

Universidad de las Ciencias Informáticas
Facultad 6



Título: “Desarrollo de algoritmos para SiRNA
Web Services para el análisis de la influencia de
las mutaciones puntuales en los miRNAs”

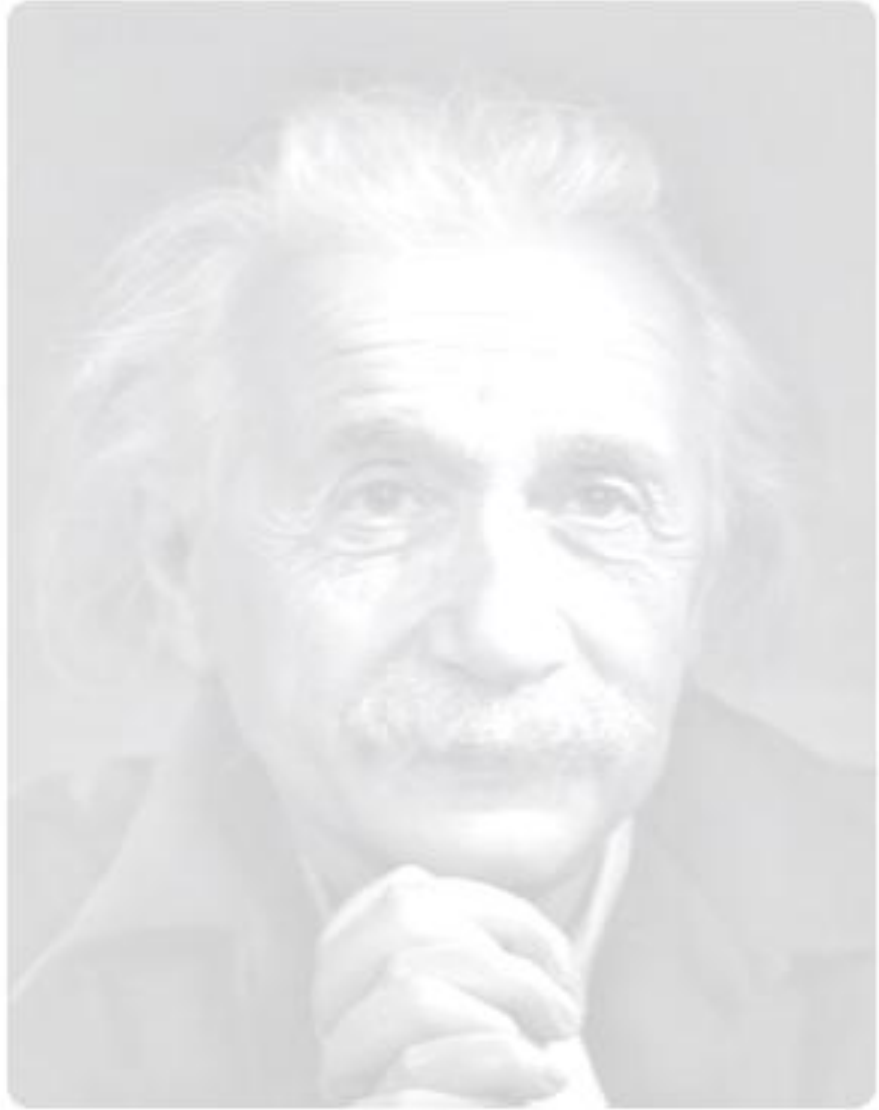
Trabajo de Diploma para optar por el título de
Ingeniero en Ciencias Informáticas

Autor: Beatriz Lara Osorio

Tutor: MSc. Vladir Antonio Parrado Cruz

Co-tutor: Dr. Ricardo Bringas Pérez

Ciudad de la Habana, Cuba, Junio de 2011



Nunca consideres el estudio como una obligación, sino como una oportunidad para penetrar en el bello y maravilloso mundo del saber.

Albert Einstein

DECLARACIÓN DE AUTORÍA

Declaro que soy el único autor de este trabajo y autorizo a la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

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

Beatriz Lara Osorio

Firma del autor

Vladir Antonio Parrado Cruz

Firma del tutor

Ricardo Bringas Pérez

Firma del Co-tutor

DATOS DE CONTACTO

Tutor:

MSc. Vladir Antonio Parrado Cruz.

Universidad de las Ciencias Informáticas, La Habana, Cuba.

Email: vparrado@uci.cu

Co-tutor:

Dr. Ricardo Bringas Pérez.

Centro de Ingeniería Genética y Biotecnología, La Habana, Cuba.

Email: ricardo.bringas@cigb.edu.cu

AGRADECIMIENTOS

Agradezco a mi familia y a todas aquellas personas que siempre desearon que este sueño se hiciera realidad, especialmente a mi madre y a mi tía, las mejores del mundo, por estar siempre pendientes de mí, por apoyar mis decisiones y guiar mis pasos por la vida, porque sin ellas nada de esto hubiera sido posible.

A Ibis, Yosué y Nerón por haberme acogido como parte de la familia.

A mi tutor por guiarme por el camino correcto para poder terminar con éxito este trabajo.

A mi profe Sonia por todo su cariño.

A mis amigos: Dino, Mario, Héctor, Aylenis, Edelis, Betty Gago, Vania, Viviana, Alberto y Rolando, por soportarme todo este tiempo y vivir juntos momentos inolvidables.

A todas las personas que he conocido estos cinco años.

A todos los profesores que influyeron en mi formación como profesional.

A la Revolución Cubana y al Comandante en Jefe por darme la oportunidad de estudiar y graduarme en esta universidad de excelencia.

Muchas gracias a todos.

DEDICATORIA

A mi hermano, mi mamá y mi tía por ser las personas más importantes de mi vida.

RESUMEN

El silenciamiento de genes por RNAs de interferencia naturales (miRNA por sus siglas en inglés) es una poderosa técnica que puede ser utilizada en el área de salud. Actualmente se conocen muchas enfermedades que se relacionan con los miRNAs, por ejemplo, las enfermedades neurodegenerativas humanas como el Alzheimer y el Parkinson. Existen varios sistemas informáticos para la predicción de miRNAs pero hasta el momento no se ha analizado la influencia que tienen las mutaciones puntuales en los mismos ni en los genes que estos silencian. En la Universidad de las Ciencias Informáticas (UCI) se ha trabajado en el desarrollo de un sistema (*alasSiRNA-Design*) que permite diseñar RNA pequeños de interferencia (siRNAs por sus siglas en inglés) a partir de información del genoma humano. Se presentan nuevos algoritmos para el componente *SiRNA Web Service* de la aplicación *alasSiRNA-Design* que permiten determinar la influencia de las mutaciones puntuales tanto en miRNAs como en los genes. Los algoritmos que resuelven este problema fueron diseñados e implementados.

Palabras clave: *alasSiRNA-Design*, miRNAs, mutaciones puntuales.

TABLA DE CONTENIDOS

AGRADECIMIENTOS	I
DEDICATORIA.....	II
RESUMEN	III
INTRODUCCIÓN	1
CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA	4
1.1 La informática y su relación con las ciencias biológicas.....	4
1.2 Silenciamiento de genes	4
1.3 Sistemas existentes para la predicción de miRNAs	7
1.4 Limitaciones de los sistemas existentes	10
1.5 Tecnologías, metodologías y herramientas	10
1.6 Resumen de la tecnología, herramientas y metodologías a utilizar	14
1.7 Conclusiones.....	15
CAPÍTULO 2: ANÁLISIS Y DISEÑO DEL SISTEMA.....	16
2.1 Procesos objeto de automatización y propuesta de sistema.....	16
2.2 Modelo del Dominio.....	17
2.3 Especificación de los requisitos de software	18
2.4 Definición de actores y casos de uso del sistema.....	20
2.5 Casos de usos definidos.....	20
2.6 El diagrama de casos de uso para <i>SiRNA Web Services</i>.....	22
2.7 Especificación de los Casos de Uso.....	22
2.8 Arquitectura de software del sistema.....	25
2.9 Objetivos del diseño.....	26
2.10 Diseño de las clases de <i>SiRNA Web Services</i>.....	26
2.11 Diseño de la base de datos (clases persistentes)	30
2.12 Diagramas de Interacción (Secuencia, Colaboración).....	33
2.13 Diagrama de despliegue.....	33
2.14 Seguridad	34
2.15 Conclusiones.....	34
CAPÍTULO 3: IMPLEMENTACIÓN Y PRUEBA	35
3.1 Diagrama de componentes.....	35

3.2 Diseño de algoritmos	36
3.3 Tratamiento de errores	41
3.4 Modelo de prueba	42
3.5 Conclusiones	46
CONCLUSIONES	47
RECOMENDACIONES	48
REFERENCIAS BIBLIOGRÁFICAS	49
BIBLIOGRAFÍA	51
ANEXOS	54
Anexo 1. Diagramas de secuencias	54
GLOSARIO	56

INTRODUCCIÓN

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. Las cadenas de Ácido Desoxirribonucleico (DNA por sus siglas en inglés) contenidas en los cromosomas son las portadoras de esta información.

