nucleótidos.

Para la función “makenucseq” se identificaron los bloques de ejecución y se enumeraron para identificarlos (Figura 15). Se obtuvieron nueve bloques y se determinó el camino básico ilustrado en la Figura 16. Identificando en cada sentencia condicional un nodo predicado del cual se derivan más de un camino a seguir, tal es el caso de los nodos 2,4 y 6.

```

public String makenucseq(int amount, int length, boolean randomLength){
    Random r = new Random();
    System.out.println("Creando secuencia de nucleótidos");
    int cant = 1;
    String res = "";
    while (cant <= amount) {
        res += "\n" + ">Generate_" + cant + "\n";
        int auxLength = 0;
        if (randomLength) {
            length = r.nextInt(500) + 100;
        }
        while (auxLength < length) {
            int posAmino = r.nextInt(4);
            res += aminoacidosNuc.charAt(posAmino);
            auxLength++;
        }
        cant++;
    }
    return res;
}
    
```

Figura 15: Función que genera secuencias de nucleótidos aleatoriamente. El código se divide por bloques de ejecución, los cuales están enumerados y constituyen los nodos del camino básico.

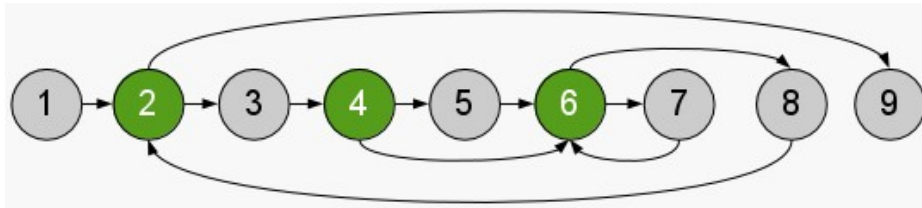


Figura 16: Camino básico de la función “makenucseq”. Los nodos resaltados en color verde corresponden a las condicionales del código de la función y por tanto son nodos predicados. Las aristas indican los posibles caminos a seguir a partir del nodo correspondiente.

Con el camino básico determinado, se aplica una de las tres formas para calcular la complejidad ciclomática, se utilizó la fórmula $V(G) = A - N + 2$, para la cual se obtuvo once aristas y nueve nodos, por lo tanto: $V(G) = 11 - 9 + 2$, quedando $V(G) = 4$. De la misma forma se pueden comprobar que las otras variantes explicadas de calcular la complejidad ciclomática arriban al mismo resultado. La complejidad ciclomática indica los posibles casos de ejecución para la función, lo que es muy útil a la hora de diseñar los casos de prueba de caja negra.

4.4 Casos de pruebas de caja negra

Para aplicar las pruebas al sistema se determinaron los casos de pruebas en un conjunto de entradas, condiciones de ejecución y resultados esperados, con tal de verificar el cumplimiento de un objetivo en particular.

Antes de ejecutar un caso de prueba es necesario definir una serie de variables que serán utilizadas. Para el CU: “Realizar alineamiento de secuencias” se definieron cuatro variables que se describen a continuación.

Tabla 5: Variables para el caso de prueba Alinear con Clustalw.

No.	Nombre de campo	Clasificación	Valor nulo	Descripción
1	secuencias	Campo de texto	No	Secuencia de los caracteres: ACSTQYGVLMPIFWDEKRHF
2	quicktree	Campo numérico	No	Número mayor que 0 y menor que 100.
3	pwgapopen	Campo numérico	No	Número mayor que 0.
4	helixendin	Campo numérico	No	Número mayor que 0.

Tabla 6: Caso de prueba 1: Alinear secuencia con Clustalw.

Escenario	Descripción	secuencias	quicktree	pwgapopen	helixendin	Respuesta del sistema	Flujo central	
EC 1.1	Introducir secuencias de proteínas y nucleótidos	Se llenan los campos correctamente y se alinean las secuencias	V ⁶ : Anexo D, Figura 22.	V: SLOW	V: 10.00	V: 3	Alinea las secuencias y muestra el resultado del alineamiento.	1- Alineamientos/ Alinear con Clustalw. 2- Se llenan los datos. 3- Enviar.
EC 1.2	Introducir valores en blanco	Se desea alinear dejando campos en blanco	V ⁷ : ()	V: SLOW	V: 10.00	V: 3	Muestra un mensaje de error en el campo vacío.	1- Alineamientos/ Alinear con Clustalw. 2- Se llenan los datos. 3- Enviar.
			V: Anexo D, Figura 22	I: ()	V: 10.00	V: 3		
			V: Anexo D, Figura 22	V: SLOW	I: ()	V: 3		
			V: Anexo D, Figura 22	V: SLOW	V: 10.00	I: ()		

6 Entrada válida

7 Entrada inválida

El caso de prueba de la Tabla 6 fue aplicado al CU “*Editar secuencia*”. De la misma forma se definió un escenario de prueba para el CU “*Generar secuencias aleatoriamente*”, las variables usadas para dicho caso de prueba se resumen en la Tabla 7.

Tabla 7: Variables para el caso de prueba Generar secuencia nucleotídica aleatoriamente.

No.	Nombre de campo	Clasificación	Valor nulo	Descripción
1	cantidad	Campo numérico	No	Número mayor que 0 y menor que 30000.
2	tamaño	Campo numérico	No	Número mayor que 0 y menor que 30000.
3	tamaño aleatorio	Campo lógico	No	N/A

Tabla 8: Caso de prueba 2: Generar secuencia nucleotídica aleatoriamente.

