

Universidad de las Ciencias Informáticas

Facultad 6



**Título:** “Módulo de visualización de servicios para la plataforma  
alasSiRNA-Design v1.0”

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

**Autor:** Ana Rosa Montané Cobas

**Tutor:** Ing. Tonysé de la Rosa Martín

**Co-tutor:** Ing. Beatriz Lara Osorio

La Habana, Junio de 2011

“Año 54 de la Revolución”



***“Cuando menos lo esperamos, la vida nos coloca delante un desafío que pone a prueba nuestro coraje y nuestra voluntad de cambio.”***

***Paulo Coelho***

**DECLARACIÓN DE AUTORÍA**

Declaro ser autor del presente trabajo “Módulo de visualización de servicios para la plataforma alasSiRNA-Design v1.0.” y reconozco a la Universidad de las Ciencias Informáticas (UCI) los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmo la presente a los \_\_\_\_ días del mes de \_\_\_\_\_ del año \_\_\_\_\_.

**Ana Rosa Montané Cobas**

---

Firma del autor

**Tonysé de la Rosa Martín**

---

Firma del Tutor

**Beatriz Lara Osorio**

---

Firma del Co-tutor

## DATOS DE CONTACTO.

---

### DATOS DE CONTACTO

**Tutor:**

Ing. Tonysé de la Rosa Martín  
Universidad de las Ciencias Informáticas, La Habana, Cuba.  
Email: tdelarosa@uci.cu

**Co-tutor:**

Ing. Beatriz Lara Osorio.  
Universidad de las Ciencias Informáticas, La Habana, Cuba.  
Email: blara@uci.cu

### AGRADECIMIENTOS

La gratitud es el más legítimo pago al esfuerzo ajeno, además de necesaria es hermosa; por eso le agradezco:

A mi madre Mayra Cobas por ser mi ejemplo a seguir, mi guía en la vida, mi mejor amiga y mi TODO.

A mi padre Alcibiades Montané por enseñarme muchas cosas de la vida, por su preocupación y dedicación. Por ser mi hombre perfecto.

A mi hermanita Roxana por ser el otro pedacito de mí y con quien puedo ser yo misma sin ataduras.

A mi hermano Alexander que aunque ahora estamos muy separados, su compañía en el primer año de la Universidad fue una de las mejores cosas que me han pasado en la vida, se convirtió en el hermano mayor que siempre quise a mi lado.

A mis hermanos Nory y Norberto por siempre estar al tanto de mí y por brindarme todo lo que tienen.

A mis primos, mis tías y mi abuela, que siempre se preocupan por mí y me han brindado todo su apoyo. En especial a mi tía Mary Elena y a mi prima Yaniet que en estos 5 años han estado muy cerca de mí.

A mi amiga Karelia, que aunque es muy olvidadiza, es la persona en la que más confié y mi confidente. Es la que mejor sabe aconsejarme y de ella he aprendido a no conformarme con tan poco.

A mi amiga Marla, que aunque es muy pesimista, ha sido mi pegolín en estos cinco años. Ha aguantado mis perretas, me ha enseñado el significado de la palabra amistad y nunca me ha abandonado.

A mi amiga Arianna por su confianza, su preocupación, por ser la mejor compañera de aula que he tenido en todos mis años de estudio y que a pesar de no estar con nosotros en la universidad en estos últimos 4 años aún la extraño muchísimo.

A mi amiga Claudia por su manera de ver la vida, por sus consejos, por su alegría y por ser una persona en la cual siempre puedo confiar.

A mi amiga Juliet por requerirme siempre cuando hago algo mal, por su locura y cordura a la vez en todo lo que hace y por siempre preocuparse por mis problemas.

A mi amiga Nelly que me ha aguantado los últimos tres años como compañera de cuarto y ha sido comprensiva con mi reguero. Por poder contar siempre con ella.

A mi amigo Ángel, por ser una persona incondicional y especial con todos y por contar con él siempre.

A mi amigo Yonnis, que aunque no nos soportábamos en primer año, se convirtió en uno de mis mejores amigos.

A mi amigo Dailig por su amistad incondicional y su cariño.

A mi amiga Adriana y Margarita por siempre contar con ella en todo momento.

A mi amiga Dalied por ser mi compañera de andanzas en 2do año y por sus consejos.

A mi amiga Yennia por su carisma y su alegría.

A mis amigos Diosvelis y Eudel por su amistad de 8 años. A Diosvelis por cuidarme siempre y a Eudel por ser siempre tan divertido.

A mi amigo Yosvanis que en cuestiones de nada se convirtió en una de las personas que más quiero, por su linda amistad y su alegría.

A mis amigas Aislín, Milagros e Indira, que a pesar de estar 5 años estudiando lejos de Guantánamo, aún siguen siendo mis amigas incondicionales.

A todos los amigos de la vocacional que estamos en la UCI y que nos hemos apoyado siempre: en las recreaciones y en los viajes. En especial a Walter, Frank Geiler y Dannier.

Al pikete del pre que aunque todos hemos cogido caminos diferentes, viví con ellos 3 buenos años de mi vida.

Al pikete de primer año, como ese ninguno. Todos insoportables pero sabíamos cómo divertirnos y hacer las aburridas clases más amenas.

Al pikete del laboratorio que siempre me han ayudado con mis problemas con la tecnología en especial a Andry, Reisel, Osvel, Edel y Orlando.

A todos los profesores en especial a: Raimundo Bermúdez, Yuleidys Mejías y Yudelkis Abad.

A mi tutor Tonysé por creer siempre en mí, por darme ánimos, por defenderme sólo como él sabe hacerlo.

A las chicas del apartamento en especial a Clara y Adrialis que juntas hemos compartido fiestas y siempre me han brindado su ayuda.

A esa personita especial que convivió conmigo 5 meses y medio en este último año. Por soportarme tanto tiempo como nadie, apoyarme en todo y demostrarme que muchas cosas lindas son posibles.

A nuestra Revolución Socialista que hizo posible que tengamos el derecho a formarnos como profesionales y realizarnos como protagonistas de su obra.

A todos los que en un momento determinado me brindaron su apoyo desinteresado que aunque no los mencione, yo estoy muy clara de quienes son.

Muchas Gracias.

**DEDICATORIA**

A mis padres por toda su dedicación y amor, por la educación que me brindaron y el apoyo incondicional para llegar hasta aquí. Sin ellos nada sería posible.

A mis amigos que siempre han estado presentes y que ocupan un lugar importante en mi vida y en mi corazón.

**RESUMEN**

El silenciamiento genético por Ácidos Ribonucleicos de interferencia naturales (miRNA por sus siglas en inglés) constituye uno de los grandes retos de la Bioinformática y una poderosa técnica para el tratamiento de disímiles enfermedades. Con el fin de realizar estudios de esta índole y proporcionar una herramienta propia para el análisis de las secuencias genómicas surge la plataforma alasSiRNA-Design en la Universidad de las Ciencias Informáticas (UCI). Dicho sistema presenta nuevos servicios en su servidor web SiRNA Web Services que permiten analizar la influencia de las mutaciones puntuales tanto en miRNAs como en los genes. Estos servicios no han sido consumidos por la aplicación cliente que compone la plataforma alasSiRNA-Design. Con el propósito de solucionar las limitaciones del sistema existente el módulo de visualización de servicios permite la visualización de los resultados obtenidos de los servicios web. Con el presente trabajo de diploma se muestran los resultados del análisis, diseño, implementación y validación del módulo de visualización de servicios para la plataforma alasSiRNA-Design. Brindándole a los investigadores la posibilidad de usar estos servicios en la búsqueda de soluciones a determinadas enfermedades.

**Palabras clave:** alasSiRNA-Design, miRNAs, silenciamiento genético, visualización.



TABLA DE CONTENIDOS

AGRADECIMIENTOS ..... I

DEDICATORIA ..... III

RESUMEN ..... IV

INTRODUCCIÓN ..... 1

CAPÍTULO 1. FUNDAMENTO TEÓRICO ..... 5

    INTRODUCCIÓN ..... 5

    1.1 LA INFORMÁTICA Y SU RELACIÓN CON LAS CIENCIAS BIOLÓGICAS ..... 5

    1.2 VISUALIZACIÓN DE DATOS ..... 7

        1.2.1 Técnicas de visualización gráfica de datos ..... 8

        1.2.2 Visualización de datos biológicos ..... 9

    1.3 SILENCIAMIENTO GENÉTICO ..... 10

        1.3.1 Mutaciones puntuales ..... 11

        1.3.2 Polimorfismo de Nucleótido Simple ..... 11

    1.4 IMPORTANCIA DE LAS APLICACIONES Y SERVICIOS WEB EN LA BIOINFORMÁTICA ..... 12

    1.5 LIMITACIONES DEL SISTEMA EXISTENTE ..... 13

    1.6 TECNOLOGÍAS, METODOLOGÍAS Y HERRAMIENTAS ..... 14

        1.6.1 Tecnología ..... 14

        1.6.2 Lenguajes de desarrollo del software ..... 15

        1.6.3 Metodología de desarrollo de software ..... 17

        1.6.4 Herramientas de desarrollo ..... 18

        1.6.5 Gestor de Base de Datos ..... 18

        1.6.6 Servidor de Aplicaciones ..... 19

        1.6.7 Servidor Web ..... 19

CAPÍTULO 2. CARACTERÍSTICAS DEL SISTEMA ..... 21

    INTRODUCCIÓN ..... 21

    2.1 PROPUESTA DE SISTEMA ..... 21

2.2 MODELO DE DOMINIO .....	22
2.3 ESPECIFICACIÓN DE LOS REQUERIMIENTOS DE SOFTWARE .....	23
2.3.1 Requerimientos Funcionales.....	23
2.3.2 Requerimientos no funcionales .....	24
2.4 DEFINICIÓN DE ACTORES Y CASOS DE USO DEL SISTEMA .....	26
2.4.1 Actores del sistema.....	26
2.4.2 Casos de Uso definidos .....	26
2.4.3 Diagrama de casos de uso del sistema.....	27
2.4.4 Descripción detallada de casos de uso. ....	27
2.5 CONCLUSIONES.....	28
<b>CAPÍTULO 3. ANÁLISIS Y DISEÑO DEL SISTEMA .....</b>	<b>29</b>
INTRODUCCIÓN .....	29
3.1 PATRONES ARQUITECTÓNICOS.....	29
3.2 VISTA LÓGICA DEL SISTEMA .....	30
3.3 PATRONES DE DISEÑO .....	32
3.4 MODELO DE DISEÑO .....	34
3.5 DIAGRAMAS DE INTERACCIÓN. DIAGRAMAS DE SECUENCIA .....	36
3.6 TRATAMIENTO DE ERRORES .....	37
3.7 SEGURIDAD.....	38
3.8 CONCLUSIONES.....	38
<b>CAPÍTULO 4. IMPLEMENTACIÓN Y PRUEBA .....</b>	<b>39</b>
INTRODUCCIÓN .....	39
4.1 MODELO DE IMPLEMENTACIÓN .....	39
4.1.1 Diagrama de Despliegue.....	39
4.1.2 Diagrama de Componentes .....	40
4.2 INTERPRETACIÓN GRÁFICA DEL PICTOGRAMA PARA EL CU VISUALIZAR SNPs INFLUYENTES DE UN GEN... 49	49
4.3 MODELO DE PRUEBA .....	50
4.3.1 Caso de prueba: Visualizar SNPs influyentes de un gen.....	51
4.3.2 Resumen de las pruebas realizadas .....	52

4.4 CONCLUSIONES .....	54
CONCLUSIONES .....	55
RECOMENDACIONES .....	56
REFERENCIAS BIBLIOGRÁFICAS .....	57
BIBLIOGRAFÍA .....	62
GLOSARIO DE TÉRMINOS .....	67

ÍNDICE DE FIGURAS

Figura 1. Modelo de Dominio .....	23
Figura 2. Diagrama de caso de uso del sistema .....	27
Figura 3. Modelo-Vista-Controlador .....	30
Figura 4. Vista lógica del sistema.....	32
Figura 5. Diagrama de clases del diseño del CU Visualizar SNPs influyentes en un gen.....	35
Figura 6. Diagrama de secuencia del CU Visualizar SNPs influyentes en un gen .....	36
Figura 7. Pantalla de error. Campo vacío.....	37
Figura 8. Pantalla de error. El transcrito no existe en la base de datos. ....	38
Figura 9. Diagrama de Despliegue.....	40
Figura 10. Diagrama de componentes del CU Visualizar SNPs influyentes de un gen.....	41
Figura 11. Código de la página index.html .....	42
Figura 12. Código de la página visualizeInfluentialSnGene.html .....	42
Figura 13. Código de la página resources.js .....	43
Figura 14. Código de la clase ControllerServlet.java .....	44
Figura 15. Código de la clase RequestUtility.java .....	45
Figura 16. Código de la clase ActionFactory.java.....	46
Figura 17. Código de la clase GeneInfluentialSNP.java .....	46
Figura 18. Código de la clase Action.java .....	46
Figura 19. Código de la clase SNPInGeneModel.java.....	47
Figura 20. Código de la clase Transcript.java .....	48
Figura 21. Código de la página transGeneInfluentialSNP.jsp .....	48
Figura 22. Código de la clase SIRNAException.java .....	48
Figura 23. Pantalla final del CU Visualizar SNPs influyentes de un gen.....	50
Figura 24. Propiedades del SNP influyente.....	50

**ÍNDICE DE TABLAS**

Tabla 1. Actores del sistema .....	26
Tabla 2. Descripción del CU Visualizar SNPs influyentes de un gen.....	27
Tabla 3. Resultado del caso de prueba Visualizar SNPs influyentes en un gen .....	51
Tabla 4. No Conformidades detectadas .....	52

### INTRODUCCIÓN

*“La posibilidad de realizar un sueño es lo que hace que la vida sea interesante.”*

A mediados del siglo XX se llevaron a cabo relevantes estudios en el área de la Biología Molecular permitiendo que a mediados de los años ´90 se descifrara la primera secuencia genómica completa de un organismo vivo. Desde entonces hasta la actualidad ha sido creciente la cifra de nuevas secuencias de genomas tanto de humanos como de otros organismos vivos [1].

Las secuencias genómicas contienen la información necesaria para llevar a cabo la síntesis de proteínas y de moléculas de Ácido Ribonucleico (RNA por sus siglas en inglés), responsables de las características estructurales y funcionales de un organismo. Esta información la portan las cadenas de Ácido Desoxirribonucleico (DNA por sus siglas en inglés) contenidas en los cromosomas. En el DNA de los cromosomas se encuentran los genes, que son segmentos que contienen la información necesaria para la síntesis de otras moléculas. Mecanismos bioquímicos en el núcleo celular seleccionan y transcriben parte de la información en un gen formando una copia de su mensaje en moléculas de RNA mensajero (mRNA por sus siglas en inglés), proceso que se conoce como transcripción del DNA. Las moléculas de mRNA son utilizadas entonces para la síntesis de la proteína correspondiente en los ribosomas, proceso conocido como traducción [2].

En las etapas de transcripción y traducción el mensaje genético se transmite donde el RNA es un intermediario imprescindible. En ambas etapas existen mecanismos de regulación donde algunos pueden interferir a nivel del RNA el proceso de traducción a cadenas de aminoácidos. Uno de estos mecanismos es el mecanismo que regula la expresión genética conocido como silenciamiento genético, donde a partir de RNAs de cadena corta se inhibe el efecto de cadenas de mRNA complementarios. Más específicamente, los RNAs de cadena corta son interceptados para bloquear la traducción del mRNA. Se conocen de genes que codifican secuencias de RNA que activan el mecanismo de interferencia. Estos se conocen como RNA de interferencia natural (miRNA por sus siglas en inglés) [3].

En 1998 Andrew Fire<sup>1</sup> y Craig C. Mello<sup>2</sup> descubrieron que se podía activar el efecto de interferencia inyectando RNA de doble cadena (dsRNA por sus siglas en inglés) con algunas decenas de nucleótidos en la célula. A estos RNA de cadena corta se les denominó “RNAs pequeños de interferencia” o siRNA. Lo

---

<sup>1</sup> *Biólogo norteamericano, profesor de patología y de genética. Premio Nobel de Medicina en el 2006 junto a Craig Mello por el descubrimiento de RNA de interferencia (RNAi).*

<sup>2</sup> *Biólogo norteamericano. Profesor Distinguido de Medicina Molecular de la Universidad de Massachusetts Medical School e investigador en el Instituto Médico Howard Hughes desde el 2000.*

importante de este descubrimiento es que se podían utilizar secuencias complementarias a genes legítimos en la célula, y que entonces el mecanismo de interferencia no sólo bloquearía la expresión de los dsRNA inyectados, sino también la expresión de los genes del organismo seleccionados por los investigadores [4].

El mecanismo de silenciamiento genético a través de RNA, puede activarse por dos vías: cuando se expresa un miRNA o por la aparición de dsRNA. Donde esta técnica de silenciamiento genético mediante los siRNAs se perfila como una poderosa herramienta terapéutica para combatir y buscar soluciones a disímiles enfermedades [5].

Con el fin de crear una herramienta propia para el diseño de siRNAs y realizar estudios sobre el silenciamiento genético; el grupo de Bioinformática BioSoft de la facultad 6 de la Universidad de las Ciencias Informáticas<sup>3</sup> (UCI) en conjunto con el Centro de Ingeniería Genética y Biotecnología<sup>4</sup> (CIGB) desarrolló la plataforma alasSiRNA-Design.

Los miRNAs a diferencia de los siRNAs están codificados en el genoma de la célula y coordinan naturalmente la expresión temporal de muchos genes. Por lo que se hace imprescindible para el diseño un siRNA conocer todo el proceso que llevan a cabo naturalmente los miRNAs.

Por este motivo en el servidor de servicios web de la plataforma SiRNA Web Services, se implementaron nuevos servicios que determinan computacionalmente la influencia de las mutaciones puntuales en la acción de los miRNAs y los genes que estos silencian [6].

Dichos servicios resultan de gran importancia para los investigadores del CIGB en la búsqueda de soluciones a determinadas enfermedades, por ejemplo enfermedades neurodegenerativas humanas como el Parkinson y el Mal de Alzheimer, pero no han sido consumidos por la aplicación cliente que compone el sistema. Por tanto los investigadores no pueden percibir los resultados de los servicios web implementados. Además no tienen acceso al fichero en formato FASTA con la información de los genes de todo el genoma humano. Dicho fichero es usado por otras aplicaciones y su almacenamiento en la PC del especialista no está automatizado.

Por tanto se identifica como **problema a resolver** ¿Cómo visualizar los datos que brindan los servicios implementados del lado del servidor en la plataforma alasSiRNA-Design v1.0?

---

<sup>3</sup> Centro de estudio creado en el fragor de las Batalla de Ideas. Universidad creada para formar profesionales en el campo de la informática con el objetivo de informatizar el país y desarrollar la industria del software.

<sup>4</sup> Institución del Polo Científico del Oeste de la Habana. Centro de excelencia a nivel mundial en el campo de la biotecnología y la ingeniería genética por sus notables resultados.

El **objeto de estudio** de este trabajo está enmarcado en las aplicaciones informáticas para la visualización de servicios web teniendo como **campo de acción** las aplicaciones informáticas que visualizan datos biológicos a partir de servicios web.

Para dar solución al problema planteado se trazó como **objetivo general** de la investigación: Implementar la visualización para el módulo de visualización de servicios para la plataforma alasSiRNA-Design v1.0.

Para dar cumplimiento al objetivo general del trabajo se trazaron los siguientes **objetivos específicos**:

1. Fundamentar los conceptos asociados a silenciamiento genético y visualización de datos biológicos.
2. Realizar el análisis y diseño del módulo de visualización de servicios para la plataforma alasSiRNA-Design v1.0.
3. Implementar el módulo de visualización de servicios para la plataforma alasSiRNA-Design v1.0
4. Validar el módulo de visualización de servicios implementados en la plataforma alasSiRNA-Design v1.0.

Para cumplir con dichos objetivos se trazaron las siguientes **tareas**:

- ✓ Revisión bibliográfica sobre los siguientes temas:
  - Proceso de silenciamiento de genes en el genoma humano.
  - Visualización de datos biológicos.
- ✓ Elaboración del análisis y el diseño del módulo de servicio para la plataforma alasSiRNA-Design v1.0.
- ✓ Implementación de la visualización del servicio buscar información de un miRNA.
- ✓ Implementación de la visualización del servicio crear archivo Fasta.
- ✓ Implementación de la visualización del servicio buscar SNPs influyentes de un gen.
- ✓ Implementación de la visualización del servicio buscar SNPs influyentes de un miRNA.
- ✓ Aplicación de las pruebas al módulo de visualización de los servicios implementados para la plataforma alasSiRNA-Design v1.0

El documento estará estructurado de la siguiente forma:

**Capítulo 1: Fundamento Teórico:** Constituye el respaldo teórico de los temas tratados en el informe, necesarios para el entendimiento correcto de la solución planteada. Se realiza un análisis de la relación de la informática con las ciencias biológicas y se describen los principales conceptos que fundamentan el contenido de este trabajo. Además se aborda la importancia del desarrollo del módulo a partir de las



limitaciones del sistema actual, describiendo las tecnologías, metodologías y herramientas utilizadas, así como la justificación de su uso.