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 (traducción).

El mensaje genético se transmite en dos etapas sucesivas, en las que el RNA es un intermediario imprescindible. Existen mecanismos de regulación en todas las etapas de la transcripción y la traducción.

El mecanismo de silenciamiento génico a través de RNA, puede activarse por dos vías: cuando se expresa un RNA de interferencia natural (miRNA del inglés micro interference RNA) o por la aparición de dsRNA (del inglés double-stranded RNA). El dsRNA puede proceder de virus, transposones (y el sistema actúa como defensa) o por inyecciones artificiales realizadas por investigadores.

En esta investigación se hará un mayor énfasis en el silenciamiento de genes por los miRNAs puesto que esta es una poderosa técnica que puede ser utilizada en el área de salud. Actualmente se conocen muchas enfermedades que se relacionan con los miRNAs, por ejemplo el caso de enfermedades neurodegenerativas humanas como el Alzheimer y el Parkinson (1).

En la Universidad de las Ciencias Informáticas (UCI) y en conjunto con el Centro de Ingeniería Genética y Biotecnología (CIGB) se ha trabajado en el desarrollo del sistema *alasSiRNA-Design* que permite diseñar RNAs pequeños de interferencia (siRNAs por sus siglas en inglés) a partir de información del genoma humano (2). Los siRNAs son moléculas de RNA artificiales que se crean con el objetivo de silenciar determinado gen. Además en este sistema se tiene en cuenta las mutaciones puntuales (ver epígrafe 1.2.1) para el diseño de los siRNAs y está compuesto por un Web Service (*SiRNA Web Services*) y la aplicación cliente que consume los servicios.

Hasta el momento no se ha analizado la influencia que tienen las mutaciones puntuales en la acción de los miRNAs y los genes que estos silencian desde el punto de vista computacional.

Actualmente existen varios sistemas informáticos para la predicción de miRNAs como es el caso de *miRanda on-line* (3), *PicTar* (4), *MiRAlign* (5), *miRBase* (6), *TargetScan* (7), *miRanda* (8), *MiRscan* (9). Además se tiene como precedente al sistema *alasSiRNA-Design* (2).

Todos estos sistemas fueron revisados y ninguno tiene en cuenta las mutaciones puntuales para la predicción de los miRNAs ni muestran información alguna al respecto.

Que los investigadores conozcan dónde se encuentran las mutaciones puntuales en los miRNAs y la influencia que pueden tener sobre el efecto del silenciamiento es un tema a resolver y que les puede ser útil en la búsqueda de soluciones a determinadas enfermedades. Es por ello que se hace necesario la implementación de nuevos algoritmos con el fin de añadir estas funcionalidades a *SiRNA Web Services* para su utilización en estudios posteriores. Por tanto el problema de la investigación es ¿Cómo determinar computacionalmente la influencia de las mutaciones puntuales en los miRNAs?

El objeto de estudio es la influencia de las mutaciones puntuales en los miRNAs y el campo de acción las aplicaciones informáticas que relacionen las mutaciones puntuales en la predicción de miRNAs.

Se traza como **objetivo general** de la investigación: Desarrollar algoritmos para el componente *SiRNA Web Services* de la aplicación *alasSiRNA-Design* que permitan determinar la influencia de las mutaciones puntuales en los miRNAs.

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

1. Diseñar los algoritmos necesarios para el análisis de la influencia de las mutaciones puntuales en los miRNAs.
2. Implementar los algoritmos diseñados en el componente *SiRNA Web Services*.
3. Validar dichos algoritmos.

Tareas a cumplir:

- ✓ Diseño del algoritmo para la búsqueda de las mutaciones puntuales en los miRNAs y sus genes dianas.
- ✓ Diseño del algoritmo para la búsqueda de las mutaciones puntuales influyentes presentes en un gen.
- ✓ Diseño del algoritmo para la búsqueda de las mutaciones puntuales que tienen influencia en el poder silenciador de un miRNA.
- ✓ Diseño del algoritmo para crear un archivo con formato Fasta a través de la búsqueda de información de todos los genes del genoma humano.
- ✓ Implementación de los algoritmos diseñados.
- ✓ Realización de pruebas a las nuevas funcionalidades en el Web Services.

El documento estará estructurado de la siguiente forma:

Capítulo 1: Fundamentación teórica: En este capítulo se hace un análisis de la relación de la informática con las ciencias biológicas. Se fundamentan las tecnologías actuales utilizadas y se definen

cuáles herramientas se utilizarán debido a sus características.

Capítulo 2: Análisis y diseño del sistema: En el presente capítulo se definen los requerimientos, tanto funcionales como no funcionales y se obtiene además, el diagrama de casos de uso del sistema. Se ofrece una descripción de la arquitectura del sistema y de los principales objetivos del diseño. Se definen y diseñan diagramas de clases e interacción para el diseño seleccionado. Se muestra el despliegue de la aplicación a través del diagrama correspondiente.

Capítulo 3: Implementación y prueba: En este capítulo se realiza el modelo de implementación a partir de los resultados del flujo de trabajo anteriormente descrito (Diseño). Se obtienen todos los algoritmos necesarios para la implementación de las nuevas funcionalidades. Además se describen las pruebas que se le han realizado al sistema.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

Introducción

En este capítulo se hace un análisis de la relación de la informática con las ciencias biológicas. Se aborda la importancia del desarrollo de nuevos algoritmos para el componente *SiRNA Web Services* de la aplicación *alasSiRNA-Design*. Además se mencionan y se analizan una serie de sistemas informáticos ya desarrollados vinculados con el tema de predicción de miRNAs. Se fundamentan las tecnologías actuales utilizadas para el logro de un Web Service con calidad capaz de proveer nuevas funcionalidades a dicho software y cuáles herramientas se utilizarán debido a sus características.

1.1 La informática y su relación con las ciencias biológicas

La informática en la actualidad da solución a muchos problemas que hace algunos años era imposible tratar. En la rama de las ciencias biológicas su papel ha sido notable, en especial en el área de la biología molecular. Esta interacción condicionó el surgimiento de una disciplina que creó vínculos indisolubles entre la Informática y las ciencias biológicas: la Bioinformática.

La Bioinformática ofrece la posibilidad de comparar y relacionar la información genética con fines deductivos. En los últimos años, se aprecia el crecimiento de una corriente de investigación y desarrollo de nuevas técnicas útiles para la investigación de distintas enfermedades y el diagnóstico clínico, las cuales establecen el camino a seguir por bioinformáticos e investigadores en el futuro. Se puede afirmar que cualquier proceso en cualquier campo biomédico requiere un apoyo informático (10).

La predicción de miRNAs y el análisis de estos es uno de los procesos que no puede prescindir de la utilización de la informática y sus herramientas. Los genomas conocidos comprenden una cantidad considerable de datos, que sólo pueden ser analizados con la asistencia de herramientas especializadas.

1.2 Silenciamiento de genes

El silenciamiento génico es un proceso llevado a cabo en los organismos eucariotas, con diferentes objetivos, dentro de los cuales se destacan la regulación de la expresión y la eliminación y el control de material genético ajeno o externo (virus y transposones) que podría causar un daño a la célula. Existe una clasificación que establece dos grandes categorías, el Silenciamiento Génico Transcripcional (TGS) y el Silenciamiento Génico Post-Transcripcional (PTGS).

La interferencia por RNA es una nueva poderosa herramienta molecular para silenciar genes a nivel post-transcripcional. Es utilizada para la búsqueda de las funciones asociadas a varios genes mediante genómica reversa y como herramienta terapéutica para reducir o bloquear la expresión de genes

alterados o sobre expresados, en terapia genética. La interferencia por RNA proporciona una estrategia fácil y rápida para degradar los RNA mensajeros, mediante la introducción de un RNA pequeño de doble cadena homólogo (siRNA) al RNA mensajero celular de interés, removiéndolo de una manera secuencial específica (11).

1.2.1 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 (12).

1.2.2 SNP

Los polimorfismos de un solo nucleótido (SNP por sus siglas en inglés) constituyen una variación en la secuencia del DNA que afecta a una sola base de una secuencia del genoma. Los SNPs forman hasta el 90% de todas las variaciones genómicas humanas, y aparecen cada 100 a 300 bases en promedio, a lo largo del genoma humano. Los SNPs que se localicen dentro de una secuencia codificante pueden modificar o no, la cadena de aminoácidos que producen, se llama SNP sinónimo a los primeros y SNP no-sinónimo a los segundos. Los SNPs que se encuentren en regiones no codificantes pueden tener consecuencias en el proceso de traducción. Estos 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 (13). Por lo tanto cuando se hable de mutación puntual se estará haciendo referencia a un SNP.