Escenario	Descripción	canti dad	tam año	tamaño aleatorio	Respuesta del sistema	Flujo central
EC 1.1 Generar secuencias de nucleótidos de tamaño invariable	Se completan los campos correctamente	V: 20	V: 200	V: No ⁸	Crea 20 secuencias con 200 caracteres de longitud.	1- Edición/ Editor de secuencias/ Crear secuencia nucleotídicas 2- Llenar los datos 3- Aceptar
EC 1.2 Introducir datos incorrectos.	Se quiere generar una secuencia con datos inválidos	I: ()	V: 250	V: Si	Muestra una ventana de error, especificando los errores.	1- Edición/ Editor de secuencias/ Crear secuencia nucleotídicas 2- Llenar los datos 3- Aceptar
		V: 100	I: ()	V: No		
		V: 100	I: ()	V: No		

La aplicación de los casos de prueba permitió verificar el cumplimiento de los requisitos funcionales del sistema, de igual forma corroboró la robustez del sistema en el manejo de errores y situaciones anormales.

⁸ El campo “tamaño aleatorio” es un caja de marca (checkbox), que se toma como No si no está marcada y como Si en caso contrario.

4.5 Conclusiones del capítulo

En este capítulo se construyó el modelo de implementación correspondiente al módulo básico, para ello se realizó el diagrama de componentes del CU: *“Realizar alineamiento de secuencias”*, específicamente para la sección *“Alinear con Clustalw”*, en el que se definieron los componentes del sistema necesario para el correcto funcionamiento de este CU. Los componentes fueron separados por paquetes tal y como se definió en el modelo del diseño del sistema y se describió cada uno de ellos para lograr una mayor comprensión acerca de su función.

Además se definió el modelo del despliegue del módulo básico que muestra los elementos necesarios para realizar el despliegue e instalación del sistema: al menos un ordenador cliente, un servidor de aplicaciones, un servidor de bases de datos y un servidor de servicios web.

Se analizaron las principales técnicas y métodos de pruebas de software, tanto de caja blanca como de caja negra. Se determinó aplicar el método del camino básico, correspondiente al método de caja blanca, a la función encargada de generar secuencias de nucleótidos aleatoriamente, la cual arrojó un valor de 4 como complejidad ciclomática de la misma. Por otro lado se definieron los casos de prueba de caja negra para la técnica de partición de equivalencia y análisis de valores al límite al CU: *“Realizar alineamiento de secuencias”*, los que se aplicaron a la sección: *“Alinear con Clustalw”* y a la función que genera secuencias de nucleótidos aleatoriamente. Estas pruebas se realizaron a través de la interfaz del sistema y se detectaron inconformidades como campos no validados y errores no tratados; todas ellas solventadas rápidamente mediante la corrección de los objetos correspondientes.

Conclusiones generales

Una vez concluida la investigación se puede afirmar que se desarrolló el módulo básico de la Plataforma de Servicios Bioinformáticos de la Universidad de las Ciencias Informáticas. Para ello:

- Se determinó el uso de componentes de interfaz web denominados portlets, que proveen información en formato HTML y pueden ser utilizados por otras aplicaciones indistintamente de las tecnologías usadas en su confección. Ello implica que el sistema puede ser trasladado o utilizado fácilmente por otras aplicaciones.
- Se definieron los requisitos del módulo básico y se agruparon los RF en 11 CU, identificando como arquitectónicamente significativo el CU *“Realizar alineamiento de secuencias”*. La arquitectura se confeccionó con la utilización de varios patrones arquitectónicos y de diseño, destacándose SOA y el MVC. Los mismos fueron aplicados en la realización de los diagramas de interacción y del modelo del diseño, obteniendo un sistema separado en capas, altamente escalable y menos vulnerable al cambio.
- Se implementaron los requisitos funcionales identificados, obteniendo un sistema capaz de realizar alineamientos múltiples de secuencias por varios programas como: Clustalw, Muscle y T-Coffee. El sistema además permite realizar diversos análisis sobre secuencias moleculares tales como: la edición de las mismas, la identificación de marcos abiertos de lecturas, la visualización de propiedades químico-físicas de las proteínas, y la generación de secuencias aleatorias tanto proteicas como nucleotídicas.
- Se validaron las funcionalidades implementadas a través de la interfaz del sistema, mediante el diseño de pruebas de caja negra de partición de equivalencia y de análisis de valores al límite. Las pruebas permitieron encontrar inconformidades en la validación de datos y el tratamiento de errores que contribuyeron a aumentar la calidad del sistema.

Recomendaciones

Debido al gran número de archivos generados por el sistema, se recomienda establecer un estándar para almacenarlos en el servidor de aplicaciones.

Teniendo en cuenta que los archivos generados por el sistema pueden a largo plazo ocupar un espacio de disco considerable se recomienda crear un portlet para la gestión de los mismos.

Bibliografía

- 1: Tirso, P.; Montero, L. A.; Febles, J. P.. November 2007. *Computational Biology in Cuba: An Opportunity to Promote Science in a Developing Country*. Plos computational biology, 3, (7): 2047-2051
- 2: Mount, D. W.. *Bioinformatics sequence and genoma analysis*. Lugar de publicación desconocido. Cold Spring Harbor Laboratory Press. 2001. 565.
- 3: Oracle Corporation. *The Java EE 5 Tutorial*. E.E. U.U.. Editor desconocido. June, 2010. 1084.
- 4: Michelle, D.; David, S.; Joseph, T.; Francis, B.. 2011. *The 2011 bioinformatics links directory update: more resources, tools and databases and features to empower the bioinformatics community*. Nucleic Acids Research, 39, (Web Server issue): W3-W7
- 5: Galperin, M.; Fernandez Suarez, X.. 2012. *The 2012 Nucleic Acids Research Database Issue and the online Molecular Biology Database Collection*. Nucleic Acids Research, 40, (Database issue): D1-D8
- 6: NCBI. *NCBI, Our Mission*. [en línea], 2004. [Consultado el: 5 de diciembre de 2011]. Disponible en: [<http://www.ncbi.nlm.nih.gov/About/glance/ourmission.html>].
- 7: EBI. *Welcome to the EBI*. [en línea], 2011. [Consultado el: 5 de diciembre de 2011]. Disponible en: [<http://www.ebi.ac.uk/Information/>].
- 8: Expasy. *Expasy Bioinformatics Resources Portal*. [en línea], 2011. [Consultado el: 5 de diciembre de 2011]. Disponible en: [<http://expasy.org/>].
- 9: Rice, P.; Longden, I.; Bleasby, A.. 2000. *EMBOSS: The European Molecular Biology Open Software Suite*. Trends in Genetics, 16, (6): 276-277
- 10: Thompson J. D.; Higgins D. G.; Gibson T. J.. 1994. *CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice*. Nucleic Acids Research, 22, (22): 4673-4680
- 11: Edgar, R. C.. 2004. *MUSCLE: multiple sequence alignment with high accuracy and high throughput*. Nucleic Acids Research, 32, (5): 1792-1797
- 12: Notredame, C.; Higgins, D.; Heringa, J.. 2000. *T-Coffee: A novel method for fast and accurate multiple sequence alignment*. J Mol Biol., 302, (1): 205-217
- 13: EBI. *Web services map*. [en línea], 2011. [Consultado el: 2 de febrero de 2012]. Disponible en: [http://www.ebi.ac.uk/~bren/alberto_map.html].
- 14: EBI. *About Web Services*. [en línea], 2012. [Consultado el: 2 de febrero de 2012]. Disponible en: [<http://www.ebi.ac.uk/Tools/webservices/>].
- 15: NCBI. *Web Services Detail*. [en línea], 2012. [Consultado el: 4 de marzo de 2012]. Disponible en: [http://www.ncbi.nlm.nih.gov/entrez/query/static/esoap_help.html].