**Capítulo 2: Características del sistema:** Se realiza una propuesta del módulo definiéndose los requerimientos funcionales y no funcionales del mismo. Se describen los actores y casos de uso del sistema donde para un mayor entendimiento de las aplicaciones se modelan en un diagrama.

**Capítulo 3: Análisis y diseño del sistema:** Se modelan los artefactos del análisis y el diseño del módulo. Se realiza una descripción de la arquitectura y se identifican los patrones de diseño utilizados. Se describe el tratamiento de errores, el sistema de seguridad y se presenta la vista lógica del módulo.

**Capítulo 4: Implementación y prueba:** Se realiza el modelo de implementación a partir de los resultados del flujo de trabajo anteriormente descrito: Diseño. Se describen las pruebas que se le han realizado al módulo.

## CAPÍTULO 1

### FUNDAMENTO TEÓRICO

“Aprender mientras hago y no mientras pienso en hacer algo...”

#### Introducción

En el presente capítulo se realiza un estudio de la visualización de datos y su importancia en la Bioinformática. Por otro lado se fundamenta la importancia de las aplicaciones y servicios web en la Bioinformática. Además se aborda la importancia del desarrollo del módulo a partir de las limitaciones del sistema actual y se fundamentan las tendencias y tecnologías actuales utilizadas en la plataforma, así como las herramientas usadas.

#### 1.1 La informática y su relación con las ciencias biológicas

La informática hoy en día constituye una pieza indispensable en todos los sucesos y ámbitos laborales existentes. Ofreciendo el poder del conocimiento y las comunicaciones con mayor rapidez. Sus aplicaciones son tan numerosas que resulta inconcebible que exista un campo o área donde no esté presente.

En la década de los 50 del pasado siglo XX los científicos Francis Crick<sup>5</sup> y James Watson<sup>6</sup> proponen la estructura de doble hélice del Ácido Desoxirribonucleico (DNA por sus siglas en inglés) y fue secuenciada la primera proteína (insulina bovina) por Frederick Sanger<sup>7</sup>. Estos descubrimientos dieron paso a una nueva época en el desarrollo de la Biología Molecular. Pero no fue hasta mediados de los años 90 que se descifra la primera secuencia genómica completa de un organismo vivo, la bacteria *Haemophilus Influenzae* por Craig Venter<sup>8</sup>. A partir de este suceso hasta la actualidad ha sido creciente la cifra de nuevas secuencias de genomas generando un volumen enorme de información incapaz de ser

---

<sup>5</sup> Físico y biólogo molecular británico. Premio Nobel de Medicina en 1962, junto a James Watson y Maurice Wilkins, por el descubrimiento de la estructura del DNA. Merecedor de las medallas Royal y Copley y de la Orden del Mérito.

<sup>6</sup> Bioquímico y genetista estadounidense. Premio Nobel de Fisiología y Medicina de 1962. Dirigió el Proyecto Genoma Humano. Se le considera uno de los padres de la biología molecular.

<sup>7</sup> Bioquímico británico considerado como uno de los más relevantes de la historia de la ciencia. Obtuvo dos veces el premio Nobel de Química.

<sup>8</sup> Biólogo molecular estadounidense. Premio Nobel de Fisiología y Medicina 1978. en 1999 arrancó su propio Proyecto Genoma Humano. Actualmente es el presidente del Instituto J. Craig Venter.

almacenada y analizada sin un sólido apoyo de las tecnologías de la información. Esta interacción indisoluble de ambas ciencias se conoce con el nombre de Bioinformática [1].

La Bioinformática se encuentra en la intersección entre las ciencias de la vida y de la información, proporciona las herramientas y recursos necesarios para favorecer la investigación biomédica. Como campo interdisciplinario, comprende la investigación y el desarrollo de sistemas útiles para entender el flujo de información desde los genes a las estructuras moleculares, su función bioquímica, su conducta biológica y, finalmente, su influencia en las enfermedades y en la salud.

El enorme volumen de datos generados por los distintos proyectos denominados genoma, los nuevos enfoques experimentales, basados en biochips, que permiten obtener datos genéticos a gran velocidad, bien de genomas individuales (mutaciones, polimorfismos) como de enfoques celulares (expresión génica) y el desarrollo de Internet, que permite el acceso universal a las bases de datos de información biológica, constituyen los pilares que propician el desarrollo de la Bioinformática en la actualidad [7].

Esta disciplina científica se convirtió en el detonador de ciencias que revolucionaron a gran escala como la Biología de Sistemas, la cual representa una estrategia analítica para relacionar los elementos de un sistema. Esta ciencia tiene como objetivo comprender las interacciones entre los componentes de los sistemas vivos, y por consiguiente, de sus procesos biológicos. Este análisis de sistemas puede aplicarse a moléculas, células, órganos, individuos o incluso ecosistemas [8].

El auge de la Biología de Sistemas con su uso intensivo de la Bioinformática ha hecho que estos dos campos se mezclen sin que hoy en día haya una separación clara entre los dos [9]. La Bioinformática evolucionó desde un conjunto de técnicas hacia una verdadera ciencia, al contribuir con el análisis para entender las secuencias genómicas e integrar sus datos creando modelos predictivos para los sistemas biológicos mediante la investigación y el desarrollo de sistemas. El objetivo primordial de esta ciencia es acelerar los descubrimientos científicos, reduciendo costos y aumentando la cantidad de experimentos [10]. En la actualidad la mayoría de los dilemas más importantes de la biología moderna y la genómica son imposibles de resolver sin el poder del cálculo de las computadoras [11]. Los estudios referentes a la identificación de genes, la predicción de estructuras y función y el silenciamiento genético se realizan con mayor rapidez y eficiencia gracias a la Informática y sus herramientas. Las herramientas informáticas son imprescindibles para la investigación de distintas enfermedades y el diagnóstico clínico, estableciéndose así el camino a seguir por biólogos e investigadores durante el siglo XXI, nombrado por los especialistas como "el siglo de las tecnologías de la información" [10].

La magnitud de la información que generan las investigaciones realizadas sobre el genoma humano es tal que, probablemente, supera la generada por otras investigaciones en otras disciplinas científicas. Por lo que uno de los retos de la Bioinformática es el desarrollo de métodos que permitan integrar los datos genómicos –de secuencia, de expresión, de estructura, de interacciones, entre otras – para explicar el comportamiento global de la célula viva, minimizando la intervención humana. Por lo que se hace necesaria la automatización de los datos y la creación de herramientas de análisis y visualización [7].

### 1.2 Visualización de datos

La sociedad de la información vive inundada por grandes volúmenes de datos resultantes de los procesos científicos y sociales de difícil digestión e interpretación. Mediante técnicas de visualización es posible mejorar ampliamente la comprensión y el análisis de la ingente cantidad de datos que abunda hoy en día en cualquier actividad; descubriendo patrones que hacen que la comunicación del conocimiento ya existente sea más sencilla y eficiente [12].

La visualización como disciplina novedosa utiliza el inmenso poder de comunicación de las imágenes para explicar de manera comprensible las relaciones de significado, causa y dependencia que se pueden encontrar entre las grandes masas abstractas de datos que generan los procesos científicos y sociales [13].

La visualización ha encontrado en la Informática un apoyo trascendental aunque su práctica es muy antigua. Sus orígenes más remotos se evidencian desde el surgimiento de la humanidad, cuando eran usadas para darle respuestas a fenómenos naturales. Aunque no fue hasta el siglo XVIII donde se encuentran ejemplos de representación gráfica de datos cuantitativos, pero el uso del medio informático ha hecho que estas representaciones pasen de ser la excepción a convertirse en la norma. Asimismo, ha posibilitado toda una serie de nuevas técnicas de visualización y de usos para éstas. Por medio del ordenador podemos visualizar conjuntos de datos más grandes. Además permite crear visualizaciones dinámicas, introducir datos en tiempo real, regresar a los datos que dieron lugar a la visualización y basar la representación gráfica de los datos en su análisis matemático, utilizando toda una variedad de métodos, desde la estadística clásica a la minería de datos [14].

Aunque la información en su mayoría se representa en grandes volúmenes de textos sin dudas las tablas, gráficos, mapas, pictografías o los diagramas comúnmente usados en la historia de la ciencia y las matemáticas, son los medios más eficientes en la visualización de datos. Estos permiten determinar respuestas sobre fenómenos y adquirir conocimientos que de otra manera tal vez no fueran muy fáciles de comprender [15]. La Bioinformática no se queda atrás en la utilización de estos medios para visualizar los

datos que generan las investigaciones realizadas, por lo que brinda principal atención en la búsqueda de métodos eficientes de acuerdo a lo que se desea mostrar.

### 1.2.1 Técnicas de visualización gráfica de datos

La visualización gráfica de datos proporciona mayor rapidez y confianza en las conclusiones obtenidas. Por lo que a lo largo de los últimos años se han desarrollado diversas técnicas de representación gráfica. Estas técnicas en ocasiones son identificadas por el tipo de datos que muestran, qué representación visual utilizan o en qué áreas de aplicación son usados [16]. Entre las principales técnicas de visualización se encuentran:

**Pictogramas:** También llamada gráfica de imágenes o pictografía. Es un diagrama que utiliza imágenes o símbolos para mostrar datos para una rápida comprensión. La mayor frecuencia se identifica por la mayor acumulación de símbolos y es más funcional y natural que cualquier otro sistema [17].

Dentro de sus ventajas se destacan algunas como:

- Su formato es libre (el usuario puede construir su propia imagen basado en lo que desee).
- Emplean una secuencia de símbolos para representar frecuencias.
- Se emplean para el tratamiento de datos tanto cualitativos como cuantitativos.

**Mapas:** Los mapas se utilizan para representar una región de la Tierra en un plano. En ellos podemos representar diferentes características, como el clima, la flora, la población, etc. Los tipos de mapas más comunes son los políticos, los físicos, los topográficos y los conceptuales [18].

**Histogramas:** Es un resumen gráfico de los valores producidos por las variaciones de una determinada característica, representando la frecuencia con que se presentan distintas categorías dentro de dicho conjunto [18].

**Redes neuronales artificiales:** Estos modelos son dirigidos a partir de los datos, es decir, son capaces de encontrar relaciones o patrones de forma inductiva por medio de los algoritmos de aprendizaje basado en los datos existentes. Además requiere la ayuda de un modelador para especificar la forma funcional y sus interacciones. En las últimas décadas las Redes Neuronales Artificiales han recibido un interés particular como una tecnología para minería de datos, puesto que ofrece los medios para modelar de manera efectiva y eficiente problemas grandes y complejos. En el campo de Biología estos modelos son muy usados para el estudio de sistemas [19].

**Gráficos de barra:** Representan valores usando trazos verticales, aislados o no unos de otros, según la variable a graficar, si es discreta o continua. Este tipo de gráfico es útil para mostrar cambios de datos en

un período de tiempo o para ilustrar comparaciones entre elementos. Estos pueden clasificarse como: gráficos de barras verticales, horizontales, proporcionales, comparativas o apiladas [20].

**Gráficos circulares:** Estos gráficos también conocidos como, gráfico de Tartas o Sectores muestra los datos como un círculo dividido en secciones de colores o diseños, se divide un círculo en tantas porciones como clases tenga la variable, de modo que a cada clase le corresponde un arco de círculo proporcional a su frecuencia absoluta o relativa [20].

**Gráficos de líneas:** Este tipo de gráfico es una serie de puntos conectados entre sí mediante rectas, donde cada punto puede representar distintos valores, los cuales son representados en dos ejes cartesianos ortogonales entre sí. Estos pueden clasificarse como: gráfico de líneas apiladas, líneas apiladas con marcadores y líneas en 3D [20].

**Gráficos de áreas:** Un gráfico de área muestra sus datos como áreas llenas de colores o diseños y destacan la magnitud del cambio en el tiempo utilizándose para llamar la atención hacia el valor total en una tendencia. Dentro de las clasificaciones de los gráficos de áreas existentes están: gráfico de áreas en 2D, en 3D y apiladas [21].

**Gráficos de burbujas:** El gráfico de Burbujas presenta los datos como una serie de burbujas, donde el tamaño de las burbujas está en proporción a la cantidad de datos. Al igual que un gráfico de dispersión, un gráfico de burbujas no usa un eje de categorías, tanto el eje horizontal como el vertical son ejes de valores. Además de los valores x e y que se dibujan en un gráfico de dispersión, un gráfico de burbujas representa los valores z (el tamaño) [21].

### 1.2.2 Visualización de datos biológicos

En las últimas décadas, el uso de ordenadores para la visualización constituye la base del nuevo campo de la visualización científica donde la medicina moderna depende de la visualización del cuerpo y su funcionamiento y la biología moderna de la visualización del DNA y las proteínas [22].

La visualización de datos biológicos mediante herramientas informáticas y bases de datos específicas constituye una de las áreas de la Bioinformática. Donde se perfecciona en brindar una buena herramienta de visualización y una estandarización a la hora de representar los datos. Es la única manera de que se pueda tener un acceso sencillo a todos los datos y desde cualquier laboratorio del mundo [22].

El proceso de silenciamiento genético como todo proceso estudiado en el campo de la Bioinformática necesita de nuevas y poderosas herramientas de análisis y visualización. La visualización de datos

constituye el eslabón fundamental que le da valor a la información obtenida y constituye la ayuda principal al análisis de las secuencias genómicas como herramienta de comunicación avanzada.

### 1.3 Silenciamiento genético

El silenciamiento genético es el mecanismo que regula la expresión genética, mediante el cual el RNA regula la expresión de ciertas regiones del Ácido Desoxirribonucleico (DNA por sus siglas en inglés) y define qué proteínas están presentes en la célula [23]. Este mecanismo puede actuar antes o después de la transcripción (silenciamiento genético transcripcional (TGS) y postranscripcional (PTGS), respectivamente) [24].

Desde hace 11 años se dispone de una nueva herramienta molecular, que ha revolucionado al mundo científico, la interferencia por RNA (RNAi), con la cual se puede reducir (knock-down) o perder (knock-out) la expresión de un gen en células de mamíferos (in vitro) e inclusive en animales (in vivo). Estas moléculas de RNA de interferencia pueden ser microRNA (miRNA) o dsRNA según su procedencia. El dsRNA puede proceder de virus, transposones (y el sistema actúa como defensa) o por inyecciones artificiales realizadas por investigadores. Los dsRNAs inyectado artificialmente originan los siRNAs. Los siRNAs son moléculas de RNA de doble hebra de 20-21 nucleótidos (nt) que se combinan con proteínas para incorporarse al Complejo de Silenciamiento Inducido (RISC por sus siglas en inglés). Estos tienen secuencias perfectamente complementarias a las secuencias de mRNA, suprimiendo su expresión genética por degradación o inhibiendo la transcripción. Estos por lo general silencian genes a partir de los cuales se han transcritos y actualmente se está probando a los siRNA como posible tratamiento para algunas enfermedades virales y genéticas, así como el cáncer. A diferencia de los siRNAs, los miRNAs están codificados en el genoma de la célula [9][2][10].

Los miRNAs son moléculas pequeñas de RNA, alrededor de 21 nt, no codificadoras que se unen a las moléculas de mRNA complementarias e inhiben su traducción. Constituyen interruptores maestros que coordinan la expresión temporal de muchos genes, que a su vez, pueden controlar otros grupos de genes. Esta compleja coordinación espacio-temporal permite el desarrollo del sistema nervioso central, del aparato digestivo, y de las extremidades inferiores o superiores, en animales [11].

Los miRNAs habitualmente silencian genes que difieren de aquellos a partir de los cuales se han transcrito y una de las cadenas del miRNA es incorporada al RISC mientras que la otra es liberada y degradada. Estos se han encontrado en todos los organismos eucariotas analizados hasta la fecha, así también como en virus; ellos controlan la expresión de genes involucrados en varios procesos biológicos, incluidos el

crecimiento, el desarrollo y el metabolismo. Los humanos poseen más de 450 miRNAs distintos; los científicos estiman que más de un tercio de todos los genes humanos son regulados por miRNAs [9].

El silenciamiento genético es un proceso sumamente importante en la mayoría de los organismos. Si una célula expresara todos y cada uno de sus genes sin una cierta regulación, el desorden generado en el interior de la misma podría conducir inclusive al frenado total de las funciones metabólicas, lo que conllevaría a una muerte celular inevitable [25].

El proceso de silenciamiento genético puede ser alterado si en la secuencia del miRNA o en la del gen ocurre una mutación. Esta mutación puede provocar disminución o aumento de los genes que son silenciados. Por tanto resulta de vital importancia determinar la influencia de las mutaciones puntuales en la acción de los miRNAs y en los genes que estos silencian.

### **1.3.1 Mutaciones puntuales**

Las mutaciones puntuales cubren una variedad de mutaciones desde cambios genuinos en un par de bases hasta pequeñas deleciones e inserciones. Un cambio en un solo par de bases puede no tener consecuencias genéticas drásticas, pero por otro lado, puede causar un efecto dominante letal. La primera alternativa es por lejos más probable porque el 95 % del DNA es no codificante y un cambio en una sola base que ocurra en este DNA es improbable que tenga algún efecto. Además, debido a que el código genético es degenerado, las mutaciones que ocurran en la posición de la tercera base en un codón no tendrán consecuencias respecto a los aminoácidos codificados [26].

El cambio de una sola base en la secuencia genómica es conocida como polimorfismo de nucleótido simple.

### **1.3.2 Polimorfismo de Nucleótido Simple**

Un Polimorfismo de Nucleótido Simple (SNP por sus siglas en inglés) es una variación en la secuencia de DNA que afecta a una sola base de una secuencia del genoma. Estos constituyen la forma más sencilla y a la vez más frecuente, de variabilidad genética entre individuos de una misma especie. Por su alta frecuencia en el genoma humano uno por cada 1000 pares de bases y por no estar relacionados a fenotipos de enfermedades, los SNP son muy útiles para determinar la susceptibilidad a padecer una enfermedad o para determinar la sensibilidad a agentes terapéuticos. Además ellos representan la referencia genómica por excelencia que los investigadores emplean para marcar la localización de genes



responsables o involucrados en los caracteres heredables y las enfermedades, como el color de los ojos, la piel, el cabello y la forma más común de albinismo, por citar algunos ejemplos [27][28].

Los SNP pueden ser un punto muy importante a tener en cuenta en el diseño de un siRNA. Si se diseña un siRNA y en esa secuencia de 21 nt existen uno o varios SNP, en ese caso si desea generalizar cierto fármaco que dependa de ese siRNA puede correr el riesgo de que para unos individuos solucione el problema y para otros no. También podría servir como base para la llamada medicina personalizada, teniendo en cuenta su código genético se diseñaría un fármaco específico [2].

Los SNPs se consideran una forma de mutación puntual que ha sido lo suficientemente exitosa evolutivamente para fijarse en una parte significativa de la población de una especie. Por lo tanto cuando se hable de mutación puntual se estará haciendo referencia a un SNP [6].

### 1.4 Importancia de las aplicaciones y servicios web en la Bioinformática

Hoy en día la diversidad, heterogeneidad y dispersión de los datos biológicos en el mundo es posiblemente la característica que mejor define la Bioinformática. Por lo que buena parte de esta disciplina se basa en aplicaciones que se ejecutan en la web haciendo uso de las bases de datos biológicas disponibles en Internet. Estos datos son usados para obtener similitud entre secuencias o reglas que permitan predecir propiedades en secuencias no conocidas.

Una de las aplicaciones basadas en web más utilizadas en Bioinformática son las relacionadas precisamente con el procesamiento de secuencias de DNA y proteínas. Entre ellas se nombran sistemas como: BLAST, Fasta, MiRscan, TargetScan, entre otras [6][29].

Las aplicaciones web son sistemas informáticos basados en tecnología y estándares del Consorcio para la web (W3C<sup>9</sup> por sus siglas en inglés) e independientes de las tecnologías de hardware y software que se utilicen. Estas permiten al usuario acceder a través de una interfaz interactiva a los recursos y servicios ofrecidos por un servidor web [30].

La tecnología de servicios web se está convirtiendo en la opción de elección para desplegar herramientas de Bioinformática que están universalmente disponibles. Una de las mayores fortalezas de este enfoque es la interoperabilidad que se consigue mediante la adopción de estándares abiertos y el intercambio de información entre aplicaciones desarrolladas en lenguajes de programación diferentes, y ejecutadas sobre cualquier plataforma [31].

---

<sup>9</sup> Consorcio para la web (W3C del inglés *World Wide Web Consortium*). Consorcio internacional de compañías y organizaciones involucradas en el desarrollo de Internet y en especial de la WWW. Su propósito es desarrollar estándares y “poner orden” en Internet.