1.2.3 Definición de siRNA

Los siRNAs son moléculas de RNA de doble hebra de 20-21 nucleótidos (nt) perfectamente complementarias, que se originan a partir de un RNA largo de doble hebra (dsRNA). Los dsRNAs pueden ser de origen endógeno, o de origen exógeno. La enzima responsable del procesamiento del dsRNA en moléculas de siRNAs es Dicer. Los siRNAs suprimen la expresión de los genes dianas mediante el corte del mRNA complementario en dos mitades, a través de la interacción de la hebra antisentido del siRNA con el complejo RISC (del inglés RNA-induced silencing complex). Las dos

mitades del mRNA son posteriormente degradadas por la maquinaria celular, lo que conlleva la supresión de la expresión del gen (14).

1.2.4 Definición de miRNA

Una vez descubiertos los siRNAs y la existencia en las células de proteínas que catalizan la degradación del mRNA, los investigadores se preguntaron si los siRNAs también estaban codificados en el genoma, y empezaron a purificar pequeños RNAs (19-25 nt) a partir de diferentes especies animales. Sin embargo, no encontraron siRNAs, sino los denominados miRNAs, que se habían identificado anteriormente de forma independiente.

Los miRNAs son moléculas de RNA transcritas a partir de genes de DNA, pero no son traducidos a proteínas.

La función de los miRNAs está relacionada con la regulación de la expresión génica. De esta forma un miRNA es complementario de una parte de uno o más mRNA. Los miRNAs generalmente inhiben la traducción del mRNA.

Dado el número de dianas potenciales de los miRNAs estos podrían constituir otra capa del circuito regulatorio que existe en las células. Según esto, cualquier desregulación de los miRNAs podría conllevar grandes problemas regulatorios en la célula (15).

En la siguiente figura (Fig. 1) se puede observar detalladamente cómo tiene lugar el proceso de creación de los miRNAs.

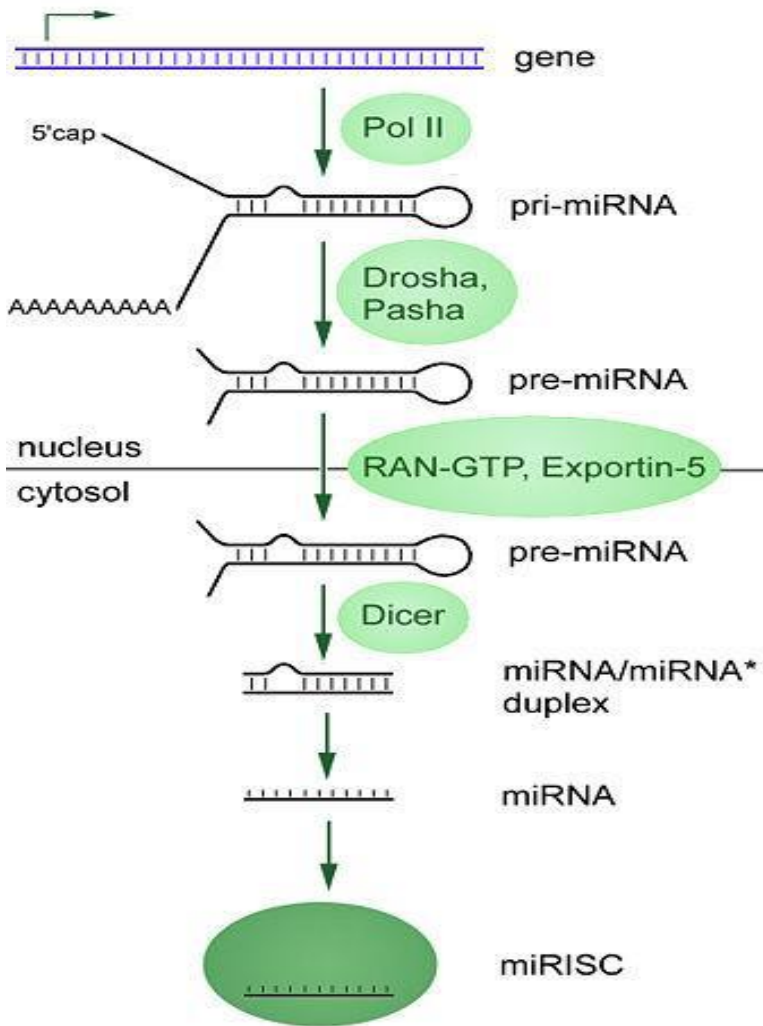


Fig. 1 Proceso de creación de un miRNA. Extraído de (16).

1.3 Sistemas existentes para la predicción de miRNAs

En la actualidad hay varias herramientas publicadas en internet dedicadas a la predicción de miRNAs, de las que se destacan:

1.3.1 *miRanda*

miRanda es un software para identificación de genes dianas de miRNAs. El algoritmo se escribió en el lenguaje de programación C y está disponible como un método de código abierto bajo licencia GPL. Se desarrolló en el Centro de Biología Computacional perteneciente al MSKCC (del inglés Memorial Sloan-Kettering Cancer Center).

El programa funciona cargando dos archivos, uno de miRNAs y otro de presuntos genes dianas y se basa en la realización de apareamientos, los cuales muestra en la pantalla de resultados, así como la energía libre asociada a estos.

Aunque es muy usado, puede resultar tedioso manejarlo (además hay que tener Java instalado en la computadora), por lo que existe una versión disponible en Internet, *miRNA Viewer* (17).

1.3.2 *miRanda on-line*

Se trata del algoritmo *miRanda*, pero manejable en Internet, lo cual puede resultar mucho más cómodo. La página pertenece a sus desarrolladores del Centro de Biología Computacional perteneciente al MSKCC.

Este visualizador permite acceder a las dianas de miRNAs conservadas entre humano y ratón/rata.

Existen dos modos de manejo de la aplicación:

- ✓ Búsqueda por miRNAs
- ✓ Búsqueda por Genes

1.3.3 *PicTar*

Se trata de un programa análogo a los anteriores, pero integrando búsqueda de dianas de miRNAs y búsqueda de miRNAs contra un gen diana en la misma aplicación.

Con el programa se pueden llevar a cabo las siguientes tareas:

- ✓ Búsqueda de dianas de miRNAs
- ✓ Búsqueda de miRNAs con un gen por diana

1.3.4 *MiRAlign*

Es un programa encaminado a la búsqueda y predicción de miRNAs en secuencias génicas dadas.

Está desarrollado por el Laboratorio de Bioinformática y el Departamento de Automatización de la Universidad de Tsinghua (China). Es una nueva aproximación computacional genómica para detectar miRNAs en animales basándose en alineamientos tanto de secuencia como de estructura. Es de fácil manejo.

Permite:

- ✓ Pegar la secuencia problema (de longitud entre 50-300 nucleótidos).
- ✓ Fijar parámetros de restricción.
- ✓ Ajustar el valor límite mínimo de energía libre de la reacción de apareamiento, en los alineamientos.

1.3.5 *miRBase*

Siendo la base de datos de miRNAs por excelencia, *miRBase* ha sustituido al antiguo *miRNARegistry*. Se trata de una base de datos del Instituto *Sanger* (18) de genes de miRNAs identificados en diferentes organismos, y de transcritos de genes que son dianas presuntas de miRNAs.

Desde la página de inicio se accede a sus tres secciones:

- ✓ *miRBase Sequences*: Base de datos de genes de miRNAs.
- ✓ *miRBase Targets*: Base de datos de genes dianas de miRNAs.
- ✓ *miRBase Registry*: Se trata de un formulario para subir a *miRBase* nuevas secuencias de miRNAs. Esto también puede realizarse vía FTP.

1.3.6 *TargetScan*

TargetScan fue el primer software desarrollado para predicción de dianas de miRNAs y está sometido a continuas mejoras y actualizaciones. La última versión es la 3.1, de noviembre de 2006.

Es una aplicación on-line muy usada por los investigadores. Su ventaja es que es restrictivo: Encuentra muchos falsos negativos pero muy pocos falsos positivos, por lo que es bastante fiable.

1.3.7 *MiRscan*

Esta aplicación web predice miRNAs analizando una secuencia horquillada (del inglés *hairpin sequence*).

Para ello, se deben introducir dos secuencias horquilladas. Ambas secuencias serían versiones homólogas de un mismo RNA horquillado en distintas especies. El problema a resolver es si dicho RNA horquillado es un miRNA candidato o no.

El programa compara el alineamiento del par de secuencias introducidas con otros alineamientos de pares de secuencias que son miRNAs, y si hay una homología significativa las secuencias del par en cuestión quedan asignadas también como miRNAs.