- 16: DDBJ. *Reference Web Services*. [en línea], 2012. [Consultado el: 5 de marzo de 2012]. Disponible en: [<http://www.ddbj.nig.ac.jp/search/references-e.html#explain>].
- 17: Erl, T.. *Service-Oriented Architecture: Concepts, Technology, and Design*. Lugar de publicación desconocido. Prentice Hall PTR. Agosto, 2005. 792.
- 18: Mateu, C.. *Desarrollo de aplicaciones web*. Cataluña, España. Eureka Media, SL. 2004. 378.
- 19: Wall, C.; Ryan Breidenbach, S.. *Spring in Action 2*. E.E. U.U.. Manning Publications Co.. 2008. 765.
- 20: King Ritchie, S.. *The Top 10 Programming Languages*. [en línea], October, 2011. [Consultado el: 3 de febrero de 2012]. Disponible en: [<http://spectrum.ieee.org/at-work/tech-careers/the-top-10-programming-languages>].
- 21: Valade, J.. *PHP 5 for dummies*. Lugar de publicación desconocido. Wiley Publishing, Inc. 2004. 411.
- 22: González Duque, R.. 2009. Python para todos. Creative Commons España. 115.
- 23: Cox, K.. *ASP.NET for dummies*. Indianapolis, Indiana. Wiley Publishing, Inc. 2008. 435.
- 24: Sanderson, S.. *Pro ASP.NET MVC 2 Framework*. E.E. U.U.. Apress. 2010. 778.
- 25: Allamaraju, S; Beust, C., et al. *Programación Java Server con J2EE*. Lugar de publicación desconocido. Anaya. 2003. 1206.
- 26: Patzer, A.. *JSP Ejemplos prácticos*. Madrid, España. Anaya. 2002. 308.
- 27: Clay Richardson, W.; Avondolio, D.; Vitale J.; Len P.; Smith K.. *Professional Portal Development with Open Source Tools*. Indianapolis, Indiana. Wiley Technology Publishing. 2004. 459.
- 28: Apache Portals Jetspeed Team. *Jetspeed-2 Open Source Enterprise Portal. Features*. [en línea], 2011. [Consultado el: 1 de diciembre de 2011]. Disponible en: [<http://portals.apache.org/jetspeed-2/features.html>].
- 29: Departamento de Ventas de Red Hat e Información General JBoss. *Plataforma de aplicaciones JBoss enterprise para portales*. [en línea], 2007. [Consultado el: 1 de diciembre de 2011]. Disponible en: [www.redhat.es/jboss].
- 30: Jonas, J.. *Liferay Portal Enterprise Intranets*. Birmingham, E.E. U.U.. Packt Publishing. 2008. 408.
- 31: RICH SEZOV, JR.. *Liferay in action*. Shelter Island, New York, E.E. U.U.. Manning Publications Co.. 2010. 378.
- 32: Liferay Inc.. *Liferay Inc, Premios y reconocimientos*. [en línea], 2011. [Consultado el: 3 de diciembre de 2011]. Disponible en: [<http://www.liferay.com/es/about-us/awards>].
- 33: Brown, D.; Chad Michael, D.. *Struts2 in action*. Lugar de publicación desconocido. Manning Publications Co.. 2007. 176.