A pesar de la popularidad de los servicios web, estos en su mayoría no ofrecen una interfaz interactiva y le corresponde al software cliente solucionar esto, por lo que las aplicaciones clientes han llegado a gobernar tales cosas como la normalización de entrada de la invocación, y la visualización de salida. Por lo que es un aspecto importante en el desarrollo de estas aplicaciones tener en cuenta una concepción de interfaz de usuario intuitiva, que facilite la comunicación entre persona y ordenador.

### **1.5 Limitaciones del sistema existente**

La plataforma alasSiRNA-Design permite el diseño de siRNAs y la búsqueda, representación y visualización de datos biológicos para la toma de decisiones. Además ofrece a los especialistas cubanos una herramienta propia donde pueden incorporar nuevas ideas y algoritmos como la secuencia o los identificadores de los genes que se desean silenciar. Esta plataforma está compuesta por dos aplicaciones informáticas: el servidor SiRNA Web Services y el cliente alasSiRNA-Design. SiRNA Web Services es un servidor web que brinda una serie de servicios web relacionadas con la información que se necesita de la base de datos y la implementación de los principales algoritmos. El cliente web alasSiRNA-Design utiliza los servicios del servidor para consultar y mostrar las peticiones de los usuarios.

Con el objetivo de mejorar la calidad de las investigaciones aplicadas y el desarrollo de los trabajos implementados en el CIGB, se desarrollaron nuevos algoritmos en SiRNA Web Services. Dichos servicios no han sido consumidos por la aplicación cliente del sistema alasSiRNA-Design. Por lo que se hace necesaria la visualización de los datos que generan estos servicios y el almacenamiento de un archivo FASTA del lado del cliente, permitiendo el análisis de la influencia de las mutaciones puntuales en los miRNAs. Cuando ocurre una mutación (tanto en un miRNA como en un gen blanco) se provocan cambios en el silenciamiento (se incrementan o disminuyen los genes que son silenciados), por lo que los SNPs influyen en que un miRNA pueda o no silenciar determinados genes. En general, estos servicios permiten conocer dónde se encuentra un SNP y si este influye o no en el silenciamiento, brindando un mayor espectro en las investigaciones relacionadas sobre silenciamiento genético.

Para la implementación de los algoritmos en SiRNA Web Services se analizaron varias herramientas publicadas en internet dedicadas a la predicción de miRNAs entre ellas se encuentran: miRanda, miRanda on-line, PicTar, MiRAlign, miRBase, TargetScan, MiRscan. Excepto el miRanda que es una aplicación de escritorio, las herramientas anteriormente presentadas, tienen su uso básico sobre Internet. Estas usualmente suelen sufrir cambios repentinos e ignorados por los usuarios. La mayor desventaja que

tienen estos sistemas analizados es que en general sólo se pueden hacer búsquedas de los genes dianas de miRNAs, ninguno hace un análisis de los SNPs ni en los miRNAs ni en los genes analizados [6].

Después de realizar un estudio de las técnicas de visualización gráfica de datos y de los SNPs se concluye que la mejor forma de representar gráficamente los SNPs en la secuencia del miRNA o del gen es utilizando un pictograma. Los pictogramas permiten diseñar la mejor forma de representar un SNP y muestra la frecuencia de ocurrencia del mismo en la secuencia del transcrito.

### 1.6 Tecnologías, metodologías y herramientas

Para la implementación de las nuevas funcionalidades a la aplicación cliente de la plataforma alasSiRNA-Design se realizó un estudio de los lenguajes, metodología y herramientas utilizadas en la implementación de la plataforma. Se tuvo en cuenta además el uso de estándares y tecnologías que utiliza el CIGB que promueven prácticas orientadas hacia la funcionalidad y la entrega del producto, así como de la implicación directa del cliente.

#### 1.6.1 Tecnología

##### Java Server Page (JSP)

JSP es una tecnología Java que permite a los programadores generar contenido dinámico para la web, en forma de documentos HTML, XML, o de otro tipo. JSP permite al código Java y a algunas acciones predefinidas ser incrustadas en el contenido estático del documento. También es un marco de trabajo de programación en Java que puede utilizarse en un servidor para generar eficaces aplicaciones web, permitiendo un rápido desarrollo de las mismas. Debido a que la tecnología JSP es abierta y multiplataforma, los servidores web, plataformas y otros componentes pueden ser fácilmente actualizados o cambiados sin que afecte a las aplicaciones basadas en esta tecnología [32].

La tecnología JSP es una extensión de la tecnología servlets, los cuales son aplicaciones 100% Java que corren en el servidor: Se crea e inicia un servlet, se procesan las peticiones recibidas y por último se destruye. Este diseño explica por qué un servlet reemplaza perfectamente a una interfaz de entrada común (CGI<sup>10</sup> por sus siglas en inglés), ya que el servlet se carga una sola vez y está residente en memoria mientras se procesan las peticiones recibidas y se generan las respuestas a los usuarios. Cada vez que un cliente solicita al servidor web una página JSP, este pasa la petición al motor de JSP el cual verifica si

---

<sup>10</sup> *Importante tecnología de la World Wide Web que permite a un cliente (navegador web) solicitar datos de un programa ejecutado en un servidor web.*

la página no se ha ejecutado antes o fue modificada después de la última compilación. Luego de la compilación donde el motor de JSP convirtió la página en Servlet, ejecuta la página y devuelve los resultados al cliente en formato HTML [33].

La especificación JSP es el producto de una colaboración amplia de varias de las industrias líderes en el desarrollo de software, liderados por Sun Microsystems<sup>11</sup> (Sun). Lo importante fue que Sun hizo la especificación de JSP disponible libremente para la comunidad de desarrollo de software, con la idea de que todos los servidores web soporten JSP, compartiendo la característica de la tecnología Java "Write Once, Run Anywhere" (Escríbelo una vez, córralo donde quieras). Es conveniente resaltar, que la tecnología JSP es un componente clave de la plataforma J2EE<sup>12</sup> propuesta por Sun Microsystems. En resumen, las tecnologías JSP y Servlets son una alternativa importante para la programación web de contenido dinámico que permiten:

- Independencia de la plataforma.
- Rendimiento mejorado.
- Separación de la lógica de la aplicación de la presentación de los datos.
- Uso de componentes (Java Beans).
- Facilidad de administración y uso.
- El respaldo importante de la tecnología sólida Java TM [33].

### 1.6.2 Lenguajes de desarrollo del software

Para la implementación de una aplicación cliente que consume los servicios del lado del servidor se podrían seleccionar numerosos lenguajes de programación usados en el mundo informático como HTML, VBScript, Java, JavaScript los cuales son comparados desde el punto de vista del desarrollador de una aplicación, en cuanto a la seguridad, a la velocidad y a la robustez entre otros factores.

#### Lenguaje Java

Java ofrece todas las ventajas de un lenguaje potente y robusto, pues fue diseñado para crear software altamente fiable. Es un lenguaje de programación basado en clases y orientado a objetos. Sus características de memoria liberan a los programadores de responsabilidades y errores. Java es

---

<sup>11</sup> Exitosa empresa informática que se dedicaba a vender ordenadores, componentes informáticos, software y servicios informáticos. Comprada por Oracle en enero del 2010.

<sup>12</sup> Java Enterprise Edition es una plataforma de programación para desarrollar y ejecutar software de aplicaciones en Lenguaje de programación Java.

compilado en un código intermedio más abstracto que el código de máquina que es ejecutado por la máquina virtual de Java [34].

El lenguaje en sí mismo toma mucha de su sintaxis de C y C++, pero tiene un modelo de objetos más simple y elimina herramientas de bajo nivel, que suelen inducir a muchos errores, como la manipulación directa de punteros o memoria. Una de las principales características que favoreció el crecimiento y difusión del lenguaje Java es su capacidad de que el código funcione sobre cualquier plataforma de software y hardware. Java es toda una tecnología orientada al desarrollo de software con el cual se puede realizar cualquier tipo de programa. Hoy en día, la tecnología Java ha cobrado auge en el ámbito de Internet gracias a su plataforma J2EE [6].

### **Lenguaje de Descripción de Servicios Web**

Lenguaje de Descripción de Servicios Web (WSDL por sus siglas inglés) describe la interfaz abstracta a través de la cual un cliente puede acceder al servicio y los detalles de cómo se debe utilizar. Este lenguaje está basado en XML y describe la forma de comunicación, es decir, los requisitos del protocolo y los formatos de los mensajes necesarios para interactuar con los servicios listados en su catálogo. Las operaciones y mensajes que soporta se describen en abstracto y se ligan después al protocolo concreto de red y al formato del mensaje. O sea, un programa cliente que se conecta a un servicio web puede leer el WSDL para determinar qué funciones están disponibles en el servidor, este cliente puede usar SOAP para hacer la llamada a una de las funciones listadas en el WSDL [35].

### **Lenguaje JavaScript**

JavaScript es un lenguaje de programación utilizado para crear pequeños programitas encargados de realizar acciones dentro del ámbito de una página web. Se trata de un lenguaje de programación del lado del cliente, porque es el navegador el que soporta la carga de procesamiento. Su uso se basa fundamentalmente en la creación de efectos especiales en las páginas y la definición de interactividades con el usuario [36].

JavaScript es interpretado, no requiere compilación. Utilizado principalmente en páginas web. Es similar a Java, aunque no es un lenguaje orientado a objetos, el mismo no dispone de herencias. La mayoría de los navegadores en sus últimas versiones interpretan código JavaScript [37].

### **Lenguaje de Marcas Hipertextuales**

El Lenguaje de Marcas Hipertextuales (HTML por sus siglas en inglés) es un lenguaje estático que ha permitido la publicación de sitios web desde el surgimiento de Internet. Desarrollado por W3C. Los archivos pueden tener las extensiones .htm y .html. Es un lenguaje sencillo y fácil de aprender que permite

describir hipertexto, y representa el texto de forma estructurada y agradable. Su despliegue es rápido y admitido por todos los exploradores aunque la interpretación puede ser diferente [37].

### **Lenguaje Unificado de Modelado**

El Lenguaje de Unificado de Modelado (UML por sus siglas en inglés) es un lenguaje que permite modelar, construir y documentar los elementos que forman un sistema software orientado a objetos. Hoy en día, UML (del inglés Unified Modeling Language) está consolidado como el lenguaje estándar en el análisis y diseño de sistemas de cómputo en la programación orientada a objetos. En el proyecto se utiliza porque plasma un sistema de software previo al proceso de escribir el código [6].

### **1.6.3 Metodología de desarrollo de software**

Las metodologías de desarrollo de software son un conjunto de procedimientos, técnicas y ayudas a la documentación para el desarrollo de software. Indican paso a paso todas las actividades a realizar para lograr el producto informático deseado, indicando además qué personas deben participar en el desarrollo de las actividades y qué papel deben de tener. Además detallan la información que se debe producir como resultado de una actividad y la información necesaria para comenzarla [38].

Es importante escoger una buena metodología si se pretende reducir costos y retrasos de proyectos así como mejorar la calidad del software, pues estas cobran gran importancia en proyectos empresariales ya que al no utilizarla adecuadamente se puede desembocar en la frustración del equipo de desarrollo y en la insatisfacción de los clientes. Por tanto el uso de una metodología es necesario para controlar el ciclo de vida de un proyecto [39].

### **OpenUp**

Para el desarrollo correcto del módulo se utiliza la metodología ágil de desarrollo OpenUP. Esta preserva las características fundamentales del Proceso Unificado de Desarrollo (RUP por sus siglas en inglés), que incluye un desarrollo iterativo, casos de uso y escenarios, administración de riesgos, entre otros elementos. Excluye la mayoría de las partes opcionales de RUP e incluye varios elementos nuevos. Es un proceso ágil y ligero, que promueve al desarrollo del software sobre las buenas prácticas, haciéndolo un proceso pequeño y extensible si es necesario (híbrido, incluye partes de otros modelos). Debido a estas ventajas que presenta OpenUp se adopta para llevar a cabo el proceso de desarrollo del módulo de visualización [34].

### 1.6.4 Herramientas de desarrollo

Existen varias herramientas para el desarrollo de un software, las cuales fundamentalmente se agrupan en: herramientas CASE y herramientas de Entorno Integrado de Desarrollo (IDE por sus siglas en ingles). Las herramientas CASE son un conjunto de programas y ayudas que dan asistencia a los analistas, ingenieros de software y desarrolladores, durante todos los pasos y procesos del ciclo de vida de desarrollo de un software. Las herramientas IDE son aplicaciones, entorno de programación, para desarrollar software que está compuesto normalmente por un editor de código, un compilador o un intérprete, un depurador y algunas veces un diseñador de interfaces gráficas incorporado [40].

#### **Visual Paradigm como Herramienta CASE**

Visual Paradigm soporta el UML como lenguaje de modelado. Además provee el modelado del proceso de negocio, un generador de mapeo objeto-relación para Java, .NET y PHP. Sirve de soporte también para 13 tipos de diagramas de UML. Una característica clave es su habilidad de generar no sólo código del modelo de clases, sino también de la estructura de la base de datos relacional adecuados para sostener persistentemente la información contenida en las clases llamadas “entidad” [41].

Rational Rose es otra herramienta CASE muy similar al Visual Paradigm pero no es una herramienta multiplataforma y tiene menos facilidades que el Visual Paradigm.

#### **NetBeans como Herramienta IDE**

NetBeans IDE es una herramienta para que los programadores puedan escribir, compilar, depurar y ejecutar programas de escritorio y/o web con múltiples lenguajes de programación. Está escrito en Java. Existe además un número importante de módulos para extender el NetBeans IDE. Es de código abierto y gratuito para uso tanto comercial como no comercial. NetBeans funciona en sistemas operativos compatibles con la máquina virtual Java (Windows XP, Vista, Windows 7, Ubuntu 9.10, Solaris, Mac OS X 10.5 o superior) [42].

Eclipse es también una herramienta IDE pero a diferencia de NetBeans carece de un buen soporte de aplicaciones web. Además los plugins de Eclipse no son ni tan potentes, ni tan sencillos como el módulo que en NetBeans viene preinstalado.

### 1.6.5 Gestor de Base de Datos

Un Sistema Gestor de Base de Datos (SGBD) es una colección de programas cuyo objetivo es servir de interfaz entre la base de datos, el usuario y las aplicaciones. Se compone de un lenguaje de definición de datos, de un lenguaje de manipulación de datos y de un lenguaje de consulta. Un SGBD permite definir los

datos a distintos niveles de abstracción y manipular dichos datos, garantizando la seguridad e integridad de los mismos [43].

### **PostgreSQL como gestor de base de datos**

PostgreSQL es un servidor de base de datos relacional orientada a objetos de software libre. Está considerado como el gestor de base de datos de código abierto más avanzado del mundo. Aproxima los datos a un modelo objeto-relacional, y es capaz de manejar complejas rutinas y reglas. Ejemplos de su avanzada funcionalidad son las consultas SQL declarativas, el control de concurrencia multi-versión, el soporte de multi-usuario, transacciones, optimización de consultas, herencia, y arreglos. En este proyecto se usa PostgreSQL además de las ventajas que posee, porque la estructura de la base de la tecnología del cliente presenta dicho gestor de base de datos y se debe de cumplir con sus peticiones. Además PostgreSQL es uno de los gestores que estableció el grupo de Bioinformática BioSoft para el desarrollo de sus proyectos [34].

### **1.6.6 Servidor de Aplicaciones**

El Servidor de Aplicaciones es un software multiusuario y multitarea que permite, a través de TCP/IP, ejecutar las aplicaciones albergadas en él. Por tanto, estas aplicaciones son accesibles tanto en red local como desde Internet, facilitando la implantación del sistema cliente/servidor en el usuario final. La seguridad viene garantizada gracias al propio protocolo de comunicaciones. Está totalmente integrado con la base de datos. Permite la ejecución de funciones remotas, dotando al sistema de interconectividad entre servidores, replicación, consolidación y arquitecturas on-off [44].

### **Glassfish**

Glassfish es un servidor de aplicaciones que implementa la plataforma JavaEE5, por lo que soporta las últimas versiones de tecnologías como: JSP, JSF, Servlets, EJBs, Java API para Servicios Web (JAX-WS), Arquitectura Java para Enlaces XML (JAXB), Metadatos de Servicios Web para la Plataforma Java 1.0, y muchas otras tecnologías. Existen personas que contribuyen al desarrollo del mismo, agregándole más características importantes a Glassfish [45].

### **1.6.7 Servidor Web**

El servidor web de la plataforma alasSiRNA-Design SiRNA Web Services, donde se encuentran implementados los nuevos servicios, utiliza Arquitectura Orientada a Servicios (SOA por sus siglas en inglés). Esta arquitectura de software define la utilización de servicios para dar soporte a los



requerimientos de software del usuario. SOA proporciona una metodología y un marco de trabajo para documentar las capacidades de negocio y puede dar soporte a las actividades de integración y consolidación [46].

En la implementación de los nuevos servicios se utilizó como IDE NetBeans, como gestor de base de datos PostgreSQL y como lenguajes de desarrollo Java y WSDL.

### **Conclusiones**

En este capítulo se adentró en los principales conceptos vinculados al tema para una mejor comprensión del lector. Se realizó un análisis de las limitaciones de alasSiRNA-Design evidenciado la importancia de desarrollo del módulo. Se analizaron las tecnologías, herramientas y metodologías utilizadas en la implementación de la plataforma con el objetivo de fundamentar su correcta utilización. Como resultado del estudio de las tecnologías, metodologías y herramientas de desarrollo a utilizar se ha llegado a la conclusión de que el módulo se desarrollará usando tecnología **JSP** y por consiguiente lenguaje de programación **Java**, y como gestor de bases de datos **PostgreSQL 9.1**. Además se determinó hacer uso de UML por su estrecha integración con RUP que constituye la base de **OpenUp**, metodología adoptada. Como herramienta de modelado se utiliza **Visual Paradigm 6.4**, herramienta IDE el **NetBeans IDE 7.1 RC1** y como servidor de aplicaciones se adoptó el **Glassfish 3.1**.

### CAPÍTULO 2

#### CARACTERÍSTICAS DEL SISTEMA

"Cuando alguien desea algo debe saber que corre riesgos, por eso la vida vale la pena."

##### Introducción

En este capítulo se realiza una propuesta del sistema que se quiere implementar (módulo de visualización) usando componentes del modelo del dominio de la metodología Open/Up. De este modelo se tendrán en cuenta la definición de las entidades y los conceptos principales, así como su representación gráfica. Se lleva a cabo la captura de los requerimientos funcionales como no funcionales y los actores que intervienen en el sistema y se obtienen además, como resultado, el diagrama de casos de uso del sistema y las descripciones de los correspondientes casos de uso.

##### 2.1 Propuesta de sistema

Atendiendo a las necesidades de los investigadores del CIGB y con el fin de alcanzar una mayor calidad en las investigaciones y en los trabajos desarrollados se hace necesaria la visualización de los resultados obtenidos del servidor web de la plataforma alasSiRNA-Design, SiRNA Web Services. Estos resultados determinan la influencia de las mutaciones puntuales en los miRNAs brindando un mayor espectro en las investigaciones relacionadas con el silenciamiento genético. Teniendo en cuenta el sistema informático desarrollado previamente (alasSiRNA-Design) y que los nuevos servicios que se quieren consumir ya están implementados en SiRNA Web Services se incorporan nuevas funcionalidades a la interfaz.

Estas funcionalidades son las siguientes:

**Mostrar información de un miRNA:** Dado el identificador (ID) del miRNA definido por el especialista se muestra los miRNAs maduros contenidos en dicho miRNA y da la posibilidad de escoger uno de ellos, así como la característica que desea visualizar ya sea los SNPs presentes en el miRNA maduro seleccionado o los genes que éste silencia.

**Crear archivo Fasta:** Se brinda la posibilidad al especialista de introducir el nombre del fichero Fasta en el sistema y la descarga del mismo en el directorio que el desee.

**Visualizar SNPs influyentes de un gen:** Se muestran los SNPs influyentes en un transcrito especificado por el especialista con su nombre.

**Visualizar SNPs influyentes de un miRNA:** Se muestran los SNPs influyentes en todos los miRNAs maduros contenidos en el miRNA especificado por el especialista con su identificador.

### 2.2 Modelo de Dominio

Debido a la falta de un proceso de negocio bien definido para llevar a cabo la propuesta de sistema se decide desarrollar un modelo de dominio. Este modelo contribuye a la comprensión y descripción de las clases más importantes en el sistema. Mediante una serie de conceptos, entidades y sus relaciones, dicho modelo permite un fácil entendimiento del sistema.

Este modelo también llamado Modelo Conceptual no es más que una representación visual de los conceptos u objetos del mundo real significativos para un problema o área de interés. Representa clases conceptuales del dominio del problema. Representa conceptos del mundo real, no de los componentes de software [47].