1.3.8 *alasSiRNA-Design*

Esta aplicación web se basa en el diseño de siRNAs mediante el análisis de secuencias de mRNA. Entre sus funcionalidades se encuentra: búsqueda de datos de genes y transcritos, análisis de secuencias de mRNA, ensamblaje de secuencias comunes a un gen o específicas de un transcrito de un gen, muestra los SNPs que estén en determinada región de un cromosoma o que coincidan con los miRNAs, así como el cálculo de energía libre. Tiene como ventaja que incorpora al diseño de siRNAs información sobre la variabilidad genética de los individuos. No analiza la influencia de los SNPs en los miRNAs.

1.4 Limitaciones de los sistemas existentes

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 solo 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.

1.5 Tecnologías, metodologías y herramientas

Para el desarrollo de los algoritmos, se realizó un estudio de las posibles herramientas a utilizar en su construcción, teniéndose en cuenta principalmente las tendencias actuales y las novedades de cada una de estas. Se tuvo en cuenta además el uso de estándares y tecnologías que utiliza el centro con el que se desarrolla el proyecto (CIGB) que promueven prácticas orientadas hacia la funcionalidad y la entrega del producto, así como de la implicación directa del cliente.

1.5.1 Lenguajes de desarrollo del software

Para el desarrollo de nuevas funcionalidades en un Web Service se podrían seleccionar numerosos lenguajes de programación usados en el mundo informático como son PHP (19), Python (20), C# (21), Scala (22), 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.

Dada las características de robustez de java, de ser multiplataforma, además de ser software libre, es muy usado en el mundo del desarrollo de aplicaciones, por lo que se estandarizó en el proyecto así como en el centro que trabaja conjunto con el equipo de desarrollo, como lenguaje de programación.

El lenguaje de descripción de servicios Web (WSDL por sus siglas en inglés, Web Service Description Language) es utilizado por el sistema, dado su extensibilidad en la exportación de funcionalidades.

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.

1.5.1.1 Lenguaje JAVA

Java es un lenguaje de programación desarrollado por Sun Microsystems a principios de los años 90. 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 J2EE¹ (23).

1.5.1.2 WSDL

WSDL (del inglés Web Services Description Language) describe la interfaz pública a los servicios Web. 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 (24).

1.5.1.3 Lenguaje de Modelado (UML)

UML (del inglés Unified Modeling Language) es un lenguaje que permite modelar, construir y documentar los elementos que forman un sistema software orientado a objetos. Se ha convertido en el estándar de facto de la industria, debido a que ha sido impulsado por los siguientes autores: Grady Booch, Ivar Jacobson y Jim Rumbaugh. Estos autores fueron contratados por la empresa Rational Software Co. para crear una notación unificada en la que se pueda basar la construcción de sus herramientas CASE (del inglés Computer Aided Software Engineering) (25). EL UML es una de las herramientas más utilizadas en el mundo para el desarrollo de software, esta particularidad de uso se debe a que permite a los desarrolladores de sistemas, generar diseños conforme a sus ideas de forma fácil y entendible.

1.5.2 Herramientas de desarrollo

Existen muchas herramientas para desarrollar software, las cuales fundamentalmente se agrupan en: herramientas CASE y herramientas IDE (del inglés Integrated Development Environment).

En cuanto a herramientas CASE se utiliza *Visual Paradigm* porque *Rational Rose* posee plataforma independiente y ser multiplataforma es una ventaja considerable para el desarrollo del proyecto.

¹ Java Enterprise Edition es una plataforma de programación para desarrollar y ejecutar software de aplicaciones en Lenguaje de programación Java

En lo referente a las herramientas IDE se utiliza *NetBeans* porque tiene un buen soporte de aplicaciones web (.war, jsp y servlets) del cual *Eclipse* carece, los plugins de *Eclipse* además no son ni tan potentes, ni tan sencillos como el módulo que en *NetBeans* viene preinstalado (26).

1.5.2.1 *Visual Paradigm*

Se puede definir a las herramientas CASE como 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 proyecto.

Visual Paradigm es una herramienta que utiliza UML y soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue (27).

1.5.2.2 *NetBeans 6.9*

El equipo de desarrollo de *NetBeans* ha anunciado la disponibilidad de la versión 6.9 del popular IDE con el que se puede escribir, compilar, depurar y ejecutar aplicaciones de escritorio y/o, web con múltiples lenguajes de programación.

Entre las mejoras y novedades de *NetBeans 6.9* se destacan:

- ✓ JavaFX Composer, una herramienta visual para construir aplicaciones JavaFX.
- ✓ Soporte de la plataforma OSGI.
- ✓ Mejoras en el soporte de los frameworks JavaFX SDK 1.3, PHP Zend, Ruby onRails 3.0 y Spring Framework 3.0.
- ✓ Mejoras en los editores y debuggers Java.

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) (28).

1.5.3 Proceso de desarrollo de software

Para desarrollar un sistema con calidad se hace necesario apoyarse en una metodología de trabajo ya conocida y después de un análisis realizado se determinó utilizar la metodología *OpenUp*. Esta excluye la mayoría de las partes opcionales de *RUP* (del inglés Rational Unified Process) 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.

1.5.3.1 *OpenUp*

OpenUp es un marco de trabajo para procesos de desarrollo de software. Fue liberado por el EPF (del inglés Eclipse Process Framework). *OpenUp* se construyó sobre una donación realizada por IBM

(Basic Unified Process). Fue entregada a Eclipse a fines de 2005 y renombrado como *OpenUp* en 2006. Preserva la esencia del Proceso Unificado. Está pensado para proyectos pequeños (29).

1.5.4 Gestor de Base de Datos

En cualquier proyecto informático para una selección de la base de datos a utilizar se requiere de un análisis profundo sobre cuál gestor utilizar. En *alasSiRNA-Design* 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 cumplir con sus peticiones. Además de que es uno de los gestores que se estableció en el departamento de Bioinformática (30). Implementa el uso de subconsultas, haciendo su funcionamiento mucho más eficaz, y ofreciendo soluciones en campos en las que *MySQL* no podría (31).

1.5.4.1 Gestor de base de datos *PostgreSQL*

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 (32). *PostgreSQL* aproxima los datos a un modelo objeto-relacional, y es capaz de manejar complejas rutinas y reglas.

1.5.5 Servidor de Aplicaciones

Las funciones de todos los servidores de aplicaciones son similares pero no todos los servidores de aplicaciones brindan sus servicios con la misma velocidad, seguridad y confiabilidad. Como los servidores en los cuales estará desplegada finalmente la aplicación, son de pocas prestaciones, se selecciona el *Tomcat 7.0* como servidor de aplicaciones, debido a que el mismo necesita menos potencial de hardware que el *JBoss* para funcionar en óptimas condiciones. Además dicha selección no afecta el rendimiento del sistema para la cantidad de usuarios para lo cual está concebido.

1.5.5.1 *Tomcat 7.0*

Tomcat 7.0 es un contenedor de Servlets con un entorno JSP. *Tomcat* puede funcionar como servidor web por sí mismo. Es usado mundialmente como servidor web autónomo en entornos con alto nivel de tráfico y alta disponibilidad (33). *Tomcat 7.0* es la última versión de este popular servidor web java actualmente desarrollado por la Apache Software Foundation.

Este nuevo lanzamiento proporciona soporte por primera vez para *Servlet 3.0*, *JSP 2.2* y lenguaje *EL 2.2*, mejoras para detectar y prevenir fugas de memoria en las aplicaciones webs, soporte nativo de

funciones y herramientas que simplifican la programación de aplicaciones web complejas y soporte para la inclusión de contenidos externos directamente en una aplicación web (34).

1.5.6 Servicio Web

Un servicio Web (en inglés Web service) es un conjunto de protocolos y estándares que sirven para intercambiar datos entre aplicaciones. Distintas aplicaciones de software desarrolladas en lenguajes de programación diferentes, y ejecutadas sobre cualquier plataforma, pueden utilizar los servicios web para intercambiar datos en redes de ordenadores como Internet. La interoperabilidad se consigue mediante la adopción de estándares abiertos. Las organizaciones OASIS y W3C (35) son los comités responsables de la arquitectura y reglamentación de los servicios Web. Para mejorar la interoperabilidad entre distintas implementaciones de servicios Web se ha creado el organismo WS-I (Web Services Interoperability por sus siglas en inglés), encargado de desarrollar diversos perfiles para definir de manera más exhaustiva estos estándares (25).