- 34: Peak, P.; Heudecker, N.. *Hibernate Quickly*. Greenwich, Londres, U.K.. Manning Publications Co.. 2006. 453.
- 35: Roughley, I.. *Starting Struts 2*. [en línea], 2006. [Consultado el: 3 de diciembre de 2011]. Disponible en: [<http://infoq.com/minibooks/starting-struts2>].
- 36: Burns, E.; Schalk, C.; Griffin, N.. *JavaServer Faces 2.0: The Complete Reference*. New York, E.E. U.U.. McGraw-Hill Companies. 2010. 753.
- 37: Minter, D.. *Beginning Spring 2. From Novice to Professional*. New York, E.E. U.U.. Apress. 2008. 298.
- 38: The jQuery Project. *jQuery jQuery is a new kind of JavaScript Library*. [en línea], 2011. [Consultado el: 20 de marzo de 2012]. Disponible en: [<http://jquery.com/>].
- 39: Zammetti, F.. *Practical DWR 2 Projects*. New York, E.E. U.U.. Apress. 2008. 570.
- 40: Potts, S.; Kopack, M.. *Web Services in 24 Hour*. Indianapolis, Indiana. SAMS. Mayo, 2003. 443.
- 41: Apache Software Foundation. *Welcome to Apache Axis2/C*. [en línea], 2009. [Consultado el: 21 de marzo de 2012]. Disponible en: [<http://axis.apache.org/axis2/c/core/>].
- 42: Gilfillan, I.. *La biblia MySQL*. Lugar de publicación desconocido. Anaya. 2009. 841.
- 43: Zeis, C.; Ruel, C.; Wessler, M.. *Oracle® 11g For Dummies®*. Indianapolis, Indiana. Wiley Publishing, Inc.. 2009. 412.
- 44: Riggs, S.; Krosing, H.. *PostgreSQL 9. Administration Cookbook*. Birmingham, E.E. U.U.. Packt Publishing. Octubre, 2010. 360.
- 45: Larman, C.. *UML y Patrones. Introducción al análisis y diseño orientado a objetos*. México. Prentice Hall. 1999. 499.
- 46: Eclipse Project Framework. *Concept: Use Case*. [en línea], 2011. [Consultado el: 19 de marzo de 2012]. Disponible en: [http://epf.eclipse.org/wikis/openup/core.tech.common.extend_supp/guidances/concepts/use_case_BB199D1B.html?nodeId=e37dcf94].
- 47: Camacho, E.; Fabio, C.; Gabriel, N.. *Arquitecturas de Software. Guía de estudio*. [en línea], Abril, 2004. [Consultado el: Fecha consulta: 2012/02/21]. Disponible en: [<http://prof.usb.ve/lmendoza/Documentos/PS-6116/Guia%20Arquitectura%20v.2.pdf>].
- 48: Garland, J.; Anthony, R.. *Large-Scale Software Architecture*. Great Britain. John Wiley & Sons Ltd. 2003. 281.
- 49: Bass L.; Clements P.; Kazman R.. *Software Architecture in Practice*. Boston, E.E. U.U.. Addison Wesley. April, 2003. 560.
- 50: Pressman, Roger S.. *Ingeniería del software. Un enfoque práctico*. Lugar de publicación desconocido. Mc Graw Hill. 1998. 614.

A. Diagramas de clases del diseño

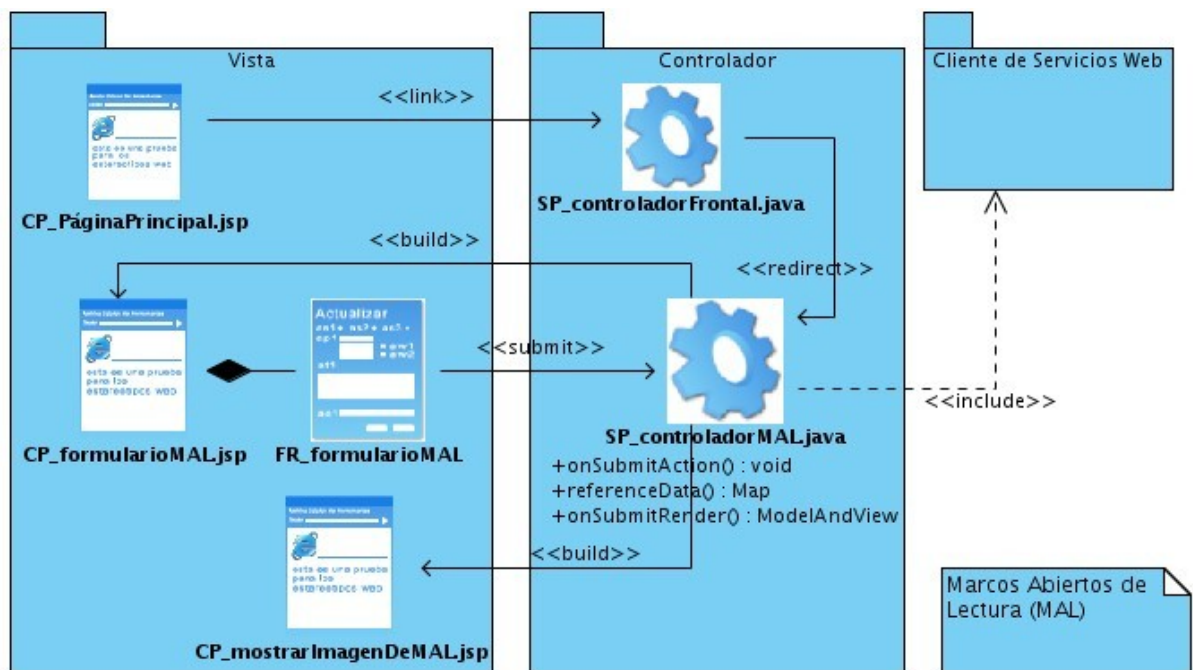


Figura 17: Diagrama del diseño para el CU Mostrar MAL de secuencias nucleotídicas.