#### **Definición de las entidades y los conceptos principales.**

Con la aplicación de un modelo del dominio (Ver Figura 1) se detectaron las siguientes entidades y conceptos (objetos):

**SiRNA Web Services:** Servidor de servicios web de la plataforma alasSiRNA-Design. Contiene los servicios que se desean visualizar: buscar información de un miRNA, buscar los SNPs influyentes en un miRNA, buscar los SNPs influyentes en un gen y crear archivo Fasta.

**Archivo Fasta:** Servicio implementado en el Web Services para crear un archivo con formato Fasta.

**Información de un miRNA:** Servicio implementado en el Web Services que busca información de todos los miRNAs maduros contenidos en un determinado miRNA.

**SNPs Influyentes de un gen:** Servicio implementado en el Web Services que busca todos los SNPs influyentes en un gen.

**SNPs Influyentes de un miRNA:** Servicio implementado en el Web Services que busca todos los SNPs influyentes por cada uno de los miRNAs maduros contenidos en un miRNA determinado.

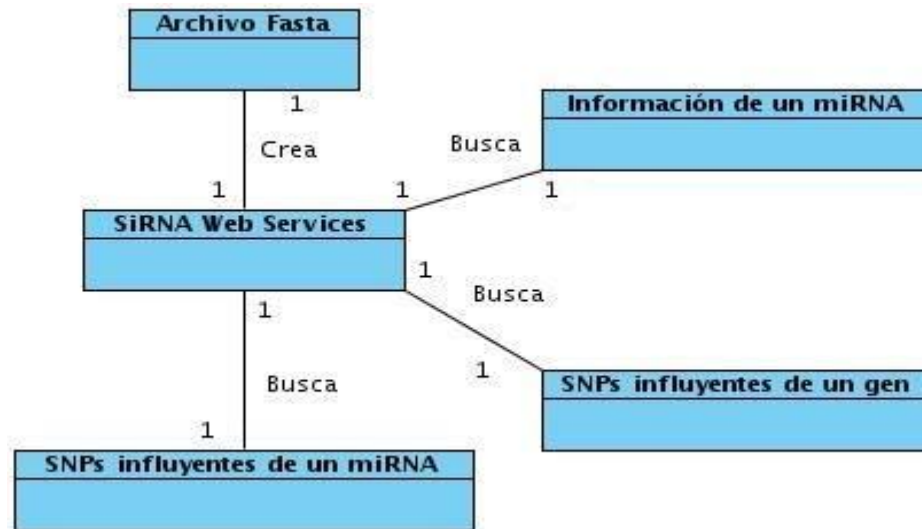


Figura 1. Modelo de Dominio

### 2.3 Especificación de los requerimientos de software

Para la especificación de los requerimientos el cliente debe tener en cuenta además de las funcionalidades o requerimientos funcionales del sistema, los requerimientos no funcionales del mismo.

#### 2.3.1 Requerimientos Funcionales

El Glosario de Terminología Estándar de Ingeniería de Software<sup>13</sup> (*IEEE*<sup>14</sup> : *Standard Glossary of Software Engineering Terminology por sus siglas en inglés*) define requerimiento como condición o capacidad que necesita un usuario para resolver un problema o lograr un objetivo. Los requerimientos funcionales permiten expresar una especificación más detallada de las responsabilidades que tiene que ser alcanzada o poseída por un sistema o componente de un sistema para satisfacer un contrato, estándar, u otro documento impuesto formalmente [48].

Los requerimientos funcionales del software propuesto son los siguientes:

**RF 1:** Mostrar SNPs presentes en un miRNA.

<sup>13</sup> Glosario de términos de las Ingenierías de Sistemas y de Software. Aporta una definición única y uniforme de los términos empleados.

<sup>14</sup> Instituto de Ingenieros Eléctricos y Electrónicos: mayor asociación mundial relacionada con el mundo de la Ingeniería y formada por profesionales de las nuevas tecnologías, como ingenieros eléctricos, en electrónica, informática, biomédica, telecomunicaciones, mecatrónica y científicos de la computación.

**RF 2:** Mostrar genes blancos de un miRNA.

**RF 3:** Guardar archivo Fasta con información de los genes en el genoma humano.

**RF 4:** Visualizar SNPs influyentes de un gen.

**RF 5:** Visualizar SNPs influyentes en el silenciamiento de un miRNA.

### 2.3.2 Requerimientos no funcionales

Los requerimientos no funcionales son propiedades o cualidades que el producto debe tener [49]. Representan las características del mismo. El módulo de visualización de servicios para la plataforma alasSiRNA-Design se desarrolla siguiendo los estándares y los requerimientos no funcionales de la aplicación ya existente. Se definen a continuación las propiedades de la plataforma que se tienen en cuenta para el desarrollo del módulo.

#### **Apariencia o interfaz externa:**

**RnF1.** Mostrar en idioma inglés con apariencia uniforme en todo el módulo los títulos de los componentes de la interfaz, los mensajes para interactuar con los usuarios y los mensajes de error. Ello es necesario porque los especialistas consultan toda su bibliografía en ese idioma, y no están familiarizados con traducciones de los términos.

**RnF2.** Representar la información de los SNPs presentes e influyentes en un miRNA como en un gen de forma gráfica.

#### **Usabilidad:**

**RnF3.** Incluir las nuevas funcionalidades que brinda el módulo en el mapa del sitio, así también como en el menú principal de la plataforma.

**RnF4.** Mostrar los títulos necesario para especificar la información que se visualiza y además que los mensajes mostrados al especialista sean lo más explicativos posibles.

**RnF5.** Garantizar la mínima cantidad de páginas para ejecutar todas las funciones posibles, es decir, agrupar funciones afines en las mismas páginas.

#### **Rendimiento:**

**RnF6.** Concebir el módulo para un ambiente cliente/servidor, garantizando la eficiencia del mismo en cuanto a la rapidez de respuesta (60 segundos) ante las solicitudes de los especialistas, al igual que la velocidad de procesamiento de la información.

#### **Soporte:**

**RnF7.** Brindar la posibilidad de mejoramiento y ampliación a otras especificaciones que se le incorporen en un futuro.

**RnF8.** Garantizar la instalación de la nueva versión de la plataforma y la capacitación de los especialistas que intercambiarán con las nuevas funcionalidades.

**Portabilidad:**

**RnF9.** Concebir el módulo para que sea multiplataforma.

**Software:**

**RnF10.** Se requiere de Windows XP (o versiones superiores) o Linux para la instalación de la aplicación tanto en la PC cliente como en los servidores. Los servidores de aplicaciones SiRNA Web Services y alasSiRNA-Design requieren de la instalación del GlassFish en su versión 3.1. Además el servidor alasSiRNA-Design requiere de un navegador web al igual que las PC clientes. El servidor de base de datos dispondrá de PostgreSQL en su versión 9.1 como gestor de base de datos.

**Hardware:**

Para el desarrollo y puesta en práctica del proyecto se requieren máquinas servidoras con los siguientes requerimientos:

**RnF11.** Para el servidor de aplicaciones de la plataforma alasSiRNA-Design y las PC clientes se necesita:

- Procesador Pentium 3 o superior.
- 512 MB de RAM.

**RnF12.** Para el servidor de SiRNA Web Services:

- Procesador Pentium 3 o superior.
- 1 GB de RAM.
- Un mínimo de 4 GB de espacio disponible en disco duro.

**RnF13.** Para el servidor de base de datos.

- Procesador Pentium 3 o superior.
- 512 MB de RAM.
- Un mínimo de 2 GB de espacio disponible en disco duro

A continuación se definen los actores y casos de usos del sistema teniendo en cuenta los requerimientos funcionales del sistema descritos anteriormente.

### 2.4 Definición de actores y casos de uso del sistema

Los cinco requerimientos funcionales del sistema se agrupan en cuatro casos de usos que se describen en el siguiente epígrafe. Además para un mayor entendimiento del sistema se modela un diagrama a partir de los actores y casos de usos identificados.

#### 2.4.1 Actores del sistema

Los actores del sistema aunque no forman parte de él, intercambian información con el sistema. Pueden representar el rol que juega una o varias personas, un equipo o un sistema automatizado [47].

A continuación se muestran los actores y el rol que desempeñan en el sistema.

**Tabla 1. Actores del sistema**

Nombre del actor	Descripción
Especialista	Representa el usuario de la versión 1.0 de alasSiRNA-Design que tiene la capacidad profesional de entender e interactuar con todas las funcionalidades específicas del servidor web.
SiRNA Web Services	Representa al servidor de servicios web que tiene desplegado los servicios que se desean visualizar o consumir por el módulo de visualización de servicios para la plataforma alasSiRNA-Design v1.0.

#### 2.4.2 Casos de Uso definidos

Los casos de uso son “fragmentos” de funcionalidad que ofrece el sistema para aportar un resultado de valor para sus actores. Son una secuencia de acciones a llevar a cabo por el sistema como respuesta a los requerimientos funcionales. Se utilizan para obtener información de cómo debe trabajar el sistema. Describen mediante el intercambio acción y reacción el comportamiento del sistema desde el punto de vista del usuario [49].

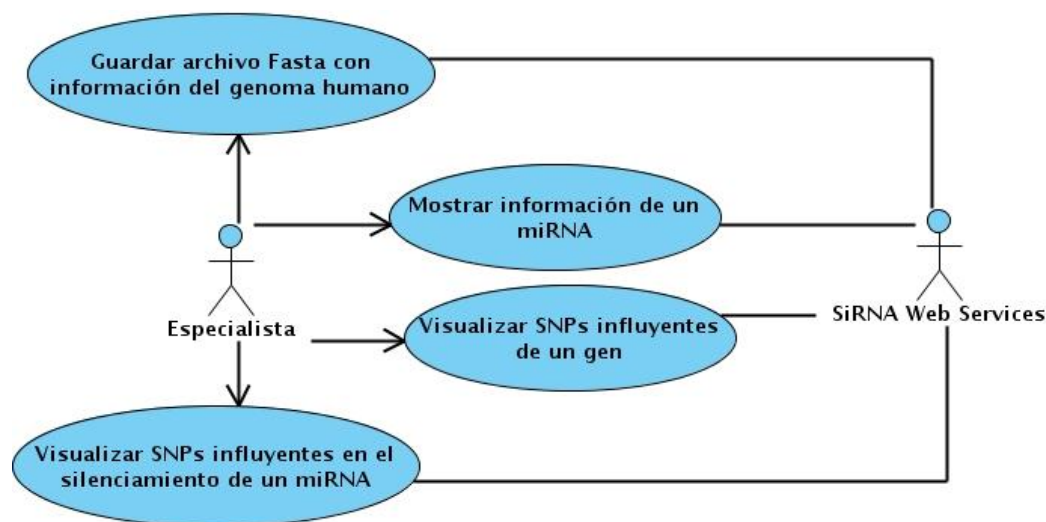
Los casos de uso definidos son los siguientes:

1. Mostrar información de un miRNA.
2. Guardar archivo Fasta con información del genoma humano.
3. Visualizar SNPs influyentes de un gen.
4. Visualizar SNPs influyentes en el silenciamiento de un miRNA.

**2.4.3 Diagrama de casos de uso del sistema.**

Los diagramas de casos de uso sirven para especificar la comunicación y el comportamiento de un sistema mediante su interacción con los usuarios y/u otros sistemas, mostrando la relación entre los actores y los casos de uso en un sistema. Además los diagramas de casos de uso se utilizan para ilustrar los requerimientos del sistema al mostrar cómo reacciona a eventos que se producen en su ámbito o en él mismo [50].

En el diagrama de casos de uso del sistema (Ver Figura 2) se evidencia como el especialista es el encargado de ejecutar las cuatro funcionalidades que brinda el módulo de visualización. Dichas funcionalidades dependen de los servicios correspondientes implementados en SiRNA Web Services.



**Figura 2.** Diagrama de caso de uso del sistema

**2.4.4 Descripción detallada de casos de uso.**

En la descripción de los casos de uso se muestra en detalles cómo actúa el sistema ante una acción del especialista. La descripción es por pasos, se tienen en cuenta por separado el cliente y el servidor como se ha visto hasta ahora. A continuación sólo se muestra la descripción del CU Visualizar SNPs influyentes de un gen (Ver más en Anexo 1).

**Tabla 2.** Descripción del CU Visualizar SNPs influyentes de un gen.

Caso de Uso:	Visualizar SNPs influyentes de un gen.
Actores:	Especialista



Propósito:	Muestra aquellos SNPs que tienen influencia en un determinado gen.
Resumen:	El caso de uso se inicia cuando el especialista desea consumir el servicio del web services que muestra los SNPs influyentes en el transcrito y la posición de estos en la secuencia del transcrito.
Referencia:	RF4
Prioridad:	Alta.
<b>Flujo Normal de Eventos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1) El especialista hace una petición a SiRNA Web Services para Visualizar SNPs influyentes de un gen. 3) El especialista introduce el nombre del transcrito.	2) El sistema muestra un formulario para que inserte el nombre del transcrito. 4) El sistema busca los SNPs influyentes así como toda la información referente a este. 5) El sistema muestra los SNPs influyentes así como toda la información referente a este.
<b>Flujo Alterno</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
	2) El sistema muestra un mensaje de error si el nombre del transcrito introducido por el especialista es incorrecto.

## 2.5 Conclusiones

En este capítulo se ganó claridad en cuanto a la concesión del sistema a construir. Los conceptos definidos fueron relacionados mediante un diagrama de Modelo de Dominio brindando de forma general cómo se desarrolla este proceso. Además se realizó el levantamiento de los requerimientos no funcionales del sistema quedando definidos trece requerimientos. Mediante la definición de los cinco requerimientos funcionales agrupados en cuatro casos de usos y los dos actores se concretó el funcionamiento del sistema que se pretende desarrollar quedando sentadas las bases para las restantes fases del proceso de diseño e implementación.

### CAPÍTULO 3

#### ANÁLISIS Y DISEÑO DEL SISTEMA

“Escoger un camino significa abandonar otros. Si pretendes recorrer todos los caminos posibles acabarás no corriendo ninguno.”

#### Introducción

Una vez definidos todos los requerimientos del sistema, se está en condiciones de traducirlos a una especificación que describa cómo implementar el sistema. Esta especificación se logra mediante la modelación del análisis y el diseño del sistema siendo este el objetivo del presente capítulo. En este capítulo se ofrece además una descripción de la arquitectura del sistema y de los patrones de diseño empleados, se diseñan los diagramas de clases e interacción para el diseño de clases web. Por otro lado se hace énfasis en la seguridad del módulo y el tratamiento de errores.

#### 3.1 Patrones Arquitectónicos

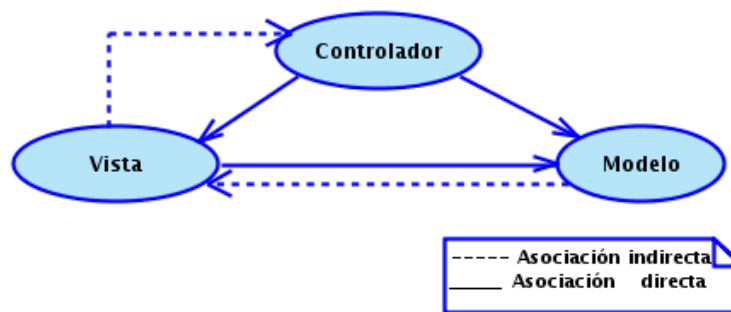
Un avance importante dentro de la construcción del software ha sido el desarrollo de la arquitectura de software, donde esta permite representar la estructura de un sistema, a un nivel mayor que el dado por la programación, o incluso por el diseño [51]. En la Arquitectura de Software, se observa que en la práctica del diseño y la implementación aparecen ciertas regularidades como respuesta a problemas similares y que al estudiarlos se agrupan en tipos específicos de acuerdo a las características de las situaciones. A estos tipos se les llamó estilos arquitectónicos, los cuales define una familia de sistemas que siguen un mismo patrón estructural [52].

El estilo arquitectónico usado para el desarrollo del módulo de visualización es de llamada y retorno, el cual refleja la estructura del lenguaje de programación y permite construir una estructura de programa relativamente fácil de modificar y ajustar a escala.

El patrón de arquitectura, descrito a continuación, expresa el esquema fundamental de organización para el módulo. Representa el nivel más alto en el sistema de patrones y cada actividad de desarrollo es gobernada por esta estructura; por ejemplo: el diseño detallado de los subsistemas, la comunicación y colaboración entre diferentes partes del módulo. Cada patrón de arquitectura ayuda a conseguir una propiedad específica en el sistema global [53].

#### Patrón Modelo-Vista-Controlador (MVC)

En la mayoría de las aplicaciones web la lógica de la interfaz de usuario cambia con más frecuencia que los almacenes de datos y la lógica de negocio. Con el Modelo Vista Controlador se trata de realizar un diseño que desacople la vista del modelo, con la finalidad de mejorar la reusabilidad. De esta forma las modificaciones en las vistas impactan en menor medida en la lógica de negocio o de datos. Para aplicaciones J2EE, la arquitectura MVC satisface las necesidades del módulo [54]. El patrón indica que se deben establecer tres componentes o capas en la arquitectura, y que cada una de estas, sólo se comunica con la adyacente (Ver Figura 3). Además un modelo puede tener diversas vistas, cada una con su correspondiente controlador. La plataforma alasSiRNA-Design implementa el Controlador Frontal, dentro de una de las variantes del patrón Modelo-Vista-Controlador (MVC) con páginas Java, con la peculiaridad que accederá a los datos a través de SiRNA Web Services



**Figura 3.** Modelo-Vista-Controlador

### 3.2 Vista Lógica del sistema

La vista lógica abarca el comportamiento arquitectónicamente significativo. Describe el diseño más importante de las clases, su organización en paquetes y subsistemas, y la organización de estos en capas. Muestra las funcionalidades diseñadas en el interior del sistema, en términos de la estructura estática y comportamiento dinámico [55].

La vista lógica del sistema (Ver Figura 4) tiene como objetivo principal representar las clases del módulo. Se pueden especificar dos subsistemas. El subsistema que representa al módulo de visualización de servicios para la plataforma alasSiRNA-Design, el cual utiliza los servicios correspondientes implementados en SiRNA Web Services.

Elementos del patrón MVC en el subsistema módulo de visualización de servicios para la plataforma alasSiRNA-Design (Ver Figura 4):

**Vista:** Contiene las clases dedicadas puramente a la creación de objetos de presentación. Para trabajar con las vistas hemos empleado el uso de plantillas. La capa Vista fue separada en 2 paquetes:

Paquete Web Pages: contiene las páginas que el cliente ve en su navegador web.

Paquete imageservlets: contiene los servlets encargados de generar las distintas imágenes que se visualizan de forma dinámica al cliente a través del navegador web.

**Controlador:** Responde a eventos o acciones del usuario e invoca cambios en el modelo y probablemente en la vista. Según el patrón Controlador Frontal, se trabaja con estructuras denominadas acciones que no son más que clases encargadas de llevar el control del flujo en el módulo. En cada acción implementada se usan los servicios brindados por SiRNA Web Services. Esta capa fue dividida en tres paquetes. Cada uno de estos paquetes contiene clases encargadas de llevar a cabo el flujo del sistema y preparar la información que se enviará a la capa Vista para ser mostrada.

Paquete controller: contiene la clase “controllerServlet” que controla a todas las acciones de este tipo.

Paquete base: brinda la Interfaz “Action” de donde tienen que heredar todas las acciones para poder implementarse y acoplarse al módulo correctamente.

Paquete actions: brinda un conjunto de clases que son las que dan la estructura general a la armazón y un correcto funcionamiento a todas las acciones. En el módulo estas clases son las encargadas de responder a las acciones del usuario invocando cambios en la vista.

**Modelo:** Representa específicamente el dominio de la información sobre la cual funciona el sistema. El modelo encapsula los datos y las funcionalidades. En el módulo está compuesta por 5 paquetes:

Paquete dbmodels: contiene todos los modelos de las acciones que se conectan a la base de datos.

Paquete exceptions: contiene las clases que definen los distintos tipos de excepciones que puede dar el módulo.

Paquete utils: contiene las clases persistentes de la base de datos, con las que se trabajan en cada consulta realizada a esta base de datos. Cuando se invocan los servicios de SiRNA Web Services se devuelve algún objeto de estos modelos.

Paquete sequencedesignmodel: contiene las clases responsables del negocio para las acciones que no se conectan a la base de datos.

Paquete Web Services Referencias: contiene todas las referencias de los servicios que están implementadas en SiRNA Web Services.