1.5.7 SOA

La Arquitectura Orientada a Servicios (en inglés Service Oriented Architecture o SOA), es un concepto de arquitectura de software que 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 (25). La misma está formada por tres partes: un proveedor, un intermediario y un cliente que no presentan ningún acoplamiento entre ellos. El proveedor ofrece un servicio determinado y que el cliente no tiene por qué conocer directamente. El cliente aprende cómo utilizar el servicio a partir de la información que le ofrece el intermediario que normalmente simplifica el uso de dicho servicio. El cliente sólo sabe cómo utilizar el servicio, es decir, como enviar y recibir datos pero no conoce ningún detalle de su implementación interna.

1.6 Resumen de la tecnología, herramientas y metodologías a utilizar

Como resultado del estudio de las tecnologías y herramientas de desarrollo a utilizar se ha llegado a la conclusión de que el sistema se desarrollará usando como lenguaje de programación Java, y como gestor de bases de datos *PostgreSQL 8.2*. Se determinó además hacer uso de *UML* por su estrecha integración con *OpenUp*, metodología adoptada. Como herramienta de modelado se utiliza *Visual Paradigm*. Como entorno de desarrollo integrado se decidió que debe ser *NetBeans 6.9* y como servidor de aplicaciones se adoptó *el Tomcat 7.0*. Se determinó realizar los web services para seguir una arquitectura orientada a servicios (SOA). El software *miRanda* se utilizará para hacer búsquedas

de los genes dianas de miRNAs debido a su utilización local en una estación de trabajo y además porque se cuenta con el código del mismo. Además puede ser ejecutado desde un método java, pasándole las direcciones de ambos ficheros de entrada, lo que permitiría utilizar el resultado para posteriores análisis.

1.7 Conclusiones

En el presente capítulo se analizaron las tecnologías, herramientas y metodologías con el propósito de poder definir cuáles utilizar para el desarrollo de la aplicación. Se profundizó en algunos conceptos necesarios para la comprensión de la solución de este trabajo y la necesidad de la creación de nuevos algoritmos para el Web Services de la aplicación *alasSiRNA-Design* dado el análisis de los sistemas actualmente utilizados para la predicción de miRNAs. Se detallaron limitantes de estos sistemas y por qué se realizan estas nuevas funcionalidades.

CAPÍTULO 2: ANÁLISIS Y DISEÑO DEL SISTEMA

Introducción

En el presente capítulo se definen los requerimientos, tanto funcionales como del sistema y se obtienen además, como resultado, el diagrama de casos de uso del sistema y las descripciones de los correspondientes casos de uso. Se ofrece además una descripción de la arquitectura del sistema y de los principales objetivos del diseño. Se definen y diseñan diagramas de clases e interacción para el diseño seleccionado. Se muestra el despliegue de la aplicación a través del diagrama correspondiente.

2.1 Procesos objeto de automatización y propuesta de sistema

En el proceso de estudio e investigaciones sobre el análisis de la influencia de las mutaciones puntuales en los miRNAs se desea automatizar todo lo referente a la información de los miRNAs así como la de los SNPs presentes en estos miRNAs y sus genes dianas.

Estas nuevas funcionalidades se centrarán principalmente en el análisis de la influencia de las mutaciones puntuales en el poder silenciador de los miRNAs. Incluirá los algoritmos desarrollados para este análisis como es el caso de: búsqueda de las mutaciones puntuales en los miRNAs y sus genes dianas, búsqueda de las mutaciones puntuales influyentes en un gen, búsqueda de las mutaciones puntuales que tienen influencia en el poder silenciador de un miRNA y el algoritmo para crear un archivo con formato Fasta a través de la búsqueda de información de todos los genes del genoma humano.

2.1.1 Descripción general de la propuesta de sistema

Con el objetivo de mejorar la calidad de las investigaciones aplicadas y el desarrollo de los trabajos implementados en el CIGB, se ha decidido desarrollar nuevos algoritmos para el análisis de la influencia de las mutaciones puntuales en los miRNAs que brinda un mayor espectro en las investigaciones relacionadas al silenciamiento de genes. Los SNPs influyen en que un miRNA pueda o no silenciar determinados genes, 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). Es importante entonces que los especialistas puedan tener medidas para realizar sus investigaciones. Estas funcionalidades permitirán conocer dónde se encuentra un SNP en un miRNA y en un gen determinado y además si esa mutación influye o no en el silenciamiento.

Teniendo en cuenta estas características y el sistema informático desarrollado previamente (*alasSiRNA-Design*) se incorporarán las nuevas funcionalidades en el web services (*SiRNA Web Services*) y se mantiene por tanto la misma arquitectura.

2.1.2 Análisis comparativo de otras soluciones existentes con la propuesta

Teniendo en cuenta otras soluciones existentes que se encuentran en internet, se puede decir, que la propuesta que brinda esta investigación es una solución viable al problema planteado por el especialista por las siguientes razones:

1. Mantiene la independencia de Internet lograda con la aplicación anterior lo cual favorece el avance en las investigaciones.
2. Brinda nuevas funcionalidades a partir de recomendaciones y necesidades de los investigadores.
3. Brinda servicios web para exportar las funcionalidades a cualquier investigador, tanto en el CIGB como en el resto del mundo.

2.2 Modelo del Dominio

Un modelo del dominio captura los tipos más importantes de objetos en el contexto del sistema. Los objetos del dominio representan los eventos que suceden en el entorno en el que trabaja el sistema. La modelación del dominio tiene como objetivo fundamental la comprensión y descripción de las clases más importantes en el sistema. Debido a que se hace difícil encontrar una estructura de los procesos de negocios que tienen que ver con el objeto de estudio, y haciéndose necesario poder entender el funcionamiento de la aplicación, se necesita describir mediante una serie de conceptos, entidades y sus relaciones, agrupados en un modelo del dominio para el logro de un fácil entendimiento de la misma.

Definición de las entidades y los conceptos principales.

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

alasSiRNA-Design: Aplicación existente para el diseño de siRNAs.

Genes: Un gen es una secuencia lineal organizada de nucleótidos en la molécula de DNA (o RNA en el caso de algunos virus).

miRNAs: Los miRNAs son moléculas de RNA transcritas a partir de genes de DNA, pero no son traducidos a proteínas, su función es regular la acción de otros genes.

Mutaciones Puntuales: Cubren una variedad de mutaciones desde cambios genuinos en un par de bases hasta pequeñas deleciones e inserciones.

Silenciamiento de genes: El silenciamiento génico es un proceso llevado a cabo en los organismos eucariotas, con diferentes objetivos, dentro de los cuales se destacan la regulación de la expresión y la eliminación y el control de material genético que podría causar un daño a la célula.

Diagrama Modelo del Dominio.

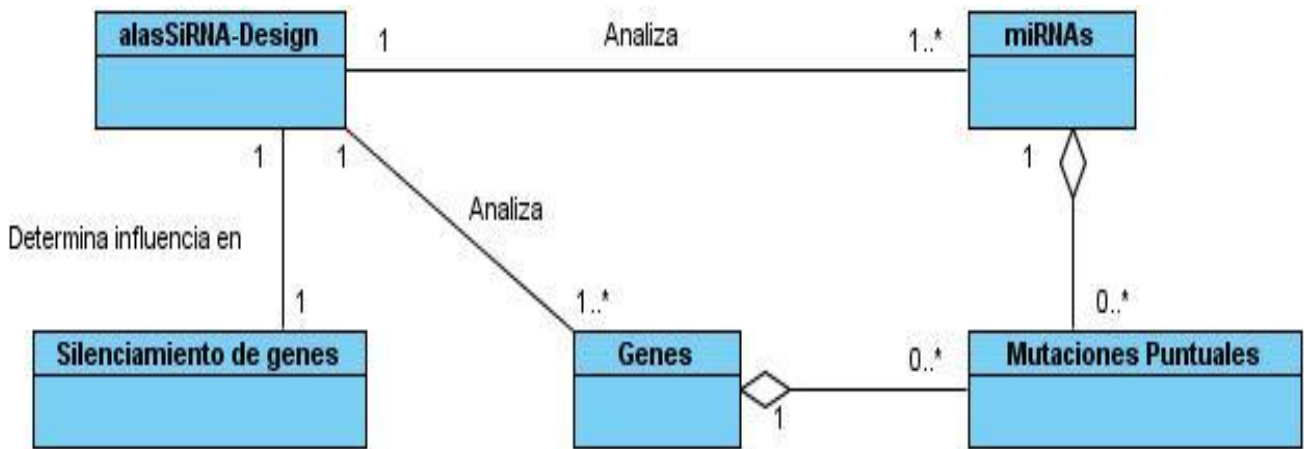


Fig. 2 Modelo del dominio

2.3 Especificación de los requisitos de software

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

2.3.1 Requerimientos Funcionales

Los requerimientos funcionales son capacidades o condiciones que el sistema debe cumplir. Permiten expresar una especificación más detallada de las responsabilidades del sistema en cuestión:

RF 1: Buscar SNPs presentes en miRNAs.