B. Diagramas de interacción

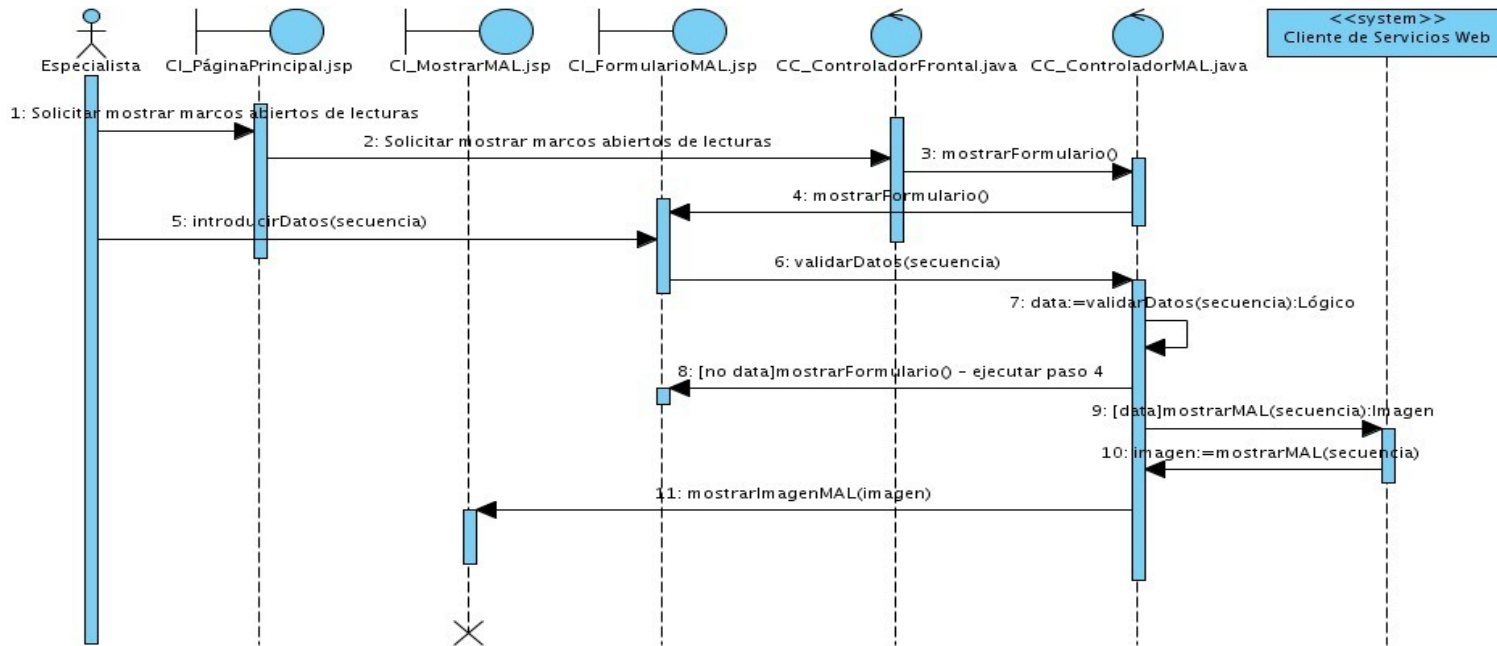


Figura 18: Diagrama de secuencia del CU Mostrar MAL de secuencias nucleotídicas.

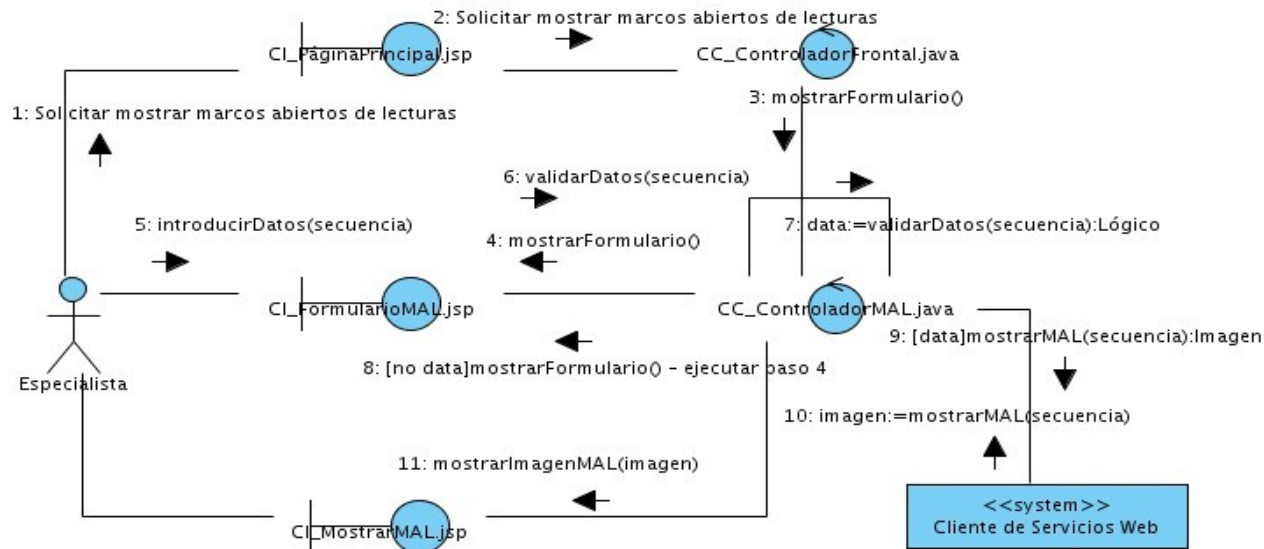


Figura 19: Diagrama de colaboración del CU Mostrar MAL de secuencias nucleotídicas.