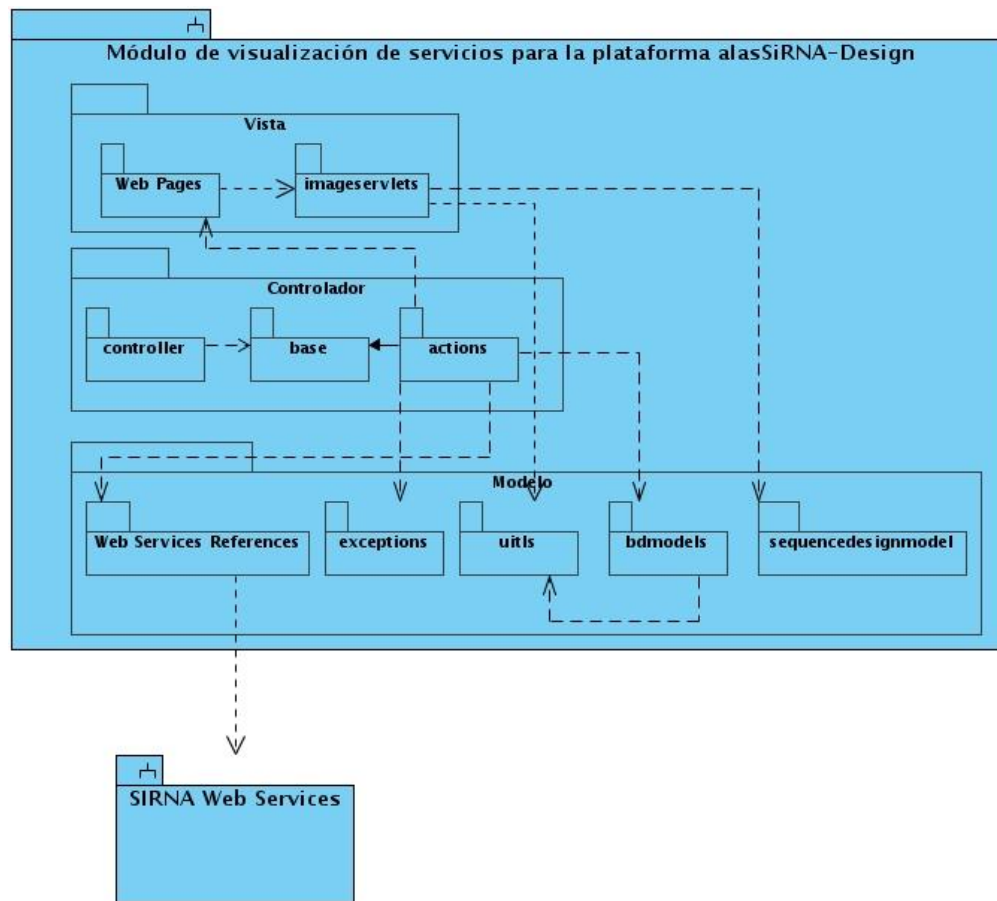


Figura 4. Vista lógica del sistema

### 3.3 Patrones de diseño

Una solución a un problema que se usa repetidamente en contextos similares con algunas variantes en la implementación es lo que se define como un patrón de diseño. Un patrón de este tipo identifica, abstrae y nombra los aspectos elementales de una estructura de diseño, donde los componentes, son las clases y objetos, y sus mecanismos de interacción son mensajes. Cada patrón prescribe una estructura de clases, sus roles y colaboraciones, y una adecuada asignación de métodos para resolver un problema de diseño en una manera flexible y adaptable. Ayudan a elegir diseños alternativos que hacen un sistema reutilizable y evitan alternativas que comprometan la reutilización [56].

Los patrones **GRASP** son patrones generales de software para asignación general de responsabilidades, es el acrónimo de "Asignación de Responsabilidad General Patrones de Software". Las responsabilidades

están relacionadas con las obligaciones de un objeto en cuanto a su comportamiento. En la solución del problema principalmente se utilizan los siguientes patrones GRASP:

**Creador:** Identifica quién debe ser el responsable de la creación o instancia de nuevos objetos o clases. Se brinda un soporte al bajo acoplamiento. En el módulo las clases que se encuentran en el paquete actions es un ejemplo de este patrón, estas son las encargadas de la creación de objetos del tipo de las clases que están relacionadas en el paquete dbmodels.

**Controlador:** Un Controlador es un objeto que se encarga de manejar un evento del sistema. Este no realiza mucho trabajo por sí mismo, sino que delega en otros objetos, coordina y controla el trabajo que se necesita hacer. En el módulo todas las clases que están en el paquete actions son clases controladoras, clases que le asignan a las clases Web Services References la funcionalidad que le corresponde.

**Experto:** Describe la asignación de responsabilidades al experto en información, clase que posee la información necesaria para cumplir con la responsabilidad. En el módulo las clases de la capa de negocio y las de acceso a datos son las que mejor reflejan este patrón donde cada una realiza funciones específicas.

**Alta Cohesión:** El uso del patrón experto en gran medida en el módulo permite una alta cohesión, el cual caracteriza a las clases con responsabilidades estrechamente relacionadas que no realicen un gran trabajo. En el módulo se tiene bien definido las responsabilidades de cada clase, por ejemplo, se definen clases del negocio por cada caso de uso, para evitar la sobrecarga de funcionalidades en una clase.

**Bajo Acoplamiento:** Cada clase está acoplada (relacionada) a las clases estrictamente necesarias, garantizando un bajo impacto de los cambios que se producen en una clase para las demás clases que se relacionan con ella. Los patrones controlador y experto garantizan en el módulo un bajo acoplamiento.

En busca de mejores resultados y una solución robusta también se tuvieron en cuenta los patrones **GOF**, acrónimo de “Banda de los Cuatro” descritos en el libro “Design Patterns: Elements of Reusable Object-Oriented Software”. Los cuales agrupan 23 patrones de diseño aplicados usualmente por expertos diseñadores de software orientado a objetos. En el módulo de visualización principalmente se utiliza el siguiente patrón GOF:

**Template Method (Método plantilla):** Patrón de comportamiento que define en una operación el esqueleto de un algoritmo, delegando en las subclasses algunos de sus pasos, esto permite que las subclasses redefinan ciertos pasos de un algoritmo sin cambiar su estructura [57].

El patrón Template Method se evidencia en las clases que se encuentran en el paquete action que heredan de Action y en las clases del paquete utils que heredan de Serializable.

### 3.4 Modelo de Diseño

El Modelo de Diseño es un modelo de objetos que describe la realización de los casos de uso, y sirve como una abstracción del modelo de implementación. Está compuesto por clases del diseño, subsistemas, paquetes, colaboraciones, y las relaciones entre ellos. El diseño es un refinamiento del análisis que tiene en cuenta los requerimientos no funcionales, en definitiva cómo cumple el sistema sus objetivos [58]. El diseño debe ser suficiente para que el sistema pueda ser implementado sin ambigüedades.

Diagrama de clases web del diseño.

La Ingeniería Web basada en UML es una metodología propuesta enfocada al modelado de aplicaciones Web basada en la extensión de la semántica del UML mediante la utilización de estereotipos. A diferencia del resto de los diagramas de clases, en el que se construye para las aplicaciones web son mucho más importantes la modelación de la lógica y estado del negocio que los detalles de presentación [59].

Se opta en este caso por modelar los artefactos del sistema, es decir: modelar las páginas, los enlaces entre estas, todo el código que irá creando las páginas, así como el contenido dinámico de estas, una vez que estén en el navegador del cliente para obtener un nivel correcto de abstracción y detalle que permita obtener un resultado final.

Los estereotipos que usa esta extensión son:

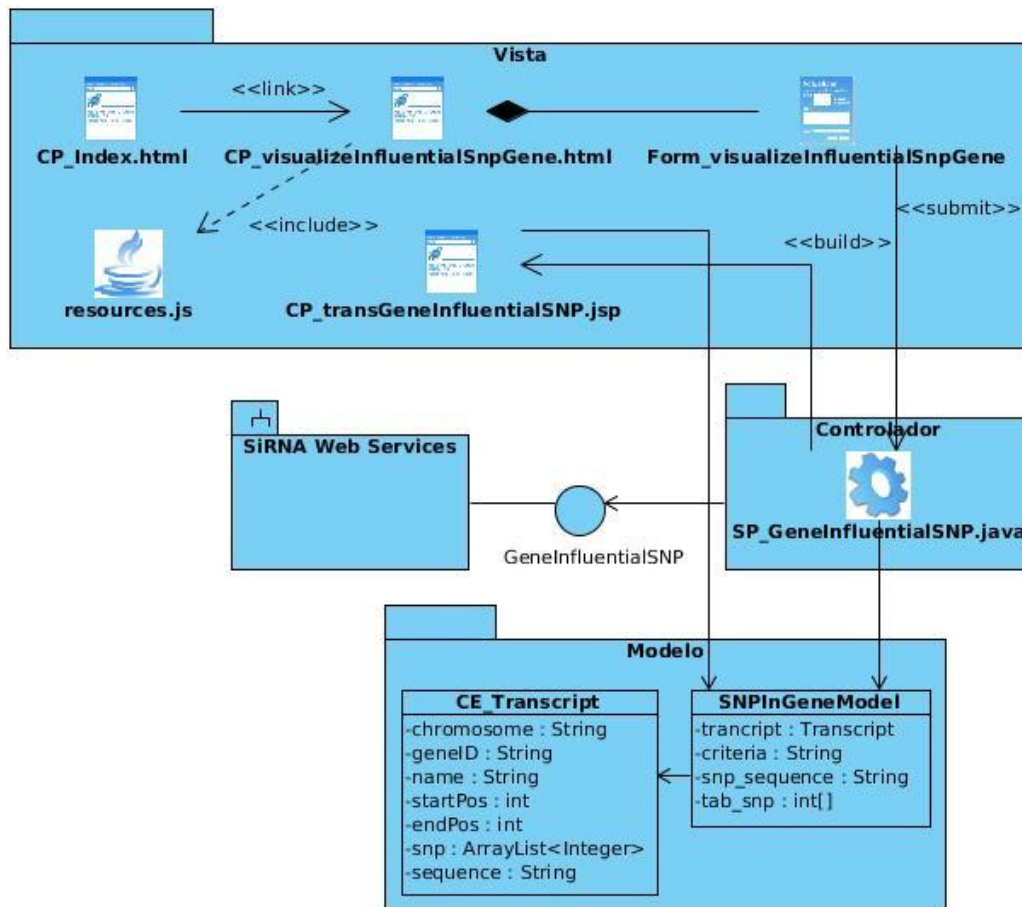
**<<Server Page>>** Representa la página con código que se ejecuta en el servidor y además interactúa con recursos en el mismo.

**<<Client Page>>** Es una página Web, con formato HTML; mezcla de datos, presentación y lógica de trabajo.

**<<Form>>** Grupo de elementos de entrada que forman parte de una página cliente que constituyen sus atributos.

Existen otros componentes como las clases JavaScript para hacer las validaciones en la entrada de datos. El diseño de las clases en Open/Up es basado en casos de uso por lo que se diseña un diagrama con estereotipos web por casos de uso significativos.

Los diagramas de clases del diseño del módulo evidencian cómo interactúan las capas de la arquitectura MVC y la interacción de las mismas con SiRNA Web Services (Ver Figura 5). Para tener una idea general del diseño se presenta un solo diagrama de clases basado en el caso de uso Visualizar SNPs influyentes en un gen (Ver más en Anexo 2).



**Figura 5.** Diagrama de clases del diseño del CU Visualizar SNPs influentes en un gen

Mediante la página principal (`CP_index.html`) se accede a las funcionalidades del módulo, donde cada funcionalidad tiene su página html asociada (`<<Client Pages>>`). En este caso de uso mediante la página principal se accede a la página web `CP_visualizeInfluentialSnpGene.html`. Esta página contiene un formulario asociado (`<<Form>>`) que captura el nombre del transcrito entrado por el usuario, dicho nombre es validado en JavaScript con la página `resources.js`.

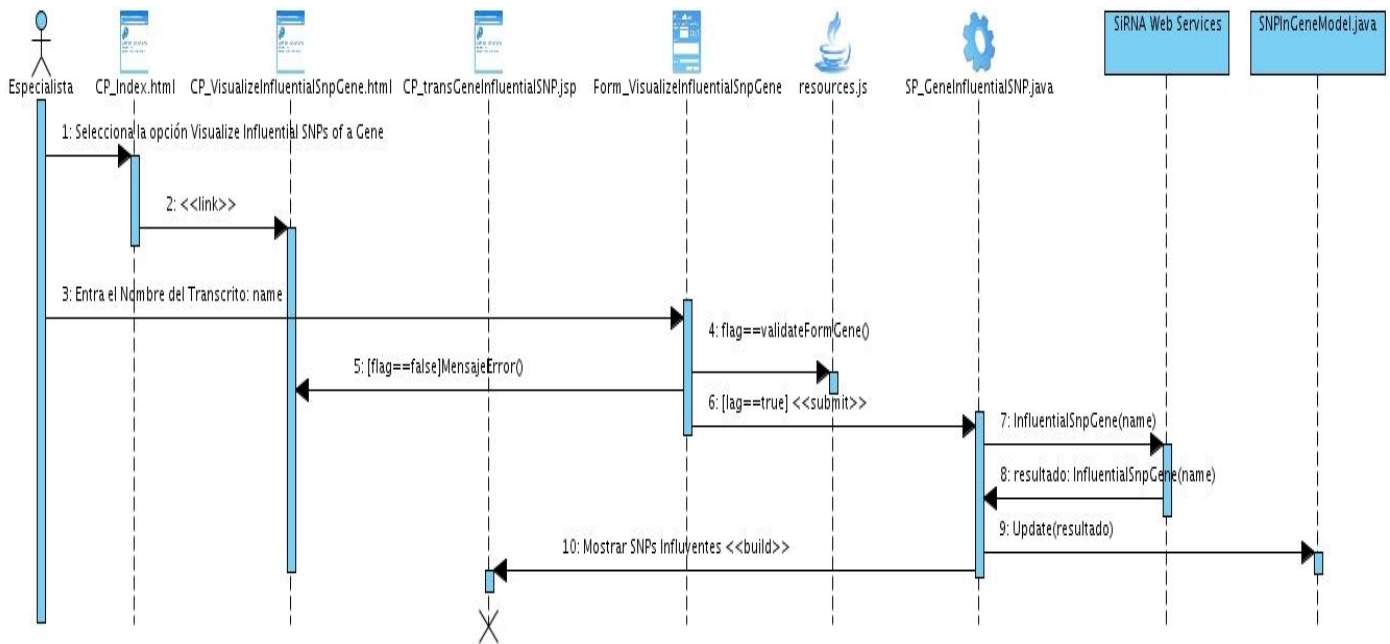
Si la validación es correcta dicho formulario brinda los datos de entrada a la página servidora (`<<Server Pages>>`) `SP_GeneInfluentialSNP.java`. Esta página controla la acción del usuario, invoca la funcionalidad correspondiente en `SiRNA Web Services` y el resultado de la funcionalidad lo almacena en la clase `SNPInGeneModel.java` de la capa de modelo. La página servidora genera la página jsp



CP\_transGeneInfluentialSNP.jsp donde se visualizan los resultados guardados en la clase SNPInGeneModel.java.

### 3.5 Diagramas de interacción. Diagramas de Secuencia

Los diagramas de interacción se utilizan para modelar los aspectos dinámicos de un sistema. Proporcionan una realización física de la ejecución de los casos de uso en términos de clases del diseño. Están formados por un conjunto de objetos y sus relaciones, incluyendo los mensajes que se pueden enviar entre ellos. Estos a diferencia de los de colaboración, destacan el orden temporal de los mensajes [60]. A continuación se muestra el diagrama de secuencia correspondiente al CU Visualizar SNPs influyentes en un gen (Ver más en Anexo 3).



**Figura 6.** Diagrama de secuencia del CU Visualizar SNPs influyentes en un gen

Como se muestra en la Figura 6 para la visualización de los SNPs influyentes de un gen, el especialista accede desde la página principal CP\_index.html a la página web CP\_visualizeInfluentialSnpGene.html. Esta página contiene un formulario con el nombre del transcrito que se desea analizar y en la misma se valida dicho nombre invocando el método validateFormGene() de la clase JavaScript resource.js. Si el nombre del transcrito no es válido (false) se muestra un mensaje de error dándole la posibilidad al especialista de introducir nuevamente el nombre. Si el nombre del transcrito es válido (true), este se usa en la clase controladora SP\_GeneInfluentialSNP.java como dato de entrada para la ejecución del servicio

InfluentialSNPGene en SiRNA Web Services. Antes de generar la vista final con los resultados la clase controladora almacena en la clase SNPInGeneModel.java los resultados obtenidos por el servicio web invocado, para luego ser usado por la página CP\_transGeneInfluentialSNP.jsp en la visualización de los SNPs influyentes en el transcrito identificado.

### 3.6 Tratamiento de errores

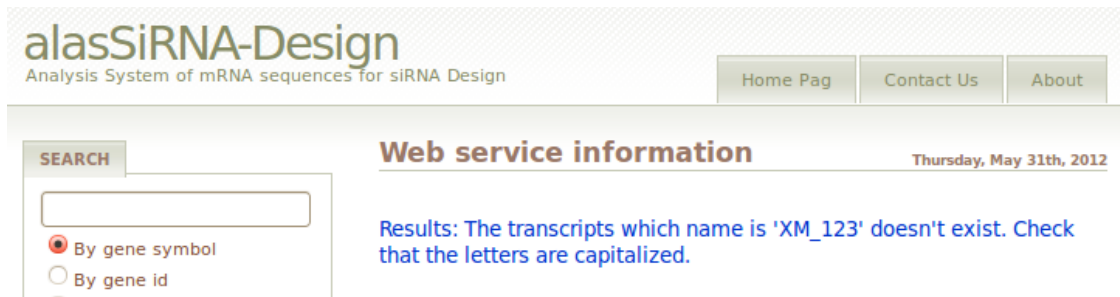
Con el objetivo de prevenir errores por parte del especialista, se verifica siempre la integridad de los datos antes de ser utilizados para realizar las búsquedas en la base de datos. Todas las prevenciones se hacen con el objetivo de evitar acciones innecesarias y resultados con errores.

El módulo de visualización hace uso de mensajes para señalarle al especialista que debe rectificar los datos introducidos antes de llevar a cabo las operaciones solicitadas por el mismo. Es de gran importancia verificar la integridad de la información introducida para de esta forma obtener un resultado de calidad. Un ejemplo de esta validación es cuando el especialista no llena el campo correspondiente al ID de un miRNA (Ver Figura 7).



**Figura 7.** Pantalla de error. Campo vacío

En el módulo también se tratan los errores a nivel de servidor en cuanto a las búsquedas que se realizan en la base de datos. Cuando un usuario busca una información que no está almacenada, el sistema a través de excepciones propias trata este tipo de error. El problema a resolver con esto es que cada vez que se realiza una búsqueda, esta desencadena dos o tres búsquedas más, si la primera existe. Una vez que se comprueba que no existe la primera, el sistema controla que las demás no se ejecuten a través de la excepciones que el implementa. Con esto no sólo se evita búsquedas innecesarias sino también demora de tiempo en los procedimientos. En el sistema esto ocurre cuando se introduce un ID de un miRNA o el nombre de un transcrito que no está en la base de datos (Ver Figura 8).



**Figura 8.** Pantalla de error. El transcrito no existe en la base de datos.

Cuando en el módulo ocurren errores inesperados, esta los trata de manera que no se pierde su integridad. Al ocurrir estos sucesos, el sistema los captura y envía al especialista a una página de error mostrándole un mensaje.

### 3.7 Seguridad

La seguridad del módulo se evidencia a través del gestor de base de datos PostgreSQL. A la información almacenada en este sólo tendrá acceso el administrador del sistema, con otra aplicación en este caso de escritorio la cual actualizará las tablas de la base de datos de manera local en el centro cliente de la plataforma.

En el módulo, se evidencia también la seguridad a través de la dirección web que se muestra en el navegador donde se está ejecutando el sistema. Todas las páginas se muestran bajo una misma dirección web, y las direcciones no coinciden con el nombre de la página mostrada. Con esto se evita que se acceda a las distintas páginas del sistema sin las precondiciones correspondientes, ya que se afectaría la integridad de toda la herramienta.

### 3.8 Conclusiones

En este capítulo se expusieron los principales artefactos del análisis y diseño del módulo: diagramas de clases del análisis, diagrama de secuencia como diagrama de interacción, diagramas de clases web del diseño y el diagrama de despliegue. El módulo de visualización utiliza el estilo arquitectónico de llamada y retorno y dentro del mismo el patrón MCV. Durante el desarrollo del módulo se hizo uso de patrones de diseño, entre los GRASP prevalecen los patrones creador, controlador, experto, bajo acoplamiento y alta cohesión y entre los GOF: Template Method. En general se ganó claridad en cuanto a la concesión del sistema a construir y se sentaron las bases para la fase de implementación y prueba del módulo.

### CAPÍTULO 4

#### IMPLEMENTACIÓN Y PRUEBA

“El camino es el que nos enseña la mejor forma de llegar y nos enriquece mientras lo estamos cruzando.”