RF 2: Buscar genes blancos de un miRNA.

RF 3: Buscar información de los genes en el genoma humano.

RF 4: Crear archivo Fasta con información de los genes del genoma humano.

RF 5: Buscar SNPs influyentes de un gen.

RF 6: Buscar SNPs influyentes en el silenciamiento en un miRNA.

2.3.2 Requerimientos no funcionales

Los requerimientos no funcionales son propiedades o cualidades que el producto debe tener. Representan las características del mismo.

Usabilidad:

Las nuevas funcionalidades de *SiRNA Web Services* brindarán la posibilidad de ser usadas por cualquier aplicación cliente que necesite consumir estos servicios, solo es necesario contar con conocimientos especializados en biología para poder entender los resultados brindados por la aplicación.

Rendimiento:

Para una mayor velocidad de procesamiento de la información se realiza la validación de los datos de entrada y la manipulación de eventos en el servidor. De esta forma se logra un mayor aprovechamiento de los recursos.

Portabilidad:

El sistema está concebido para que sea multiplataforma.

Software:

Se debe disponer de Windows XP (o versiones superiores) o Linux para la instalación del Web Service. La base de datos es independiente de la aplicación.

Hardware:

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

Para el servidor de aplicaciones:

Procesador Pentium 3 o superior.

512 MB de RAM.

Para el servidor del Web Services:

Procesador Pentium 3 o superior.

1 GB de RAM.

Un mínimo de 4 GB de espacio disponible en disco duro.

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.

2.4 Definición de actores y casos de uso del sistema

Un actor es una entidad externa al sistema que interactúa con él. Siempre se beneficia de la realización de un caso de uso. Estimula el sistema con eventos de entrada y salida. Es un rol de un usuario que puede intercambiar información o puede ser un recipiente pasivo de información. Representa un ser humano, una máquina o un software que interactúa con el sistema.

Tabla.1 Actores del Sistema

Nombre del actor	Descripción
Aplicación_Cliente	Representa el sistema que accederá a los servicios de <i>SiRNA Web Services</i> y que podrá obtener todas sus funcionalidades.

2.5 Casos de usos 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.

Los casos de uso definidos para *SiRNA Web Services* son los siguientes:

1. Buscar información de miRNAs.
2. Crear archivo Fasta con información del genoma humano.
3. Buscar SNPs influyentes de un gen.
4. Buscar SNPs influyentes de un miRNA.

Tabla.2 CU Buscar información de miRNAs

CU 1	Buscar información de miRNAs
Actor	Aplicación_Cliente
Descripción	Obtiene toda la información referente a un determinado miRNA.
Referencia	RF 1, RF 2.

Tabla.3 CU Crear archivo Fasta con información del genoma humano

CU 2	Crear archivo Fasta con información del genoma humano
Actor	Aplicación_Cliente
Descripción	Obtiene la información referente a los genes en el genoma humano y crea un archivo con formato Fasta con dicha información.
Referencia	RF 3, RF 4.

Tabla.4 CU Buscar SNPs influyentes de un gen

CU 3	Buscar SNPs influyentes de un gen
Actor	Aplicación_Cliente
Descripción	Obtiene aquellos SNPs que tienen influencia en un determinado gen.
Referencia	RF 5.

Tabla.5 CU Buscar SNPs influyentes de un miRNA

CU 4	Buscar SNPs influyentes de un miRNA
Actor	Aplicación_Cliente
Descripción	Obtiene aquellos SNPs que tienen influencia en el poder silenciador de un miRNA determinado.
Referencia	RF 6.

2.6 El diagrama de casos de uso para *SiRNA Web Services*

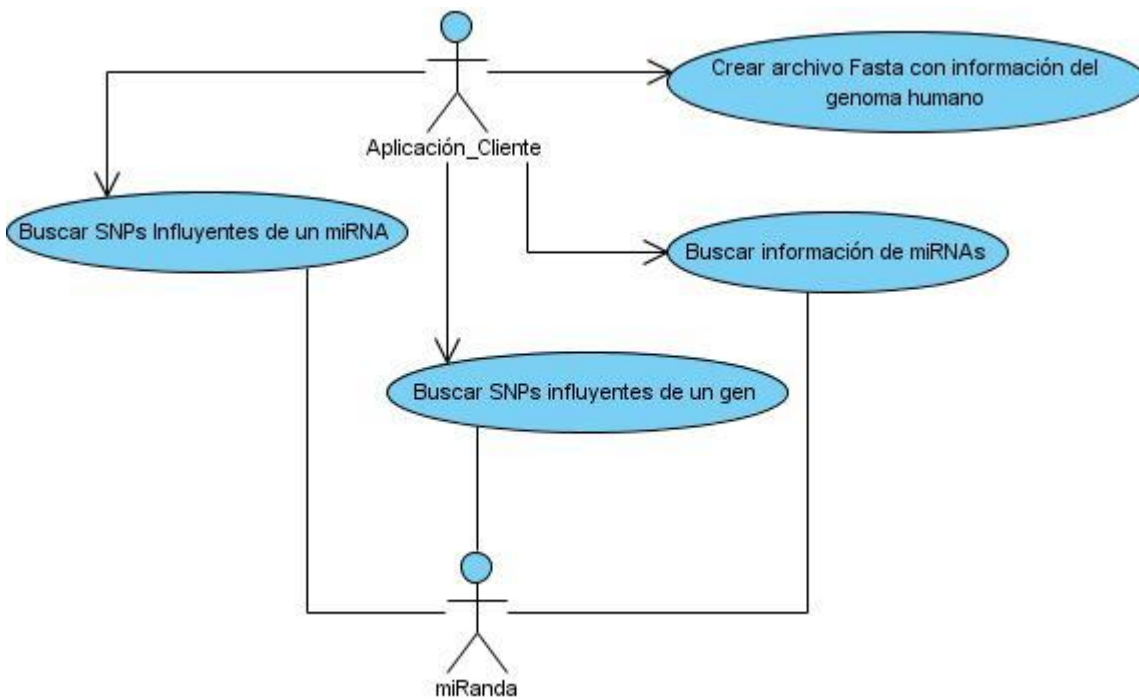


Fig. 3 Diagrama de casos de uso del sistema *SiRNA Web Services*.

2.7 Especificación de los Casos de Uso

CU Buscar información de miRNAs. *SiRNA Web Services*

Tabla.6 Descripción del CU Buscar información de miRNAs

Caso de Uso:	Buscar información de miRNAs
Actores:	Aplicación_Cliente.
Propósito:	Buscar toda la información referente a los miRNAs.
Resumen:	El caso de uso se inicia cuando la Aplicación_Cliente accede a las funcionalidades del web services para conocer información de los miRNAs.
Referencia:	RF 1, RF 2.
Precondiciones:	
Poscondiciones:	
Flujo Normal de Eventos	

Acción del Actor	Respuesta del Sistema
1) La Aplicación_Cliente hace una petición al web services para buscar información de un miRNA pasando como parámetro el ID del miRNA.	2) Busca miRNAs maduros contenidos en dicho miRNA. 3) Busca SNPs presentes en el miRNA maduro seleccionado. 4) Busca genes blancos del miRNA maduro seleccionado.
Prioridad:	Secundario.

CU Crear archivo Fasta con información del genoma humano. *SiRNA Web Services*

Tabla.7 Descripción del CU Crear archivo Fasta con información del genoma humano

Caso de Uso:	Crear archivo Fasta con información del genoma humano
Actores:	Aplicación_Cliente.
Propósito:	Buscar información de todos los genes en el genoma humano y crear un archivo en formato Fasta con esta información.
Resumen:	El caso de uso se inicia cuando la Aplicación_Cliente accede a las funcionalidades del web services para crear un archivo Fasta con información de todos los genes en el genoma humano.
Referencia:	RF 3, RF 4.
Precondiciones:	
Poscondiciones:	
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1) La Aplicación_Cliente hace una petición al web services para crear un archivo Fasta con información de todos los genes en el genoma humano.	2) Busca información de los genes en el genoma humano. 3) Crea un archivo con formato Fasta con la información obtenida.
Prioridad:	Secundario.

CU Buscar SNPs influyentes de un gen. *SiRNA Web Services*

Tabla.8 Descripción del CU Buscar SNPs influyentes de un gen

Caso de Uso:	Buscar SNPs influyentes de un gen
Actores:	Aplicación_Cliente.
Propósito:	Buscar aquellos SNPs que sean influyentes en un determinado gen.
Resumen:	El caso de uso se inicia cuando la Aplicación_Cliente accede a las funcionalidades del web services para conocer los SNPs influyentes de un gen determinado.
Referencia:	RF 5.
Precondiciones:	
Poscondiciones:	
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1) La Aplicación_Cliente hace una petición al web services para buscar los SNPs influyentes en un gen determinado pasando como parámetro el ID del gen.	2) Busca los SNPs influyentes del gen seleccionado.
Prioridad:	Crítico.