C. Diagramas de componentes

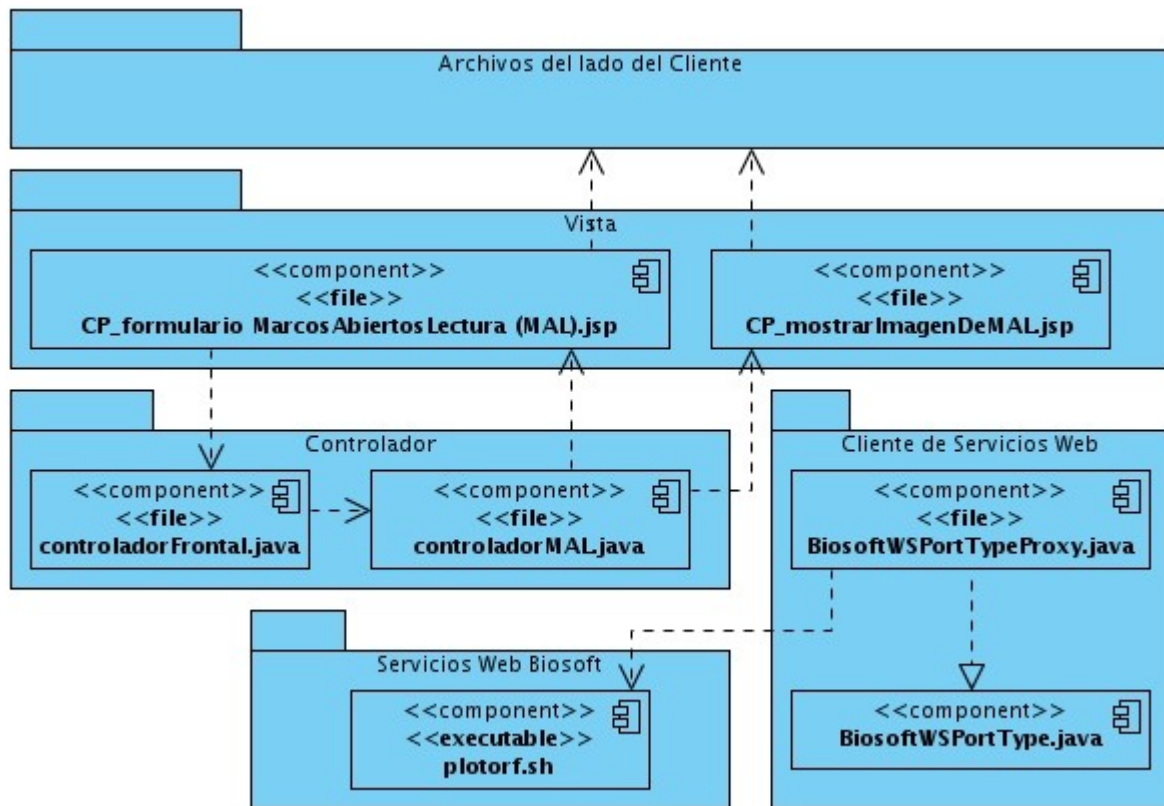


Figura 20: Diagrama de componentes del CU Mostrar MAL de secuencia nucleotídica.

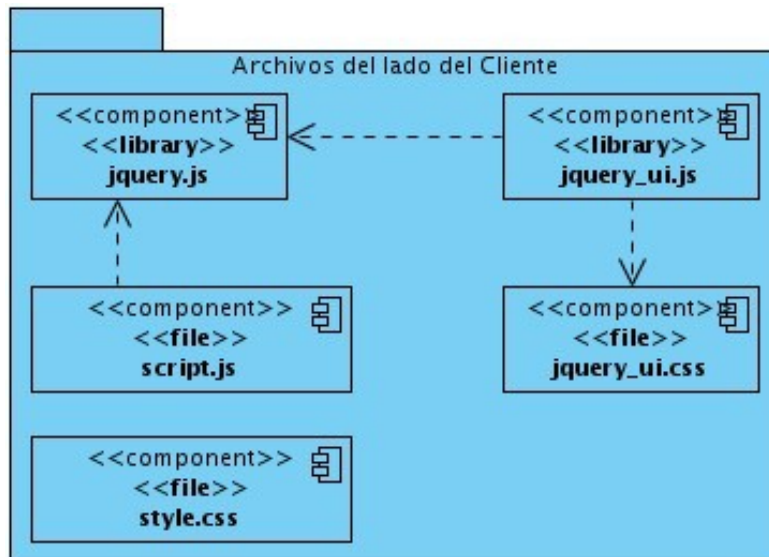


Figura 21: Diagrama de componentes del paquete Archivos del lado del cliente.

D. Secuencia utilizada para las pruebas

>gi|1172128|gb|AAA86618.1| SipA [Salmonella enterica subsp. enterica serovar Typhimurium]

MVTSVRTQPPVIMPGMQTEIKTQATNLAANLSAVRAKCHSDAVREIKGPQLEDFPALIKQA
SLDALFKCGKDAEALKEVFTNSNNVAGKKAIMEFAGLFRSALNATSDSPEAKTLLMKVGAEY
AAQIIKDGLKEKSAFGPWLPEPKKAEAKLENLEKQLLDIIKNNTGGELSKLSTNLVMQEVMPYI
ASCIEHNFGCTLDPLTRSNLTHLVDKAAAKAVEALDMCPQKLTQEQGTSVGREARHLEMQT
LIPLLLRNVFAQIPADKLPDPKPEPAAGPVPDGGKKAAPTGINININIDSSNHSVDNSKHINNA
EPVDNGQRHIDNSNHDNSRKTIDNSRTFIDNSQRNGESHSTNSSNVSHSHSRVDSTTHQT
ETAHSASTGAIDHGIAGKIDVTAHATAEAVTNASSESKDGVVTSEKGTGTTGETTSFDEVDGV
TSKSIIGKPVQATVHGVDDNKQQSQTAIEVNVKPLASQLAGVENVKDTLQSDTTVITGNKA
GTTDNDNSQTDKTGPFSGLKFKQNSFLSTVPSVTNMHSMHFDARETFLGVIRKALEPDTST
PFPVRRAFDGLRAEILPNDTIKSAALKAQCSDIDKHPELKAKMETLKEVITHHPQKEKLAEL
QFAREAGLTRLKGETDYVLSNVL DGLIGDGSWRAGPAYESYLNKPGVDRVITVDGLHMQR

>gi|62181384|ref|YP_217801.1| cell invasion protein [Salmonella enterica subsp. enterica serovar Choleraesuis str. SC-B67]

MVTSVRTQPPVIMPGMQTEIKTQATNLAANLSAVRESATATLSGEIKGPQLEDFPALIKQA
SLDALFKCGKDAEALKEVFTNSNNVAGKKAIMEFAGLFRSALNATSDSPEAKTLLMKVGAEY
TAQIIKDGLKEKSAFGPWLPEPKKAEAKLENLEKQLLDIIKNNTGGELSKLSTNFVMQEVMPYI
ASCIEHNFGCTLDPLTRSNLTHLVDKAAAKAVEALDMCHQKLTQEQGTSVGREARHLEMQT
LIPLLLRNVFAQIPADKLPDPKPEPAAGPVPDGGKKAAPTGINININIDSSNHSVDNSKHINNS
RSHVDNSQRHIDNSNHDNSRKTIDNSRTFIDNSQRHGESHHSTNSSNVGHSHSRVDSTTH
QTETAHSASTGAIDHGIAGKIDVTAHATAEAVTNASSESKDGVVTSEKGTGTTGETTSFDEVD
GVTSKSIIGKPVQATVHGVDDNKQQSQTAIEVNVKPLASQLAGVENVKDTLQSDTTVITGN
KAGTTDNDNSQTDKTGPFSGLKFKQNSFLSTVPSVTNMHSMHFDARETFLGVIRKALEPDT
STPFPVRRAFDGLRAEILPNDTIKSAALKAQCSDIDKHPELKAKMETLKEVITHHPQKEKLAEL
ALQFAREAGLTRLKGETDYVLSNVL DGLIGDGSWRAGPAYESYLNKPGVDRVITVDGLHM
QR

Figura 22: Secuencia de proteína válida.