#### Introducción

Este capítulo está enfocado en el flujo de trabajo de implementación para dar solución a los requerimientos especificados. Se realiza el modelo de implementación a partir de los resultados del flujo de trabajo anteriormente descrito: Diseño. Se visualizan los 4 servicios implementados en SiRNA Web Services para la correcta ejecución de las nuevas funcionalidades. Además se describen las pruebas realizadas al módulo.

#### 4.1 Modelo de Implementación

El modelo de implementación describe como los elementos del diseño se implementan en términos de componentes, ficheros de código fuente, ejecutables, datos, entre otros. Describe también cómo se organizan los componentes de acuerdo a los mecanismos de estructuración disponibles en el entorno de implementación y lenguajes de implementación empleados, y cómo dependen los componentes unos de otros [60]. Esta descripción es de gran utilidad a la hora de implementar el módulo, facilita la organización del trabajo y lo hace más entendible a los desarrolladores. El modelo de implementación está compuesto por el diagrama de despliegue y el diagrama de componentes.

##### 4.1.1 Diagrama de Despliegue

Modelo de objeto que describe la distribución física del sistema en términos de cómo se distribuyen las funcionalidades entre los nodos de cómputo. Cada nodo representa un recurso computacional, generalmente un procesador (capacidad de procesamiento) o un dispositivo (sin capacidad de procesamiento) sobre el que se pueden desplegar los componentes; estos tienen conexiones bidireccionales que pueden ser USB, HTTP, JDBC, ADO, HTTPS, entre otras [60].

Para el despliegue de la plataforma (Ver Figura 9) desde una PC cliente el especialista accede a la aplicación alasSiRNA-Design (nodo alasSiRNA-Design) mediante el protocolo HTTP. La aplicación alasSiRNA-Design, que contiene implementado el módulo de visualización, se conecta mediante el protocolo HTTP al servidor SiRNA Web Services (nodo SiRNA Web Services). En el nodo Servidor DB, se encuentra la base de datos de la plataforma. A esta base de datos se accede a través del componente acceso a datos DAO que se encuentra en el nodo SiRNA Web Services usando el protocolo JDBC<sup>15</sup>.

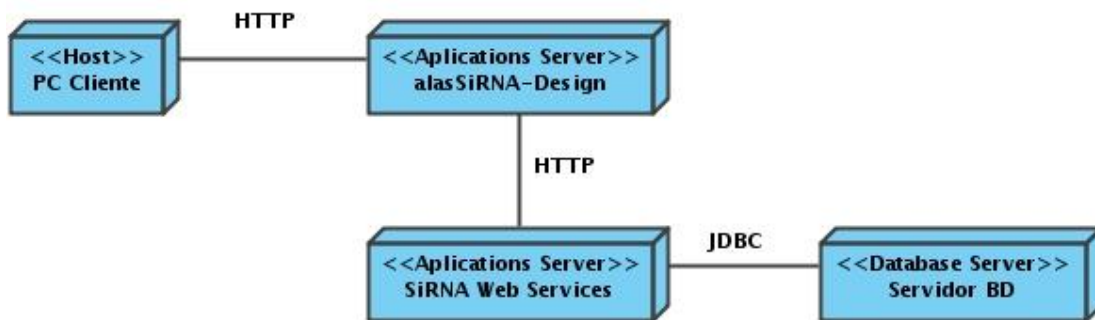


Figura 9. Diagrama de Despliegue

### 4.1.2 Diagrama de Componentes

El diagrama de Componentes muestra las organizaciones y dependencias lógicas entre componentes de software, sean estos ficheros de código fuente, binarios o ejecutables. Los elementos de modelado que lo conforman son los componentes y paquetes y muestran la estructura del sistema en términos de implementación a un alto nivel [60].

A continuación se muestra el diagrama de componentes asociado al CU Visualizar SNPs influyentes de un gen. En el cual se evidencia cada una de las capas del patrón MVC (ver Figura 10).

---

<sup>15</sup> Acrónimo de Java Database Connectivity. Es la API de Java que utiliza la herramienta Hibernate para lograr la persistencia de los objetos. Permite a cualquier programa Java acceder a sistemas de bases de datos de forma homogénea.

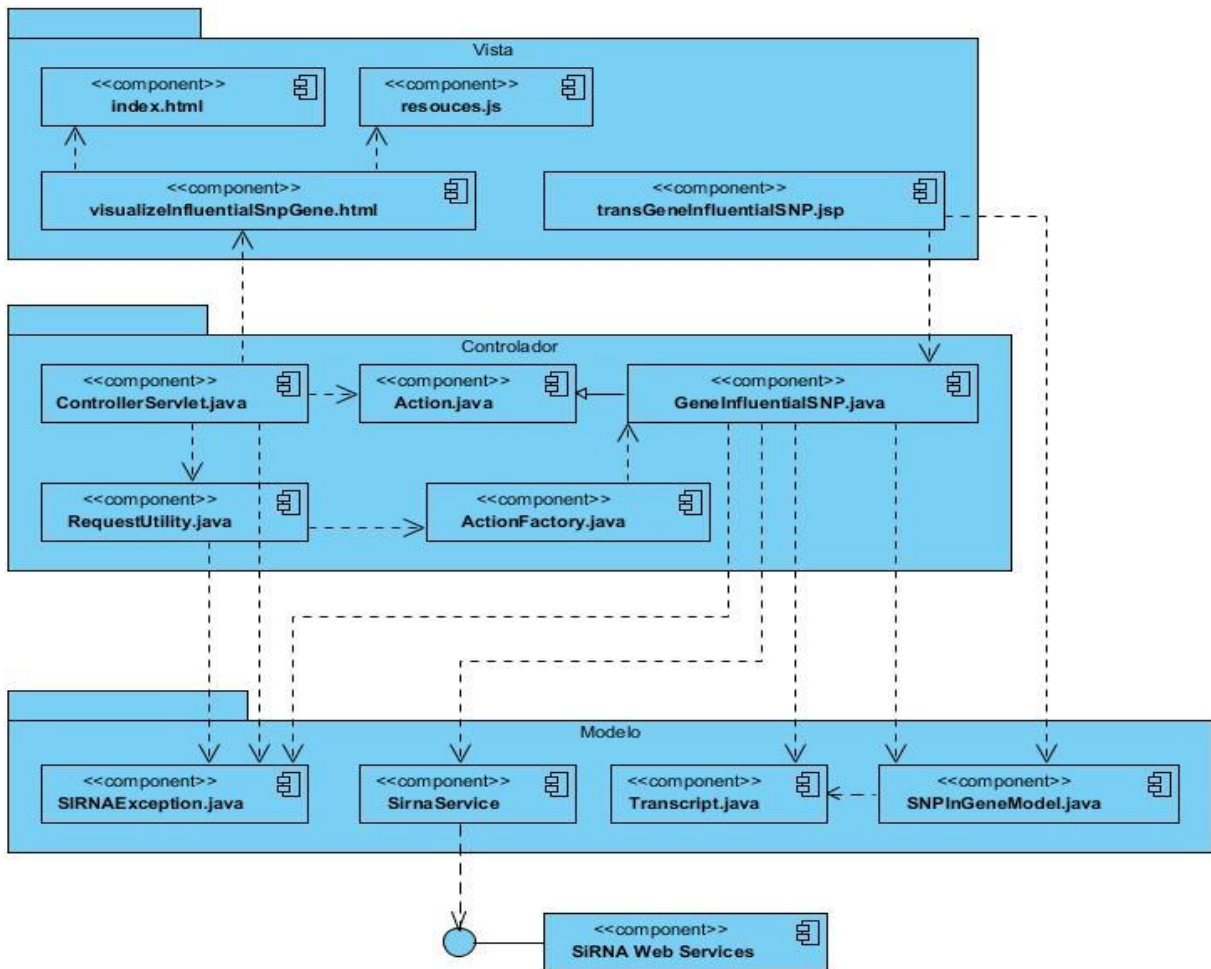


Figura 10. Diagrama de componentes del CU Visualizar SNPs influyentes de un gen

**Descripción de los componentes:**

**index.html:** página principal de la plataforma donde se encuentra el hipervínculo a la funcionalidad que se dese visualizar “Visualize influential SNPs of a Gene” (Ver Figura 11).

```

<li><a href="displayMirnasInfo.html">Display information of miRNAs</a></li>
<li><a href="createFastaFile.html">Create Fasta file</a></li>
<li><a href="visualizeInfluentialSnpGene.html">Visualize influential SNPs of a Gene</a></li>
<li><a href="visualizeInfluentialSnpMirna.html">Visualize influential SNPs of a miRNA</a></li>

```

Figura 11. Código de la página index.html

**visualizeInfluentialSnpGene.html:** página correspondiente a la funcionalidad que se desea visualizar (Ver Figura 12). Contiene un formulario para validar el nombre del transcrito que el especialista desea analizar (criterio), llamando a la función `validateGeneForm()` de la página JavaScript `resources.js`. Esta página mediante el componente de tipo `hidden` envía el nombre del evento que se desea ejecutar ("GeneInfluentialSNP") a la clase `ControllerServlet.java` si la validación en JavaScript es correcta (`true`).

```

<form id="mirnaForm" name="geneForm" method="post" action="main.html" onsubmit="return validateGeneForm()">
  <fieldset>
    <legend>
      Visualize influential SNPs of a gene
    </legend>
    <p>
      <label for="r005" class="right">
        Enter the name of the transcript:
      </label>
      <input type="text" name="criteria" class="textField" />
    </p>
    <p>
      <input type="submit" name="submit" id="submit" class="button" value="Display" />
      <input type="hidden" name="action" id="action" value="GeneInfluentialSNP" />
    </p>
  </fieldset>
</form>

```

Figura 12. Código de la página visualizeInfluentialSnpGene.html

**resources.js:** página JavaScript con las validaciones a los datos de entrada. Se ejecuta la función `validateGeneForm()` (Ver Figura 13 ). Esta función valida si el campo del nombre del transcrito está vacío y si el mismo es correcto o no. Si el nombre entrado por el usuario contiene dos letras seguidas por un guión bajo (`_`) y luego números devuelve `true`; de lo contrario envía un mensaje de error al especialista.

```
function validateGeneForm(){
    var textCriteria = document.getElementById("mirnaForm").criteria.value;
    if(textCriteria.length==0){
        alert("Please, type a name in the textbox.");
        return false;
    }else{
        var array = textCriteria.split("_", 100);
        if(array.length != 2){
            alert("Results: The transcripts which name is '"+textCriteria+"' isn't valid.");
            return false;
        }else{
            var expReg = new RegExp("[a-zA-Z]*$");
            var text= array[0];
            if(!expReg.test(text) || text.length != 2){
                alert("Results: The transcripts which name is '"+textCriteria+"' isn't valid.");
                return false;
            }
            expReg = new RegExp("[0-9]*$");
            text= array[1];
            if(!expReg.test(text)){
                alert("Results: The transcripts which name is '"+textCriteria+"' isn't valid.");
                return false;
            }
        }
    }
    return true;
}
```

**Figura 13.** Código de la página `resources.js`



**ControllerServlet.java:** clase controladora de los eventos de la capa vista. Hereda de la clase abstracta `HttpServlet`. `ControllerServlet.java` crea un objeto de la clase `RequestUtility.java` para ejecutar la acción enviada por la página `visualizeInfluentialSnpGene.html` (Ver Figura 14).

```
public class ControllerServlet extends HttpServlet {

    protected void processRequest(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException, Exception
    {
        try
        {
            RequestUtility reqUtil = new RequestUtility(request);

            Action action = reqUtil.getAction();

            action.execute(request, response);

            request.setAttribute("model", action.getModel());

            RequestDispatcher reqDisp = request.getRequestDispatcher(action.getView());
            reqDisp.forward(request, response);
        }
        catch(SIRNAException _s)
        {
            RequestDispatcher reqDisp = request.getRequestDispatcher("error_page_user.jsp?title="+_s.getTitle()+"&errorMessage="+_s.getMessage());
            reqDisp.forward(request, response);
        }
    }
}
```

**Figura 14.** Código de la clase `ControllerServlet.java`

**RequestUtility.java:** clase que captura la acción “GeneInfluentialSNP” enviada por la página `visualizeInfluentialSnpGene.html` para crear un objeto de la clase `ActionFactory.java` (Ver Figura 15).

```

public class RequestUtility
{
    HttpServletRequest request;

    public RequestUtility(HttpServletRequest _request)
    {
        request=_request;
    }

    public Action getAction() throws Exception
    {
        if(request.getParameter("action")!=null)
        {
            String action = request.getParameter("action");
            return ActionFactory.createAction(action);
        }
        throw new SIRNAException("Error Page", "Error Message: Your are trying to access to an unknown resource.");
    }
}

```

Figura 15. Código de la clase RequestUtility.java

**ActionFactory.java:** clase abstracta que identifica la acción que se va a ejecutar en correspondencia a la enviada por la página visualizelInfluentialSnpGene.html (Ver Figura 16). Crea un objeto de la clase GeneInfluentialSNP.java para ejecutar dicha acción.

```

public abstract class ActionFactory
{
    public static Action createAction(String _action) throws Exception
    {
        if(_action.equals("CreateFastaFile"))
            return new CreateFastaFile();

        if(_action.equals("DisplayMirnasInfo"))
            return new DisplayMirnasInfo();

        if(_action.equals("GeneInfluentialSNP"))
            return new GeneInfluentialSNP();

        if(_action.equals("MirnaInfluentialSNP"))
            return new MirnaInfluentialSNP();
    }
}

```

Figura 16. Código de la clase ActionFactory.java

**GeneInfluenialSNP.java:** clase que hereda de Action.java y ejecuta el servicio web correspondiente obteniendo los datos que devuelve el mismo para el nombre del transcrito especificado por el especialista (criteria). A dicho servicio se accede a través de su referencia en SirnaService (Web Services References). Los resultados obtenidos después de ejecutado el servicio web se almacenan en la clase SNPInGeneModel.java como un objeto de tipo la clase Transcript.java y se muestra la vista final con los resultados en la clase transGeneInfluenialSNP.jsp (Ver Figura 17). La clase GeneInfluenialSNP.java para el tratamiento de las excepciones instancia la clase SIRNAException.java.

```

try
{
    service.webservice.SirnaService service = new service.webservice.SirnaService();
    service.webservice.Sirna port = service.getSirnaPort();
    service.webservice.Transcript result = port.geneInfluenialSNP(criteria);
    t = new Transcript(result.getChromosome(), result.getGeneID(), result.getName(),
        result.getStartPos(), result.getEndPos(), result.getSequence(), result.getSnpsSequence());
    ArrayList<Integer> snps = new ArrayList<Integer>();
    for (int i = 0; i < result.getSnps().size(); i++) {
        snps.add(result.getSnps().get(i));
    }
    t.setSnps(snps);
}
catch (Exception ex)
{
    throw new SIRNAException("Web service information", ex.getMessage());
}
}
model = new SNPInGeneModel(_request);
model.setTranscript(t);
model.setSnps(t.getSnps());
model.initSNPSequence();
view = "transGeneInfluenialSNP.jsp";

```

Figura 17. Código de la clase GeneInfluenialSNP.java

**Action.java:** clase genérica que define un conjunto de funcionalidades que serán implementadas por otras clases (GeneInfluenialSNP.java) (Ver Figura 18).

```

public interface Action
{
    public void execute(HttpServletRequest _request, HttpServletResponse _response) throws Exception;
    public String getView();
    public Object getModel();
}

```

Figura 18. Código de la clase Action.java

**SirnaService:** Referencia en el paquete Web Services References a SiRNA Web Services. Contiene las referencias a todos los servicios implementados en el servidor web.

**SNPInGeneModel.java:** clase modelo que almacena los resultados que se van a visualizar, con la finalidad de ser instanciada por la página transGeneInfluentialSNP.jsp (Ver Figura 19).

```
public class SNPInGeneModel
{
    private Transcript transcript;
    private String criteria;
    private String snp_sequence;
    private int[] tab_snp;

    public SNPInGeneModel(HttpServletRequest _request){...}
    public Transcript getTranscript(){...}
    public void setTranscript(Transcript transcript){...}
    public String getCriteria() {...}
    public String getSNPSequence(){
        return snp_sequence;
    }
}
```

**Figura 19.** Código de la clase SNPInGeneModel.java

**Transcript.java:** clase de la capa de modelo que contiene los atributos que definen las características de un transcrito (Ver Figura 20).

```
public class Transcript implements Serializable
{
    private String chromosome;
    private String geneID;
    private String name;
    private int startPos;
    private int endPos;
    private ArrayList<Exon> exons;
    private ArrayList<Integer> snps;
    private int[] tab_snp;
    private String sequence;
    private String snpSequence;
```

Figura 20. Código de la clase Transcript.java

**transGeneInfluentialSNP.jsp**: página de la clase vista que visualiza de forma gráfica el resultado que se obtuvo del servicio web. Esta obtiene los resultados instanciando la clase SNPInGeneModel.java. Con dichos resultados diseña un gráfico con imágenes mostrando de forma sencilla todos los datos del transcrito que el especialista identificó (Ver Figura 21).

```

<table style="border-collapse:collapse"><tr>
  <%
    int flag = 0;
    int n = 0;
    int bases = end - start;
    for (int i = 0; i < 234 - cantSNP; i++) {
      if (n < cantSNP) {
        int preal = (234 * (snps.get(n) - start)) / bases;
        if (i == preal) {
          flag = 1;%>
        }
      }
    }
  </tr></table>
  <td style="padding:0px">
    <a href="#">
      <br>Position:<%=snps.get(n)%');return false;">
    </img></a></td>
  <td style="padding: 0px"></img></td>
  <td style="padding: 0px"></img></td>
  <td style="padding: 0px"></img></td>
  </tr></table>
  <%
    if(flag == 0){
  </tr></table>
  <p style="color: #FF4545;Font-size: 120%; text-align: center"><br/>It doesn't exist any influential SNPs in this
  <br/> - Influential SNP
  
```

Figura 21. Código de la página transGeneInfluentialSNP.jsp

**SIRNAException.java**: clase que trata las excepciones que generan las funcionalidades (Ver Figura 22).

```

public class SIRNAException extends Exception
{
  private String title;

  public SIRNAException(String _title, String _message)
  {
    super(_message);
    title=_title;
  }

  public String getTitle() {
    return title;
  }
}

```

Figura 22. Código de la clase SIRNAException.java

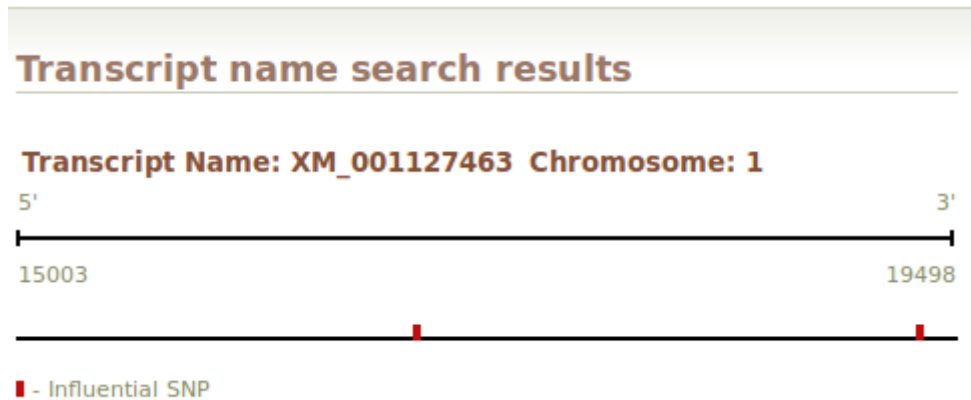
**SiRNA Web Services:** Servidor web de la plataforma que contiene los servicios implementados.

### 4.2 Interpretación gráfica del pictograma para el CU Visualizar SNPs influyentes de un gen

La representación gráfica para mostrar los SNPs en el módulo de visualización se realiza mediante pictogramas, los cuales permiten diseñar la imagen que mejor representa un SNP y una secuencia, tanto de un miRNA como de un gen.

En la Figura 23 se muestra el pictograma para el CU Visualizar SNPs influyentes de un gen. En el mismo la imagen diseñada para representar la secuencia del transcrito es una barra horizontal de color negro donde tendrá asociada en la parte superior una regla indicando la posición inicial y final de la secuencia. Sobre la barra horizontal se muestran los SNPs influyentes con una pestaña roja. El diseño de dichos SNPs en forma de pequeñas pestañas sobre la barra horizontal brinda una medida de la frecuencia de ocurrencia de los SNPs, dando un nivel de la estabilidad del gen para ser utilizado en el proceso de silenciamiento genético.