CU Buscar SNPs influyentes de un miRNA. *SiRNA Web Services*

Tabla.9 Descripción del CU Buscar SNPs influyentes de un miRNA

Caso de Uso:	Buscar SNPs influyentes de un miRNA
Actores:	Aplicación_Cliente.
Propósito:	Buscar aquellos SNPs que sean influyentes en el poder silenciador de un determinado miRNA.
Resumen:	El caso de uso se inicia cuando la Aplicación_Cliente accede a las funcionalidades del web services para conocer los SNPs influyentes de un miRNA determinado.
Referencia:	RF 6.

Precondiciones:	
Poscondiciones:	
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1) La Aplicación_Cliente hace una petición al web services para buscar los SNPs influyentes en un gen determinado pasando como parámetro el ID del miRNA.	2) Busca los SNPs influyentes de los miRNAs maduros contenidos en el miRNA seleccionado.
Prioridad:	Crítico.

2.8 Arquitectura de software del sistema

La Arquitectura del Software es la organización fundamental de un sistema. Este sistema está formado por:

- ✓ componentes
- ✓ las relaciones entre ellos
- ✓ el contexto en el que se implantarán
- ✓ los principios que orientan su diseño y evolución.

Para que un sistema sea extensible y reutilizable, su arquitectura tiene que estar diseñada en base a ello. Para tal propósito y buen uso de la misma se utilizan los patrones arquitectónicos.

Patrón de arquitectura: Los patrones arquitectónicos expresan el esquema de organización estructural fundamental para sistemas de software. Provee un conjunto de subsistemas predefinidos, especifica sus responsabilidades e incluye reglas y pautas para la organización de las relaciones entre ellos.

La aplicación informática se diseña con una arquitectura orientada a servicios. Este tipo de arquitectura ha surgido como la mejor manera de hacer más con menos recursos. Promete hacer la reutilización y la integración mucho más fáciles, ayudando a reducir el tiempo de desarrollo y aumentando la agilidad organizacional.

Vista lógica del sistema

La vista lógica 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. El siguiente diagrama tiene como objetivo principal representar las clases de la aplicación. Se pueden especificar cuatro subsistemas. El subsistema Sistemas de Diseño de siRNA se refiere a las aplicaciones que se conecten con el Web Services. De esta representación abstracta hereda *alasSiRNA-Design* (subsistema) que será el primer cliente de *SiRNA Web Services* (subsistema) que a su vez utiliza funcionalidades del software *miRanda* (subsistema) (fig.5). En esta representación solo se harán cambios en el subsistema *SiRNA Web Services*, al cual le serán añadidas las nuevas funcionalidades.

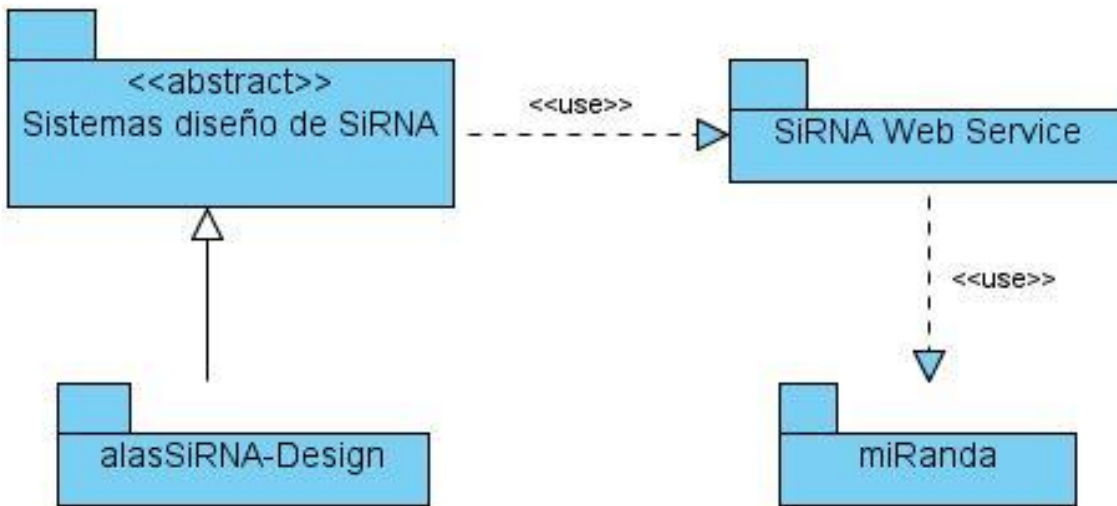


Fig. 4 Vista lógica del sistema.

2.9 Objetivos del diseño

El diseño tiene como principales objetivos comprender detalladamente los requisitos funcionales y no funcionales, sistemas operativos, tecnologías de distribución, restricciones relacionadas con el lenguaje de programación. Además, el diseño crea un punto de partida para las actividades de implementación que siguen.

2.10 Diseño de las clases de *SiRNA Web Services*

Los servicios Web, no son aplicaciones con una interfaz gráfica con la que los usuarios puedan interactuar, sino que son software accesible a través de Internet o en redes privadas que usen tecnologías Internet por otras aplicaciones.

Los siguientes diagramas de clases no siguen un patrón de arquitectura conocido pero cumplen con los requisitos de seguridad, confiabilidad y disponibilidad que debe tener todo web services.

Caso de Uso Buscar SNPs influyentes de un gen

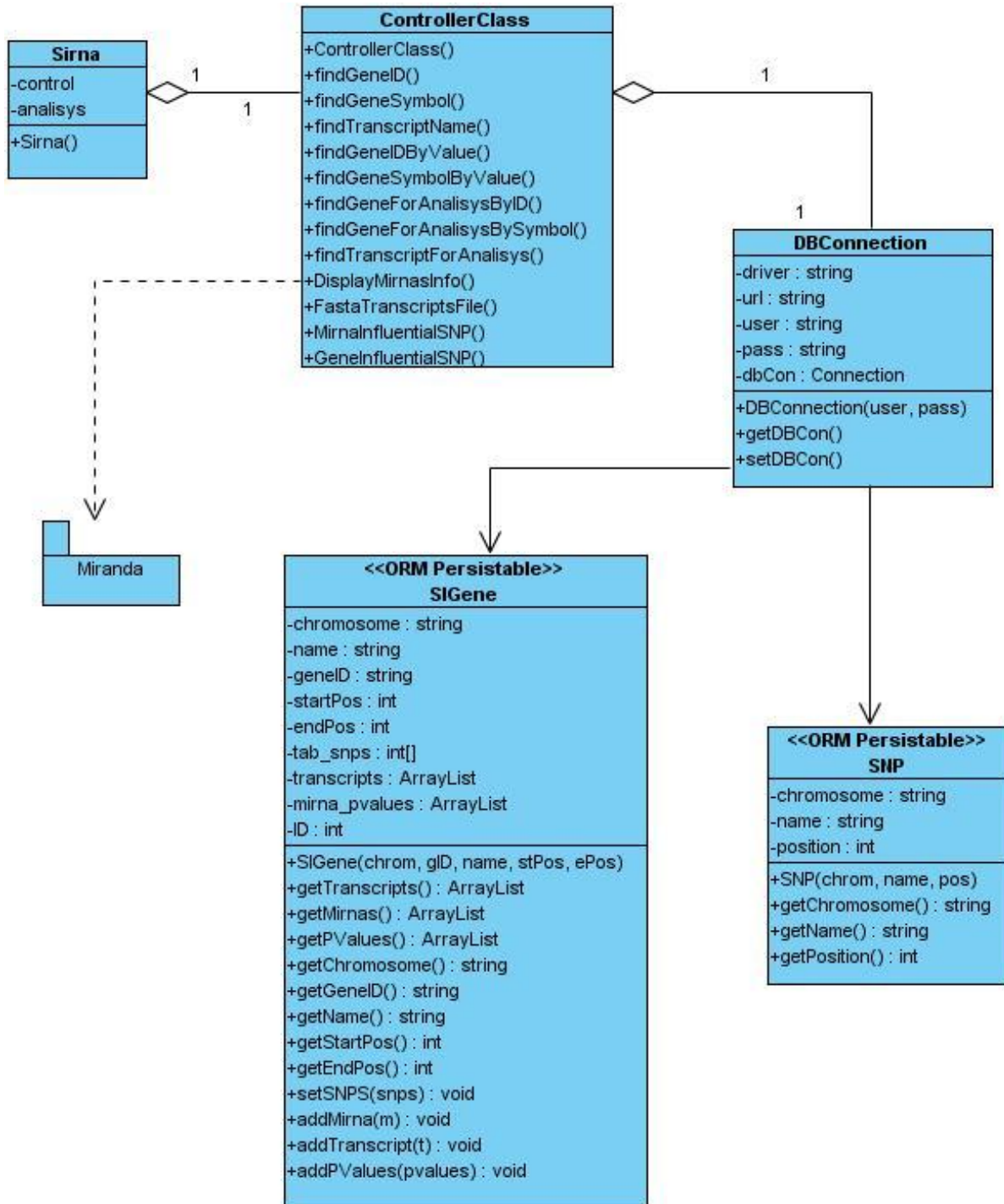


Fig. 5 Diagrama de clases del diseño del CU Buscar SNPs influyentes de un gen.