E. Descripciones de los CU del Sistema

Tabla 9: Descripción del CU Administrar alineamiento.

Actores:	Especialista
Propósito:	Permitir al especialista realizar alineamientos, eliminarlos, consultarlos y actualizarlos.
Resumen:	El CU se inicia cuando el especialista accede a la página principal del módulo de análisis de secuencias, donde puede elegir opciones como realizar alineamiento, listar alineamientos para eliminarlos o ver el resultado de estos.
Referencia:	RF1
Precondiciones:	El especialista debe estar autenticado y debe tener al menos un alineamiento realizado.
Postcondiciones:	El especialista obtiene el resultado del alineamiento.
Flujo normal de eventos	
Acción del Actor	Respuesta del sistema
1 El especialista accede a la página principal del módulo de análisis de secuencia	1.1 El sistema muestra las opciones: <ul style="list-style-type: none"> Realizar alineamiento Ver alineamientos
2 El especialista escoge una opción.	2.1 Si escoge: <ul style="list-style-type: none"> Realizar alineamiento, ver CU “Realizar alineamiento de secuencias”. Ver alineamientos, ver sección Listar alineamientos.
Sección Listar alineamientos	
	1.1 El sistema muestra una lista con todos los alineamientos del especialista en cuestión.
2 El especialista escoge una opción.	2.1 Si escoge: <ul style="list-style-type: none"> Consultar resultado de alineamiento, ver sección consultar resultado de alineamiento. Eliminar alineamiento, ver sección eliminar alineamiento.
Sección consultar resultado de alineamiento	
1 El especialista selecciona un alineamiento.	1.1 El sistema busca el alineamiento seleccionado y muestra su resultado en una nueva ventana.
Sección eliminar alineamiento	
1 El especialista selecciona un alineamiento.	1.1 El sistema elimina el alineamiento seleccionado y muestra todos los alineamientos restantes del especialista.

Prioridad	Crítico
------------------	---------

Tabla 10: Descripción del CU Visualizar moléculas en tres dimensiones.

Actores:	Especialista
Propósito:	Visualizar estructuras moleculares en el espacio tridimensional.
Resumen:	El CU se inicia cuando el especialista accede al visualizador molecular.
Referencia:	RF3
Precondiciones:	
Postcondiciones:	El sistema muestra en el visualizador molecular la estructura especificada.
Flujo Normal de Eventos	
Acción del Actor	Respuesta del sistema
1 El especialista accede a la página principal del visualizador molecular.	1.1 El sistema muestra la página del visualizador mostrando una estructura de ejemplo.
2 El especialista selecciona la opción subir fichero.	2.2 El sistema muestra un navegador de directorios para que el especialista busque el fichero deseado.
3 El especialista selecciona el fichero y presiona "Aceptar". 3.1 El especialista envía el fichero al servidor.	3.2 El sistema recibe el fichero y lo muestra en el visualizador molecular.
Flujo alterno de eventos	
3 El especialista entra en fichero no válido.	3.1 El sistema muestra una página mostrando el error correspondiente.
Prioridad	Secundario

Tabla 11: Descripción del CU Visualizar marcos abiertos de lectura.

Actores:	Especialista
Propósito:	Mostrar un esquema con los marcos abiertos de lectura de una secuencia.
Resumen:	El CU se inicia cuando el especialista selecciona la opción "Mostrar marcos abiertos de lectura" en la página inicial del módulo "Análisis de secuencia".
Referencia:	RF4

Precondiciones:	
Postcondiciones:	El especialista cuenta con los marcos abiertos de lectura mostrados por el sistema en formato de imagen.
Flujo Normal de Eventos	
Acción del Actor	Respuesta del sistema
1 El especialista selecciona la opción "Mostrar marcos abiertos de lectura".	1.1 El sistema muestra la página con el formulario para que el usuario entre los datos necesarios.
2	2.1 El sistema muestra una imagen con los marcos abiertos de lectura de la secuencia especificada.
Flujo alterno de eventos	
2 El especialista entra los datos incorrectos.	2.1 El sistema muestra los errores correspondientes, especificando a que campos se refieren los mismos. 2.2 El sistema retorna al paso 1.1 del flujo normal de eventos
Prioridad	Secundario

Tabla 12: Descripción del CU Visualizar propiedades químico-físicas de secuencias proteicas.

Actores:	Especialista
Propósito:	Mostrar un esquema con las propiedades amino-acídicas de la secuencia entrada.
Resumen:	El CU se inicia cuando el especialista selecciona la opción "Mostrar propiedades amino-acídicas" en la página inicial del módulo "Análisis de secuencia".
Referencia:	RF5
Precondiciones:	
Postcondiciones:	El sistema muestra una imagen con las propiedades químico-físicas de la secuencia especificada.
Flujo Normal de Eventos	
Acción del Actor	Respuesta del sistema
1 El especialista selecciona la opción "Mostrar propiedades químico-físicas".	1.1 El sistema muestra la página con el formulario para que el usuario entre los datos necesarios.
2 El especialista llena los datos: - sequence, aaproperties, aahydrophathy, hwindow, graph, generalplot, hydrophathyplot.	2.1 El sistema muestra una imagen con las propiedades químico-físicas de la secuencia entrada.

Flujo alterno de eventos	
2 El especialista entra los datos incorrectos.	2.1 El sistema muestra los errores correspondientes, especificando a que campos se refieren los mismos. 2.2 El sistema retorna al paso 1.1.
Prioridad	Secundario

Tabla 13: Descripción del CU Visualizar segmentos de transmembrana de proteínas.