Las posiciones 5' y 3' viene dada por el estándar de representación de los transcritos. Estas posiciones corresponden a los carbonos que intervienen en los enlaces en la construcción de la cadena. La secuencia del transcrito viene representada desde el carbono en la posición 5' que se encuentra en la composición del azúcar del nucleótido hasta la posición 3' donde se encuentra el grupo hidroxilo y por donde se van añadiendo otros nucleótidos a la cadena. De esta forma la secuencia siempre empieza por un carbono 5' del primer nucleótido y termina por el grupo hidroxilo 3' del último nucleótido.



**Figura 23.** Pantalla final del CU Visualizar SNPs influyentes de un gen

La representación gráfica utilizada para mostrar los SNPs brinda además la posibilidad al usuario de interactuar con la visualización. El especialista al seleccionar la pestaña correspondiente a uno de los SNP representados revela la información del mismo (posición en la secuencia) demostrando la propiedad reversible de la visualización como campo novedoso (Ver Figura 24). En otras palabras la representación gráfica utilizada permite que el especialista regrese a los datos que dieron lugar a la visualización.



**Figura 24.** Propiedades del SNP influyente

### 4.3 Modelo de Prueba

Las pruebas son una medida de la calidad del software. La calidad de un sistema, componente o proceso de desarrollo de software, se obtiene en función del cumplimiento de los requerimientos iniciales especificados por el cliente o usuario final [61].

Además existen muchas variedades de tipos de pruebas que se pueden aplicar [62]. Para el caso que de esta investigación sólo se aplican las pruebas de caja negra.

### Casos de prueba de caja negra

Las pruebas de caja negra se refieren a las pruebas que se llevan a cabo sobre la interfaz del software. Los casos de prueba por tanto tienen como objetivo demostrar que las funcionalidades del software son operativas, que los datos de entrada se aceptan de forma adecuada y que se produce una salida correcta. De esta forma se garantiza la integridad de la información que se almacena y procesa. Se examinan fundamentalmente algunos aspectos del modelo del sistema sin profundizar mucho en la estructura interna del software [63]. A continuación se muestra el caso de prueba correspondiente al CU Visualizar SNPs influyentes de un gen (Ver más en Anexo 4).

#### 4.3.1 Caso de prueba: Visualizar SNPs influyentes de un gen.

##### Flujo central:

- El especialista selecciona la opción “Visualize influential SNP of a gene” para visualizar los SNPs influyentes.

##### El especialista:

1. Introduce el nombre del transcrito del cual desea conocer los SNPs influyentes.
2. Ordena la visualización.

##### El sistema:

3. Valida que el nombre del transcrito sea correcto.
4. Busca los SNPs influyentes en el transcrito.
5. Muestra gráficamente los SNPs influyentes y la posición de estos en la secuencia del transcrito. Además muestra la posición inicial y final del transcrito.

##### Flujo alterno:

- 1.1 El especialista introduce un nombre de un transcrito incorrecto.
  - 1.1.1 El sistema muestra un mensaje de error.

##### Iteración: 3

**Tabla 3.** Resultado del caso de prueba Visualizar SNPs influyentes de un gen



Clases válidas	Resultado esperado	Resultado de la prueba
<Se selecciona la opción “Visualize influential SNPs of a gene” y se introduce el siguiente dato: XM_001127778>	<El sistema verifica que el nombre del transcrito sea correcto, busca los SNPs influyentes en el transcrito y muestra si encuentra alguno los SNPs influyentes así como la posición de estos en la secuencia del transcrito.>	<Satisfactorio>
<Se selecciona la opción “Visualize influential SNPs of a gene” y se introduce el siguiente dato: XR_015303>	<El sistema verifica que el nombre del transcrito sea correcto, busca los SNPs influyentes en el transcrito al no encontrar ninguno muestra el siguiente mensaje: “It doesn’t exist any influential SNPs in this transcript.”.>	<Satisfactorio>
<Se selecciona la opción “Visualize influential SNPs of a gene” y se introduce el siguiente dato: XM_123>	<El sistema verifica que el dato . Check that the letters are capitalized”>	<Satisfactorio>

### 4.3.2 Resumen de las pruebas realizadas

Para la validación del módulo de visualización se realizaron pruebas a nivel de desarrollador de tipo funcionales usando el método de caja negra y dentro de esta la técnica de partición de equivalencia. En el proceso se realizaron 4 casos de pruebas correspondientes a los 4 servicios que se visualizan. Además se realizaron 3 iteraciones donde se encontraron 6 no conformidades (Ver Tabla 4) resueltas en su totalidad.

**Tabla 4.** No Conformidades detectadas

No	Elemento	No Conformidad	Aspecto	Etapas de detección	Signifi
----	----------	----------------	---------	---------------------	---------

## CAPÍTULO 4: IMPLEMENTACIÓN Y PRUEBA.

			Correspondiente		cativo
1	Interfaz	Error de idioma en los títulos de los componentes de la interfaz.	Visualización de los SNPs influyentes en un gen.	Etapa de aplicación de pruebas al módulo.	X
2	Implementación	Ausencia de validación del lado del cliente.	Introducción del ID del miRNA a analizar y el nombre del transcrito.	Etapa de aplicación de pruebas al módulo.	X
3	Interfaz	Ausencia de títulos necesarios para la comprensión por parte del especialista de los datos.	Visualización de los genes diana de un miRNA maduro seleccionado.	Etapa de aplicación de pruebas al módulo.	X
4	Implementación	Desajuste de las gráficas visualizadas, fuera del marco de trabajo.	Visualización de los SNPs influyentes en un gen.	Etapa de aplicación de pruebas al módulo.	X
5	Implementación	Representación incorrecta de los genes que silencia un miRNA maduro seleccionado.	Visualización de los genes diana de un miRNA maduro seleccionado.	Etapa de aplicación de pruebas al módulo.	X
6	Interfaz	Formato incorrecto en los textos de la interfaz.	Visualización de los SNPs influyentes en un miRNA maduro seleccionado	Etapa de aplicación de pruebas al módulo.	X

### 4.4 Conclusiones

Se describieron detalladamente los diagramas de componentes que corresponden al modelo de implementación del módulo para un mejor entendimiento de las funcionalidades a implementar. Se implementó el módulo a partir de los artefactos generados en los flujos de trabajo anteriores y se obtuvo una nueva versión de la aplicación con las siguientes pantallas del sistema (Ver Anexo 5). Se realizaron las pruebas de caja negra donde se pudo comprobar las funcionalidades del módulo. Durante la etapa de prueba se realizaron 3 iteraciones donde se encontraron 6 no conformidades que se le dieron solución en su totalidad.

### CONCLUSIONES

“La gran victoria que hoy parece fácil fue el resultado de pequeñas victorias que pasaron desapercibidas.”

Con la presente investigación se logró la implementación de la visualización de las nuevas funcionalidades para la plataforma alasSiRNA-Design. Este módulo es capaz de brindar servicios de eficiencia no sólo en materia de funcionalidad sino también de confiabilidad y seguridad de la información que se maneja. Las funcionalidades son añadidas a la plataforma alasSiRNA-Design con el propósito de ser utilizadas por los investigadores del CIGB.

Como resultado del trabajo desarrollado se ha obtenido una aplicación capaz de ubicar dónde se encuentra un SNP y si este influye o no en el silenciamiento, brindando un mayor espectro en las investigaciones relacionadas sobre el diseño de siRNAs y por consiguiente sobre el silenciamiento genético. Una adecuada explotación de estas funcionalidades garantiza una herramienta poderosa de avance en los estudios científicos en el campo de la genética, con la búsqueda de tratamientos a determinadas enfermedades.

El módulo se desarrolló siguiendo la metodología OpenUp y utilizando representaciones de todos los artefactos generados por dicha metodología. Mediante el análisis de las funcionalidades, se diseñó e implementó la visualización de los nuevos servicios de SiRNA Web Services teniendo en cuenta la tecnología, herramientas y arquitectura del sistema desarrollado previamente (alasSiRNA-Design).

Al resultado del trabajo se le realizó diferentes pruebas con resultados satisfactorios por lo que se concluye que los objetivos propuestos han sido cumplidos satisfactoriamente. Se incluyen además varias recomendaciones que deben tenerse en cuenta para un futuro seguimiento de este trabajo.

### RECOMENDACIONES

Con el desarrollo del trabajo se han cumplido los objetivos del mismo, a pesar de ello nuevas ideas han surgido, lo cual da la posibilidad de que se llegue a crear una aplicación más funcional y eficiente, que brinde toda clase de servicios sobre la base de las investigaciones en el campo de los estudios genéticos, por tanto recomendamos:

- Desplegar la nueva versión de la aplicación en el CIGB con las nuevas funcionalidades añadidas para su inmediata utilización.
- Realizar la implementación de los algoritmos en la plataforma de tareas distribuidas (T-Arenal) para disminuir el tiempo de ejecución de los mismos.

### REFERENCIAS BIBLIOGRÁFICAS

1. **Valdivia, Alain Joan; Febles, Juan Pedro.** Primer Congreso Internacional de Bioinformática: reflexiones y perspectivas. [En línea] 18/07/2004. [Consultado: 21/11/2011.] [http://bvs.sld.cu/revistas/aci/vol12\\_4\\_04/aci09404.htm](http://bvs.sld.cu/revistas/aci/vol12_4_04/aci09404.htm).
2. **Parrado, Vladir Antonio.** Sistema informático para el diseño de siRNAs mediante el análisis de secuencias de ARN mensajero. Ciudad de La Habana: s.n., Enero, 2010.
3. **E. Bernstein, A.A; Caudy, S.M; Hammond; G.J. Hannon.** Role for a bidentate ribonuclease in the initiation step of RNA interference. Nature, 2001.
4. **Fire, S. Xu; Montgomery, M.K.; Kostas, S.A.; Driver,S.E.; Mello, C.C.** Potent and specific genetic interference by double-stranded RNA in *Caenorhabditis elegans*. NATURE-LONDON-, 1998.
5. **Bustos, Jaimes; Castañeda, C., J. Hernández, Oria; Rendón, E.; Reyes, H.; Romero, I.; A.M.C.** Biología molecular de rotavirus: una mirada a través de la interferencia de RNA. Universitaria, and DF México. 2008.
6. **Lara, Beatriz.** Desarrollo de algoritmos para SiRNA Web Services para el análisis de la influencia de las mutaciones puntuales en los miRNAs. Ciudad de La Habana, Cuba : s.n., Junio, 2011.
7. **Perezleo, Ligeya.** Impacto de la Bioinformática en las ciencias biomédicas. [En línea] ACIMED., 8/06/2003. [Consultado: 21/11/2011.] [http://bvs.sld.cu/revistas/aci/vol11\\_4\\_03/aci07403.htm](http://bvs.sld.cu/revistas/aci/vol11_4_03/aci07403.htm).
8. **Mayorga, Luis S.** Biología de Sistemas y su aplicación en Biología Celular y Molecular. Facultad de Ciencias Médicas-UNCuyo. 2008.
9. **Adjounian, Maria Fernanda; Adjounian, Haroution; Adjounian, Sarkis Horacio.** Silenciamiento de genes mediante RNA interferencia. [En línea] 06/2008. [Consultado: 22/11/2011.] [http://www.scielo.org.ve/scielo.php?script=sci\\_arttext&pid=S0798-02642008000100006&lng=es&nrm=iso](http://www.scielo.org.ve/scielo.php?script=sci_arttext&pid=S0798-02642008000100006&lng=es&nrm=iso).
10. **Pierce, Benjamin A.** Genética: Un Enfoque Conceptual. España: Médica Panamericana, 2009.
11. **Berg, Jeremy Mark; Stryer, Lubert; Tymoczko, John L.** Bioquímica. España: Reverte, 2008.
12. Programa de postgrado de visualización de la información. España. [En línea] 2009 [Consultado: 24/11/2011.] <http://programa-de-postgrado-de-visualizacion-de-la-informacion.html>.

13. **Zhu, Bin; Chen, Hsinchun.** Information Visualization. Annual Review of Information Science and Technology. 2005.
14. **Martig, Sergio; Castro, Silvia.** Visualización de Información. 2008.
15. **Manocivh, Lev.** La visualización de datos como nueva abstracción y antisublime. 2004.
16. **Bajaj, Chandrajit; Krishnamurthy, Bala.** Técnicas de visualización de datos. 1999.
17. **eumed.net.** Pictogramas. [En línea] [Consultado: 24/11/2011.]  
<http://www.eumed.net/libros/2007a/239/3e.htm>.
18. **elpais.com. 2007.** elpais.com. Mapas y Gráficos. [En línea] 25/08/2007. [Consultado: 24/11/2011.]  
<http://lacomunidad.elpais.com/labrandero/2007/8/25/mapas-y-graficos>.
19. **Salas, Rodrigo.** Redes Neuronales Artificiales. Universidad de Valparaíso, Departamento de Computación: s.n., 2007.
20. **Díaz, S. Pértega; Fernández, S. Pita.** Representación gráfica en el Análisis de Datos. [En línea] 02/04/2001. [Consultado: 25/11/2011.] <http://www.fisterra.com/mbe/investiga/graficos/graficos.asp>.
21. **desarrolloweb.com.** Gráficos para representar datos. [En línea] 2002. [Consultado: 24/11/2011.]  
<http://www.desarrolloweb.com/articulos/875.php>.
22. **Trelles, Oswaldo.** Breve lección de Bioinformática. Diversidad, heterogeneidad y dispersión. 2009.
23. **Gene Silencing.** [En línea] 09/02/2010. [Consultado: 25/11/2011.]  
[http://www.proz.com/kudoz/english\\_to\\_spanish/biology\\_tech\\_chemmicro\\_/3688018-gene\\_silencing.html](http://www.proz.com/kudoz/english_to_spanish/biology_tech_chemmicro_/3688018-gene_silencing.html).
24. **Silenciamiento génico.** [En línea] 2007. [Consultado: 25/11/2011.]  
<http://www.argenbio.org/index.php?action=glosario&car=s>.
25. **González Corredor, Rubén.** ARN de interferencia y su importancia en la biomedicina molecular. [En línea] 07/12/2006. [Consultado: 25/11/2011.] <http://alumnatbiogeo.blogspot.com/2006/12/arn-de-interferencia-y-su-importancia.html><http://www.docstoc.com/docs/103702530/mutaciones>.
26. **Vázquez, Evileidys; Bruce, Neima; Baéz, Rosanna; Machado, María.** Polimorfismo genético y respuesta a los medicamentos. [En línea] 2008. [Consultado: 26/11/2011.]  
[http://bvs.sld.cu/revistas/mciego/vol15\\_supl2\\_09/revisiones/r2\\_v15\\_supl209.htm](http://bvs.sld.cu/revistas/mciego/vol15_supl2_09/revisiones/r2_v15_supl209.htm).
27. **Ferrer, Annia; Nazabal, Marcelo; Novoa, Lidia Ines.** UNA APROXIMACIÓN A LA GENÓMICA. [En línea] 2004. [Consultado: 26/11/2011.] <http://elfosscientiae.cigb.edu.cu/PDFs/Muestras/chapter18.pdf>.

28. Los ojos y la secuencia genética. [En línea] 24/04/2010. [Consultado: 26/11/2011.] <http://www.telepinar.icrt.cu/index.php/variadas/8774-los-ojos-y-la-secuencia-genetica>.
29. **Hernández, Gerardo; Mora, Manuel; Patiño, Jesús.** Desarrollo de una plataforma de análisis de datos en bioinformática basada en servicios web. Facultad de Informática Universidad Complutense de Madrid. Curso 2005-2006.
30. **Kappel, Gerti.** Web Engineering: The Discipline of Systematic Development of Web Applications, chapter An Introduction to Web Engineering. John Wiley & Sons Ltd., 2006.
31. jORCA: fácil integración bioinformática de Servicios Web. Departamento de Arquitectura de Computadores de la Universidad de Málaga, 29071 Málaga, España. [En Línea] 03/01/ 2010. [Consultado: 26/11/2011.]
32. **Fernández, Yanileyky; García, Yoendris.** SISTEMA DE ANÁLISIS DE SECUENCIAS MRNA PAR A EL DISEÑO DE SIRNA. Ciudad de la Habana, Cuba: s.n., 2007.
33. **Downey, Tim.** Web Development with Java. Using Hibernate, JSPs and Servlets. Springer. Universidad Internacional de Florida. USA, 2007.
34. **Kader, Naivys; Stable, Antonio.** SISTEMA DE ANÁLISIS DE SECUENCIAS MRNA PAR A EL DISEÑO DE SIRNA. V 1.1q. Ciudad de la Habana, Cuba: s.n., 2008.
- 35 **Chinnici, R.; Moreau, J.J.; Ryman, A.; Weerawarana, S.** Web services description language (WSDL) version 2.0 part 1: Core language. Technical report, Technical report, World Wide Web Consortium, 2007.
36. Tutorial para la Materia de Programación Web. [En línea] JCH, 2010. [Consultado: 27/11/2011.] [http://tuto-mpweb.webcindario.com/2\\_2.html](http://tuto-mpweb.webcindario.com/2_2.html).
37. **Valdés, Damián.** Los diferentes lenguajes de programación para la web. [En línea] 11/2007. [Consultado: 27/11/2011.] <http://www.maestrosdelweb.com/principiantes/los-diferentes-lenguajes-de-programacion-para-la-web/>.
- 38 **Menéndez-Barzanallana, Rafael.** Metodología de desarrollo de software. [En línea] Facultad Derecho UMU, 13/10/2011. [Consultado: 27/11/2011.] <http://www.um.es/docencia/barzana/IAGP/IAGP2-Metodologias-de-desarrollo.html#BM1>.
39. **Castellón, Yanet; Hernández, Rafael.** Pruebas de liberación, su aplicación al sistema informático: Silenciamiento de Genes [siRNA). Ciudad de la Habana, Cuba: s.n., 2009.



40. **Valencia, Luis.** Herramientas IDE - CASE - Presentation Transcript. [En línea] 2011. [Consultado: 27/11/2011.] <http://www.slideshare.net/LuisEduardoPelaez/herramientas-ide-case>.
41. **Sierra, Daniel.** Visual Paradigm For Uml. [En línea] 2007. [Consultado: 27/11/2011.] <http://www.slideshare.net/vanquishdarkenigma/visual-paradigm-for-uml>.
42. *NetBeans* [Consultado: 27/11/2011.] [http://www.netbeans.org/index\\_es.html](http://www.netbeans.org/index_es.html).
43. ¿Qué es un Sistema Gestor de Bases de Datos o SGBD? [En línea] 2009. [Consultado: 30/11/2011.] <http://www.cavsi.com/preguntasrespuestas/que-es-un-sistema-gestor-de-bases-de-datos-o-sgbd/>.
44. **Velneo.** Introducción y características técnicas. [En línea] 2005. [Consultado: 30/11/2011.] [http://forum.velneo.com/Docs/Dossier\\_tec\\_SP.pdf](http://forum.velneo.com/Docs/Dossier_tec_SP.pdf).
45. **Solano, Banny.** ¿Qué es Glassfish? [En línea] 23/08/ 2009. [Consultado: 26 de Noviembre de 2011.] <http://bannysolano.wordpress.com/2009/08/23/%C2%BFque-es-glassfish/>.
46. **Erl, T.** Service-oriented architecture: concepts, technology, and design. Prentice Hall PTR Upper Saddle River, NJ, USA, 2005.
47. **Dapena, M. D. D.** Definición del modelo del negocio y del dominio utilizando Razonamiento Basado en Casos. [En línea] 2006. [Consultado: 18/01/2012.] <http://www.inf.udec.cl/~revista/ediciones/edicion8/Rbc.pdf>.
48. **IEEE.** Standard Glossary of Software Engineering Terminology. [En línea] IEEE. 1990. [Consultado: 04/01/2012.] <http://standards.ieee.org/findstds/standard/610.12-1990.html>.
49. **EVA UCI.** Conferencia #3. Flujo de trabajo de Requerimientos. Curso 2005-2006.
50. Diseño e Implementación de una tienda virtual. [En línea] 2005 [Consultado: 20/01/2012.] <http://www.slideshare.net/dkwolf/proyecto-5-semestre-5032734>.
51. **Garlan, David; Shaw, Mary.** Software Architecture Perspectives on an Emerging Discipline. Prentice Hall, 1996.
52. **Garlan, David; Shaw, Mary.** An Introduction to Software Architecture. 2003. [Consultado: 02/02/2012.] [http://www.cs.cmu.edu/afs/cs/project/able/ftp/intro\\_softarch/intro\\_softarch.pdf](http://www.cs.cmu.edu/afs/cs/project/able/ftp/intro_softarch/intro_softarch.pdf).
53. **Welicki, León.** Patrones y antipatrones. [En línea] 2007. [Consultado: 02/02/2012.] <http://msdn.microsoft.com/es-es/library/bb972251.aspx>.