Actores:	Especialista
Propósito:	Mostrar un esquema con los segmentos de transmembrana de la secuencia entrada.
Resumen:	El CU se inicia cuando el especialista selecciona la opción “Mostrar segmentos de transmembrana” en la página inicial del módulo “Análisis de secuencia”.
Referencia:	RF6
Precondiciones:	
Postcondiciones:	El sistema muestra una imagen con los segmentos de transmembrana de la secuencia especificada.
Flujo Normal de Eventos	
Acción del Actor	Respuesta del sistema
1 El especialista selecciona la opción “Mostrar segmentos de transmembrana”.	1.1 El sistema muestra la página con el formulario para que el usuario entre los datos necesarios.
2 El especialista llena los datos: - sequence, graph, rformat.	2.1 El sistema muestra una imagen con los segmentos de transmembrana de la secuencia especificada.
Flujo alterno de eventos	
2 El especialista entra los datos incorrectos.	2.1 El sistema muestra los errores correspondientes, especificando a que campos se refieren los mismos. 2.2 El sistema retorna al paso 1.1.
Prioridad	Secundario

Tabla 14: Descripción del CU Generar secuencias aleatoriamente.

Actores:	Especialista
Propósito:	Generar secuencias de nucleótidos con caracteres aleatorios y de tamaño aleatorio o especificado por el especialista.
Resumen:	El CU se inicia cuando el especialista selecciona la opción

	"Generar secuencias aleatoriamente" en el editor de secuencias.
Referencia:	RF7
Precondiciones:	
Postcondiciones:	El sistema genera la cantidad de secuencias especificadas.
Flujo Normal de Eventos	
Acción del Actor	Respuesta del sistema
1 El especialista selecciona la opción "Generar secuencias aleatoriamente".	1.1 El sistema muestra el formulario para entrar ellos datos.
2 El especialista entra los datos: - tamaño, cantidad, proteica.	2.1 El sistema muestra las secuencias generadas aleatoriamente.
Flujo alterno de eventos	
2 El especialista entra los datos incorrectos.	2.1 El sistema muestra los errores correspondientes, especificando a que campos se refieren los mismos. 2.2 El sistema retorna al paso 1.1.
Prioridad	Secundario

Tabla 15: Descripción del CU Mostrar características básicas de una secuencia.

Actores:	Especialista
Propósito:	Mostrar las características de una secuencia.
Resumen:	El CU se inicia cuando el especialista selecciona la opción "Información de secuencia" en el módulo de análisis de secuencia.
Referencia:	RF8
Precondiciones:	
Postcondiciones:	El sistema extrae las características de la secuencia y muestra el resultado.
Flujo Normal de Eventos	
Acción del Actor	Respuesta del sistema
1 El especialista selecciona la opción "Información de secuencia".	1.1 El sistema muestra el formulario para entrar la secuencia.
2 El especialista llena los datos: -sequence, delimiter, html, heading, name, accession, type, length, pgc.	2.1 El sistema muestra las características de la secuencia en una página nueva.

Flujo alterno de eventos	
2 El especialista entra una secuencia inválida.	2.1 El sistema muestra el error correspondiente, especificando el campo en cuestión. 2.2 El sistema retorna al paso 1.1.
Prioridad	Secundario

Tabla 16: Descripción del CU Editar secuencia.

Actores:	Especialista
Propósito:	Realizar la transformación de secuencias para ajustarlas a las necesidades de determinados análisis.
Resumen:	El CU se inicia cuando el especialista selecciona la opción “Editor de secuencia” en el módulo de análisis de secuencia.
Referencia:	
Precondiciones:	El especialista debe especificar una secuencia.
Postcondiciones:	El sistema muestra la secuencia original y la secuencia con los cambios realizados.
Flujo Normal de Eventos	
Acción del Actor	Respuesta del sistema
1 El especialista selecciona la opción “Editor de secuencia”.	1.1 El sistema muestra la página inicial del editor de secuencias con un formulario para entrar la secuencia a editar.
2 El especialista entra una secuencia en el formulario.	2.2 El sistema muestra las opciones: <ul style="list-style-type: none"> • Invertir secuencia • Complementar secuencia • Unir secuencias • Ordenar secuencias por tamaño • Extraer subsecuencia
Sección: Invertir secuencia	
1 El especialista selecciona la opción “invertir secuencia” del editor de secuencias.	1.1 El sistema invierte la secuencia y muestra el resultado en un campo de texto.
Sección: Complementar secuencia	
1 El especialista selecciona la opción “Complementar secuencia” del editor de secuencias.	1.1 El sistema complementa la secuencia y muestra el resultado en un campo de texto.
Sección: Unir secuencias	

1 El especialista selecciona la opción "Unir secuencia" del editor de secuencias.	1.1 El sistema une las subsecuencias de la secuencia especificada y muestra el resultado en un campo de texto.
Sección: Ordenar secuencias por tamaño	
1 El especialista selecciona la opción "Ordenar secuencias por tamaño" del editor de secuencias.	1.1 El sistema muestra un formulario para que el especialista especifique si en orden ascendente o descendente.
2 El especialista escoge las opciones y presiona "Aceptar".	2.1 El sistema ordena las subsecuencias de la secuencia especificada y muestra el resultado en un campo de texto.
Sección: Extraer subsecuencias	
1 El especialista selecciona la opción "Extraer subsecuencia" del editor de secuencias.	1.1 El sistema muestra un formulario para que el especialista indique la posición inicial y final de la región a extraer.
2 El especialista entra los datos: - inicio, fin.	1.1 El sistema extrae la subsecuencia que abarca la región especificada y muestra el resultado en un campo de texto.
Flujo alterno	
1 El especialista entra posiciones incorrectas.	1.1 El sistema muestra un mensaje de error. 1.2 El sistema retorna al paso 1.1 del flujo normal de eventos.
Prioridad	Secundario