54. **Gorostiaga, Jesus.** Frameworks PHP y CODEIGNITER ¿Por qué usarlos? [En línea] 2009 [Consultado: 04/03/2012.] <http://es.scribd.com/doc/76348225/Frameworks-Php-y-Codeigniter>.
55. Introducción a la Teoría General de Sistemas. [En línea] 2010 [Consultado: 04/03/2012.] <http://es.scribd.com/doc/17171897/INTRODUCCION-A-LA-TEORIA-GENERAL-DE-SISTEMAS>.
56. **Larman, Craig.** UML y Patrones. Introducción al análisis y diseño orientado a objetos. s. l. Prentice Hall, 2004.
57. **Gamma, Erich; Helm, Richard; Johnson Ralph; Vlissides, John.** Design Patterns: Element of Reusable Object-Oriented Software. Massachusetts, EUA: Addison-Wesley, 1995.
58. EVA UCI. Conferencia #6 Fase de Construcción. Flujo de trabajo de Implementación. Curso 2005-2006.
59. Ingeniería Web UML. [En línea] 2006 [Consultado: 22/04/2012.] <http://es.scribd.com/doc/92602281/Ing-Web-Uml>.
60. **Rumbaugh, James; Jacobson, Ivar; Booch, Grady.** Unified Modeling Language Reference Manual, The Pearson Higher Education, 2004.
61. **Jung, H.W.; Kim, S.G.; Chung, C.S.** Measuring software product quality: A survey of ISO/IEC 9126. IEEE software, 2004.
62. **Deissenboeck, F.; Juergens, E.; Lochmann, K.; Wagner, S.** Software quality models: Purposes, usage scenarios and requirements. In Proc. 7th International Workshop on Software Quality (WoSQ09). IEEE Computer Society, 2009.

## BIBLIOGRAFÍA

1. **Valdivia, Alain Joan; Febles, Juan Pedro.** Primer Congreso Internacional de Bioinformática: reflexiones y perspectivas. [En línea] 18/07/2004. [Consultado: 21/11/2011.] [http://bvs.sld.cu/revistas/aci/vol12\\_4\\_04/aci09404.htm](http://bvs.sld.cu/revistas/aci/vol12_4_04/aci09404.htm).
2. **Parrado, Vladir Antonio.** Sistema informático para el diseño de siRNAs mediante el análisis de secuencias de ARN mensajero. Ciudad de La Habana: s.n., Enero, 2010.
3. **E. Bernstein, A.A; Caudy, S.M; Hammond; G.J. Hannon.** Role for a bidentate ribonuclease in the initiation step of RNA interference. *Nature*, 2001.
4. **Fire, S. Xu; Montgomery, M.K.; Kostas, S.A.; Driver,S.E.; Mello, C.C.** Potent and specific genetic interference by double-stranded RNA in *Caenorhabditis elegans*. *NATURE-LONDON-*, 1998.
5. **Bustos, J.; Castañeda, C., Hernández, O.; Rendón, E.; Reyes, H.; Romero.** Biología molecular de rotavirus: una mirada a través de la interferencia de RNA. Universitaria, and DF México. 2008.
6. **Lara, Beatriz.** Desarrollo de algoritmos para SiRNA Web Services para el análisis de la influencia de las mutaciones puntuales en los miRNAs. Ciudad de La Habana, Cuba : s.n., Junio, 2011.
7. **Pierce, Benjamin A.** Genética: Un Enfoque Conceptual. España: Médica Panamericana, 2009.
8. **Berg, Jeremy Mark; Stryer, Lubert; Tymoczko, John L.** Bioquímica. España: Reverte, 2008.
10. **Post, Frits; Nielson, Gregory; Bonneau, Georges-Pierre.** Visualización de Datos: El Estado del Arte. 2002.
11. Programa de postgrado de visualización de la información. España. [En línea] 2009 [Consultado: 24/11/2011.] <http://programa-de-postgrado-de-visualizacion-de-la-informacion.html>.
12. **Liff, Stewart; Pamela, A. Posey,** ver para creer: ¿Cómo el nuevo arte de la gestión visual puede mejorar el rendimiento en toda la organización, AMACOM, Nueva York. 2007.
13. **Zhu, Bin; Chen, Hsinchun.** Information Visualization. *Annual Review of Information Science and Technology*. 2005.
14. **Dürsteler, Juan C. 2008.** Cartogramas. 2008.
15. **Keller, Peter R; Keller, María.** Señales visuales: visualización de datos útiles. 1993.
16. **Martig, Sergio; Castro, Silvia.** Visualización de Información. 2008.

17. **Gorban, Alexander; Balázs, Kégl; Donald, Wunsch; Andrei, Zinóviev.** Colectores principales para visualización de datos y reducción de la dimensión. LNCSE 58. Springer. 2008.
18. **Manocivh, Lev.** La visualización de datos como nueva abstracción y antisublime. 2004.
20. **Cleveland, William S.** Visualización de datos. Hobart Press. 1993.
21. **Cleveland, William S.** Los elementos de datos de gráficos. Hobart Press. 1994.
22. **Bajaj, Chandrajit; Krishnamurthy, Bala.** Técnicas de visualización de datos. 1999.
23. **eumed.net.** Pictogramas. [En línea] [Consultado: 24/11/2011.] <http://www.eumed.net/libros/2007a/239/3e.htm>.
24. **elpais.com. 2007.** elpais.com. Mapas y Gráficos. [En línea] 25/08/2007. [Consultado: 24/11/2011.] <http://lacomunidad.elpais.com/labrandero/2007/8/25/mapas-y-graficos>.
25. **Salas, Rodrigo.** Redes Neuronales Artificiales. Universidad de Valparaíso, Departamento de Computación: s.n., 2007.
26. **Díaz, S. Pértiga; Fernández, S. Pita.** Representación gráfica en el Análisis de Datos. [En línea] 02/04/2001. [Consultado: 25/11/2011.] <http://www.fisterra.com/mbe/investiga/graficos/graficos.asp>.
27. **desarrolloweb.com.** Gráficos para representar datos. [En línea] 2002. [Consultado: 24/11/2011.] <http://www.desarrolloweb.com/articulos/875.php>.
28. **Cañedo, Rubén; Arencibia, Ricardo.** *Bioinformática : en busca de los secretos moleculares de la vida.* [En línea] INFOMED, 10/12/2004. [Consultado: 24/11/2011.] [http://bvs.sld.cu/revistas/aci/vol12\\_6\\_04/aci02604.htm](http://bvs.sld.cu/revistas/aci/vol12_6_04/aci02604.htm).
29. **Joan, Alain; Febles, Juan Pedro.** *Primer Congreso Internacional de Bioinformática : reflexiones y perspectivas.* [En línea] 18/07/2004. [Consultado: 24/11/2011.] [http://bvs.sld.cu/revistas/aci/vol12\\_4\\_04/aci09404.htm](http://bvs.sld.cu/revistas/aci/vol12_4_04/aci09404.htm).
30. **Trelles, Oswaldo.** Breve lección de Bioinformática. Diversidad, heterogeneidad y dispersión. 2009.
31. **Gene Silencing.** [En línea] 09/02/2010. [Consultado: 25/11/2011.] [http://www.proz.com/kudoz/english\\_to\\_spanish/biology\\_tech\\_chemmicro\\_/3688018-gene\\_silencing.html](http://www.proz.com/kudoz/english_to_spanish/biology_tech_chemmicro_/3688018-gene_silencing.html).
32. **Silenciamiento génico.** [En línea] 2007. [Consultado: 25/11/2011.] <http://www.argenbio.org/index.php?action=glosario&car=s>.
33. **González Corredor, Rubén.** ARN de interferencia y su importancia en la biomedicina molecular. [En línea] 07/12/2006. [Consultado: 25/11/2011.] <http://alumnatbiogeo.blogspot.com/2006/12/arn-de-interferencia-y-su-importancia.html><http://www.docstoc.com/docs/103702530/mutaciones>.

34. **Vázquez, Evileidys; Bruce, Neima; Baéz, Rosanna; Machado, María.** Polimorfismo genético y respuesta a los medicamentos. [En línea] 2008. [Consultado: 26/11/2011.] [http://bvs.sld.cu/revistas/mciego/vol15\\_supl2\\_09/revisiones/r2\\_v15\\_supl209.htm](http://bvs.sld.cu/revistas/mciego/vol15_supl2_09/revisiones/r2_v15_supl209.htm).
35. **Ferrer, Annia; Nazabal, Marcelo; Novoa, Lidia Ines.** UNA APROXIMACIÓN A LA GENÓMICA. [En línea] 2004. [Consultado: 26/11/2011.] <http://elfoscientiae.cigb.edu.cu/PDFs/Muestras/chapter18.pdf>.
36. Los ojos y la secuencia genética. [En línea] 24/04/2010. [Consultado: 26/11/2011.] <http://www.telepinar.icrt.cu/index.php/variadas/8774-los-ojos-y-la-secuencia-genetica>.
37. **Arias, Juan.** Methods and Compositions capable of Causing Post-Transcriptional Silencing of Gene Expression in a Synergic Manner. [En línea] 6/08/2009. [Consultado: 27/11/2011.] <http://www.sumobrain.com/patents/wipo/Methods-compositions-capable-causing-post/WO2009095517.html>.
38. **Guerra, Jorge Jesús Llibre y Arjona, Lorna García.** *ARN interferente: una herramienta y un novedoso mecanismo de regulación génica.* [En línea] 28/04/2008. [Consultado: 27/11/2011.] <http://www.16deabril.sld.cu/rev/235/04.html>.
39. **González, Rubén.** ARN de interferencia y su importancia en la biomedicina molecular . [En línea] 7/12/2006. [Consultado: 28/11/2011.] <http://alumnatbiogeo.blogspot.com/2006/12/arn-de-interferencia-y-su-importancia.html>.
40. **Hernández, Gerardo; Mora, Manuel; Patiño, Jesús.** Desarrollo de una plataforma de análisis de datos en bioinformática basada en servicios web. Facultad de Informática Universidad Complutense de Madrid. Curso 2005-2006.
41. **Kappel, Gerti.** Web Engineering: The Discipline of Systematic Development of Web Applications, chapter An Introduction to Web Engineering. John Wiley & Sons Ltd., 2006.
42. jORCA: fácil integración bioinformática de Servicios Web. Departamento de Arquitectura de Computadores de la Universidad de Málaga, 29071 Málaga, España. [En Línea] 03/01/ 2010. [Consultado: 26/11/2011.]
43. **Fernández, Yanileyky; García, Yoendris.** SISTEMA DE ANÁLISIS DE SECUENCIAS MRNA PAR A EL DISEÑO DE SIRNA. Ciudad de la Habana, Cuba: s.n., 2007.
44. **Downey, Tim.** Web Development with Java. Using Hibernate, JSPs and Servlets. Springer. Universidad Internacional de Florida. USA, 2007.

44. **Kader, Naivys; Stable, Antonio.** SISTEMA DE ANÁLISIS DE SECUENCIAS MRNA PAR A EL DISEÑO DE SIRNA. V 1.1q. Ciudad de la Habana, Cuba: s.n., 2008.
45. **Chinnici, R.; Moreau, J.J.; Ryman, A.; Weerawarana, S.** Web services description language (WSDL) version 2.0 part 1: Core language. Technical report, Technical report, World Wide Web Consortium, 2007.
46. Tutorial para la Materia de Programación Web. [En línea] JCH, 2010. [Consultado: 27/11/2011.] [http://tuto-mpweb.webcindario.com/2\\_2.html](http://tuto-mpweb.webcindario.com/2_2.html).
47. **Valdés, Damián.** Los diferentes lenguajes de programación para la web. [En línea] 11/2007. [Consultado: 27/11/2011.] <http://www.maestrosdelweb.com/principiantes/los-diferentes-lenguajes-de-programacion-para-la-web/>.
48. **Menéndez-Barzanallana, Rafael.** Metodología de desarrollo de software. [En línea] Facultad Derecho UMU, 13/10/2011. [Consultado: 27/11/2011.] <http://www.um.es/docencia/barzana/IAGP/IAGP2-Metodologias-de-desarrollo.html#BM1>.
49. **Castellón, Yanet; Hernández, Rafael.** Pruebas de liberación, su aplicación al sistema informático: Silenciamiento de Genes [siRNA). Ciudad de la Habana, Cuba: s.n., 2009.
50. **Valencia, Luis.** Herramientas IDE - CASE - Presentation Transcript. [En línea] 2011. [Consultado: 27/11/2011.] <http://www.slideshare.net/LuisEduardoPelaez/herramientas-ide-case>.
51. **Sierra, Daniel.** Visual Paradigm For Uml. [En línea] 2007. [Consultado: 27/11/2011.] <http://www.slideshare.net/vanquishdarkenigma/visual-paradigm-for-uml>.
52. *NetBeans* [Consultado: 27/11/2011.] [http://www.netbeans.org/index\\_es.html](http://www.netbeans.org/index_es.html).
53. ¿Qué es un Sistema Gestor de Bases de Datos o SGBD? [En línea] 2009. [Consultado: 30/11/2011.] <http://www.cavsi.com/preguntasrespuestas/que-es-un-sistema-gestor-de-bases-de-datos-o-sgbd/>.
54. **Velneo.** Introducción y características técnicas. [En línea] 2005. [Consultado: 30/11/2011.] [http://forum.velneo.com/Docs/Dossier\\_tec\\_SP.pdf](http://forum.velneo.com/Docs/Dossier_tec_SP.pdf).
55. **Solano, Banny.** ¿Qué es Glassfish? [En línea] 23/08/ 2009. [Consultado: 26 de Noviembre de 2011.] <http://bannysolano.wordpress.com/2009/08/23/%C2%BFque-es-glassfish/>.
56. **Erl, T.** Service-oriented architecture: concepts, technology, and design. Prentice Hall PTR Upper Saddle River, NJ, USA, 2005.
57. **Dapena, M. D. D.** Definición del modelo del negocio y del dominio utilizando Razonamiento Basado en Casos. [En línea] 2006. [Consultado: 18/01/2012.] <http://www.inf.udec.cl/~revista/ediciones/edicion8/Rbc.pdf>.

58. **IEEE**. Standard Glossary of Software Engineering Terminology. [En línea] IEEE. 1990. [Consultado: 04/01/2012.] <http://standards.ieee.org/findstds/standard/610.12-1990.html>.
59. **EVA UCI**. Conferencia #3. Flujo de trabajo de Requerimientos. Curso 2005-2006.
60. Diseño e Implementación de una tienda virtual. [En línea] 2005 [Consultado: 20/01/2012.] <http://www.slideshare.net/dkwolf/proyecto-5-semestre-5032734>.
61. **Garlan, David; Shaw, Mary**. Software Architecture Perspectives on an Emerging Discipline. Prentice Hall, 1996.
62. **Garlan, David; Shaw, Mary**. An Introduction to Software Architecture. 2003. [Consultado: 02/02/2012.] [http://www.cs.cmu.edu/afs/cs/project/able/ftp/intro\\_softarch/intro\\_softarch.pdf](http://www.cs.cmu.edu/afs/cs/project/able/ftp/intro_softarch/intro_softarch.pdf).
63. **Welicki, León**. Patrones y antipatrones. [En línea] 2007. [Consultado: 02/02/2012.] <http://msdn.microsoft.com/es-es/library/bb972251.aspx>.
64. **Gorostiaga, Jesus**. Frameworks. PHP y CODEIGNITER ¿Por qué usarlos? [En línea] 2009 [Consultado: 04/03/2012.] <http://es.scribd.com/doc/76348225/Frameworks-Php-y-Codeigniter>.
65. Introducción a la Teoría General de Sistemas. [En línea] 2010 [Consultado: 04/03/2012.] <http://es.scribd.com/doc/17171897/INTRODUCCION-A-LA-TEORIA-GENERAL-DE-SISTEMAS>.
66. **Larman, Craig**. UML y Patrones. Introducción al análisis y diseño orientado a objetos. s. l. Prentice Hall, 2004.
67. **Gamma, Erich; Helm, Richard; Johnson Ralph; Vlissides, John**. Design Patterns: Element of Reusable Object-Oriented Software. Massachusetts, EUA: Addison-Wesley, 1995.
68. **EVA UCI**. Conferencia #6 Flujo de trabajo de Implementación. Curso 2005-2006.
69. Ingeniería Web UML. [En línea] 2006 [Consultado: 22/04/2012.] <http://es.scribd.com/doc/92602281/Ing-Web-Uml>.
70. **Rumbaugh, James; Jacobson, Ivar; Booch, Grady**. Unified Modeling Language Reference Manual, The Pearson Higher Education, 2004.
71. **Jung, H.W.; Kim, S.G.; Chung, C.S**. Measuring software product quality: A survey of ISO/IEC 9126. IEEE software, 2004.
72. **Deissenboeck, F.; Juergens, E.; Lochmann, K.; Wagner, S**. Software quality models: Purposes, usage scenarios and requirements. In Proc. 7th International Workshop on Software Quality (WoSQ09). IEEE Computer Society, 2009.

## GLOSARIO DE TÉRMINOS

**Aminoácidos:** sustancias cristalinas, casi siempre de sabor dulce; tienen carácter ácido como propiedad básica y actividad óptica; veinte aminoácidos diferentes son los componentes esenciales de las proteínas.

**Bases nitrogenadas:** compuestos orgánicos cíclicos, que incluyen dos o más átomos de nitrógeno. Las bases Adenina (A), Timina (T), Guanina (G) y Citosina (C) se encuentran en el DNA, mientras que en el RNA en lugar de Timina aparece el Uracilo.

**Biochips:** dispositivo a pequeña escala, análogo a un circuito integrado, ensamblado o usado para analizar moléculas orgánicas asociadas con los organismos vivientes.

**Codón:** triplete de bases (A-T-C) que se encuentra en el mRNA.

**Cromosomas:** estructuras físicas de la célula eucariota que portan los genes. Estos cromosomas solo son visibles durante la división celular. Desde el punto de vista de su composición los cromosomas están formados de DNA y proteínas.

**Degradación:** reacciones en que las moléculas orgánicas pierden uno o varios átomos de carbono, o donde las moléculas complejas se descomponen en otras más simples.

**Deleciones:** En genética, tipo especial de anomalía estructural cromosómica que consiste en la pérdida de un fragmento de DNA de un cromosoma. Lleva consigo la desaparición de la información genética contenida en él.

**FASTA:** formato basado texto para representar cualquiera ácido nucleído o secuencias, en que pares bajos o aminoácidos se representan usando códigos single-letter. El formato también permite nombres de la secuencia y comenta para preceder las secuencias. La simplicidad del formato de FASTA hace fácil manipular y analizar secuencias usando las herramientas del proceso de textos e idiomas scripting como Python y Perl.

**Fenotipos:** rasgos particulares y genéticamente heredados de cualquier organismo que lo hacen único e irrepitable en su clase.

**Gen:** unidad de información hereditaria de los seres vivos y una secuencia lineal de nucleótidos en la molécula de DNA o RNA, que contiene la información necesaria para la síntesis de una macromolécula con función celular específica. El gen es considerado como la unidad de almacenamiento de información y unidad de herencia al transmitir esa información a la descendencia.

**Inhibición:** condición de interrupción de algunas de las funciones de un determinado gen para resolver alguna acción importante de la función de los genes en la expresión genética del organismo.



**Moléculas:** partícula más pequeña que presenta todas las propiedades físicas y químicas de una sustancia.

**Nucleótidos:** moléculas orgánicas formadas por la unión covalente de un monosacárido de cinco carbonos (pentosa), una base nitrogenada y un grupo fosfato. Son los monómeros de los ácidos nucleicos (DNA y RNA) en los cuales forman cadenas lineales de miles o millones de nucleótidos, pero también realizan funciones importantes como moléculas libres.

**Plataforma:** un sistema que sirve como base para hacer funcionar determinados módulos de hardware o de software con los que es compatible.

**Proteína:** macromoléculas formadas por cadenas lineales de aminoácidos. Imprescindibles para el crecimiento del organismo y uno de los tres componentes principales de los alimentos.

**Software:** equipamiento lógico o soporte lógico de un sistema informático, comprende el conjunto de los componentes lógicos necesarios que hacen posible la realización de tareas específicas.

**Hardware:** partes tangibles de un sistema informático; sus componentes son: eléctricos, electrónicos, electromecánicos y mecánicos.

**Traducción:** segundo proceso de la síntesis proteica (parte del proceso general de la expresión génica). Dicho proceso tiene cuatro fases: activación, iniciación, elongación y terminación.

**Transcripción:** la transcripción del DNA es el primer proceso de la expresión génica, mediante el cual se transfiere la información contenida en la secuencia del DNA hacia la secuencia de proteína utilizando diversos RNA como intermediarios. Durante la transcripción genética, las secuencias de DNA son copiadas a RNA mediante una enzima llamada RNA polimerasa que sintetiza un mRNA que mantiene la información de la secuencia del DNA.

**Transcrito:** molécula de RNA transcrita a partir de una hebra complementaria de DNA.

**Transposones:** secuencia de DNA que puede moverse de manera autosuficiente a diferentes partes del genoma de una célula. Pueden causar mutaciones y cambio en la cantidad de DNA del genoma.