Caso de Uso Buscar SNPs influyentes de un miRNA

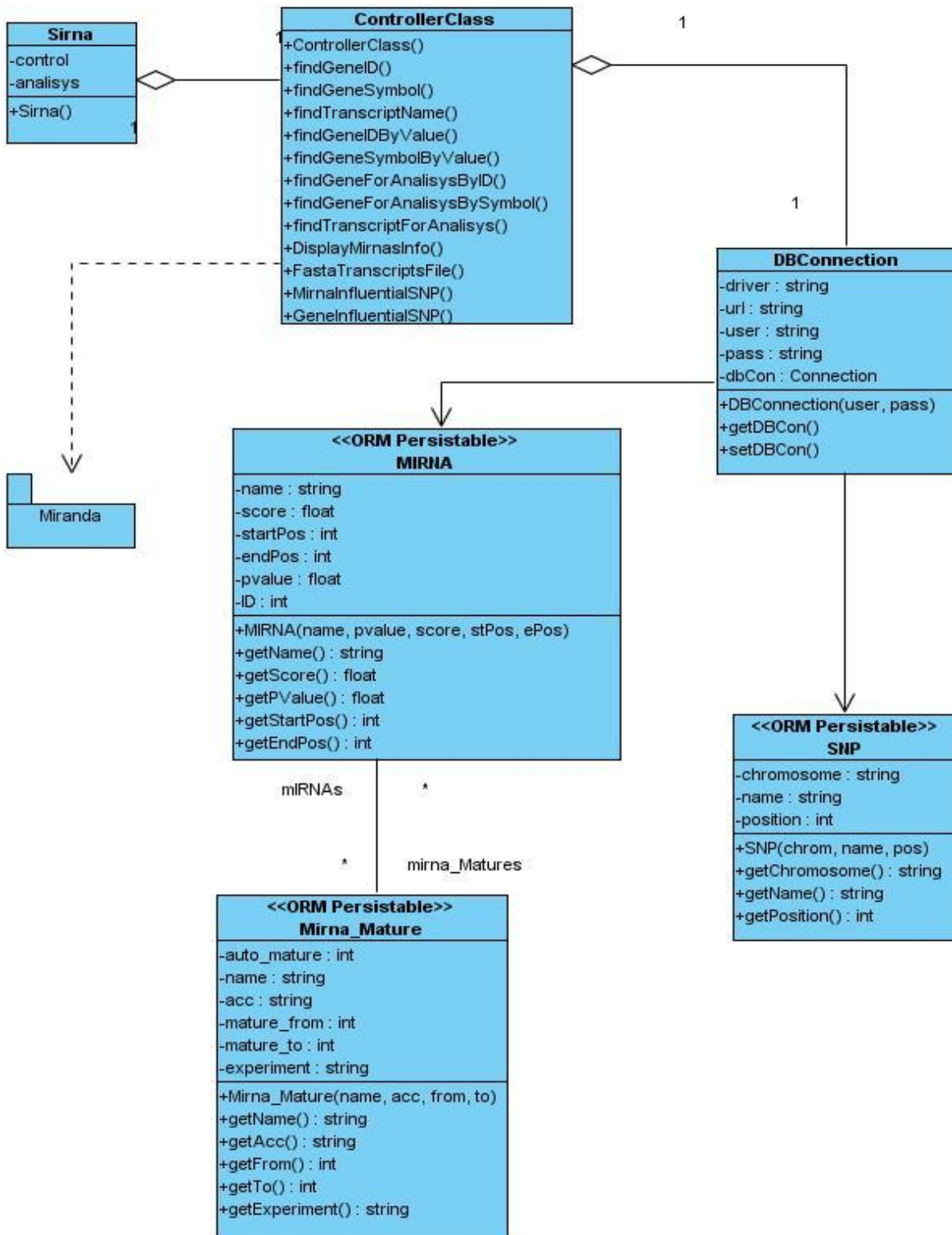


Fig. 6 Diagrama de clases del diseño del CU Buscar SNPs influyentes de un miRNA.

Caso de Uso Crear archivo Fasta con información del genoma humano

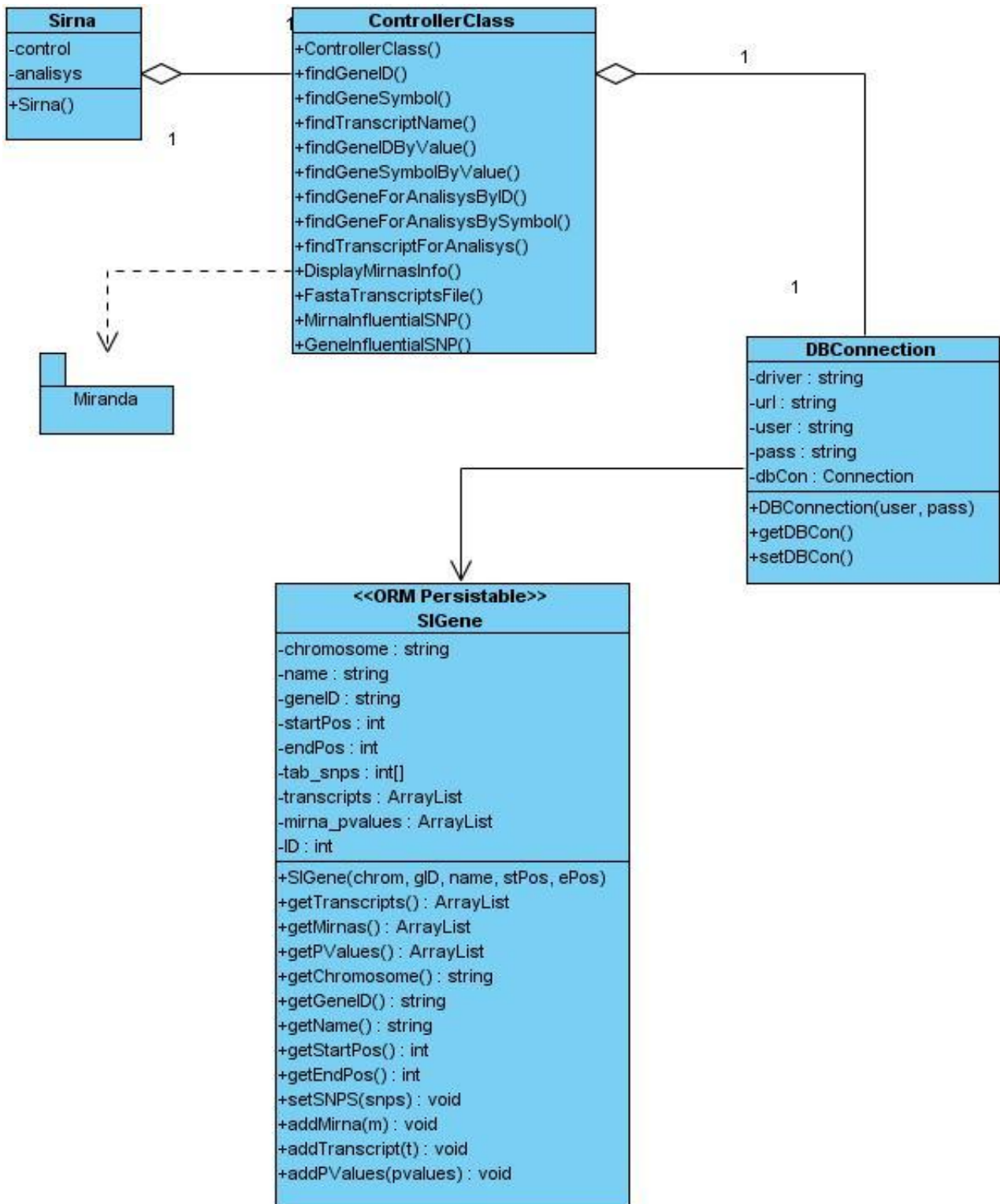


Fig. 7 Diagrama de clases del diseño CU Crear archivo Fasta con información del genoma humano.

Caso de Uso Buscar información de miRNAs

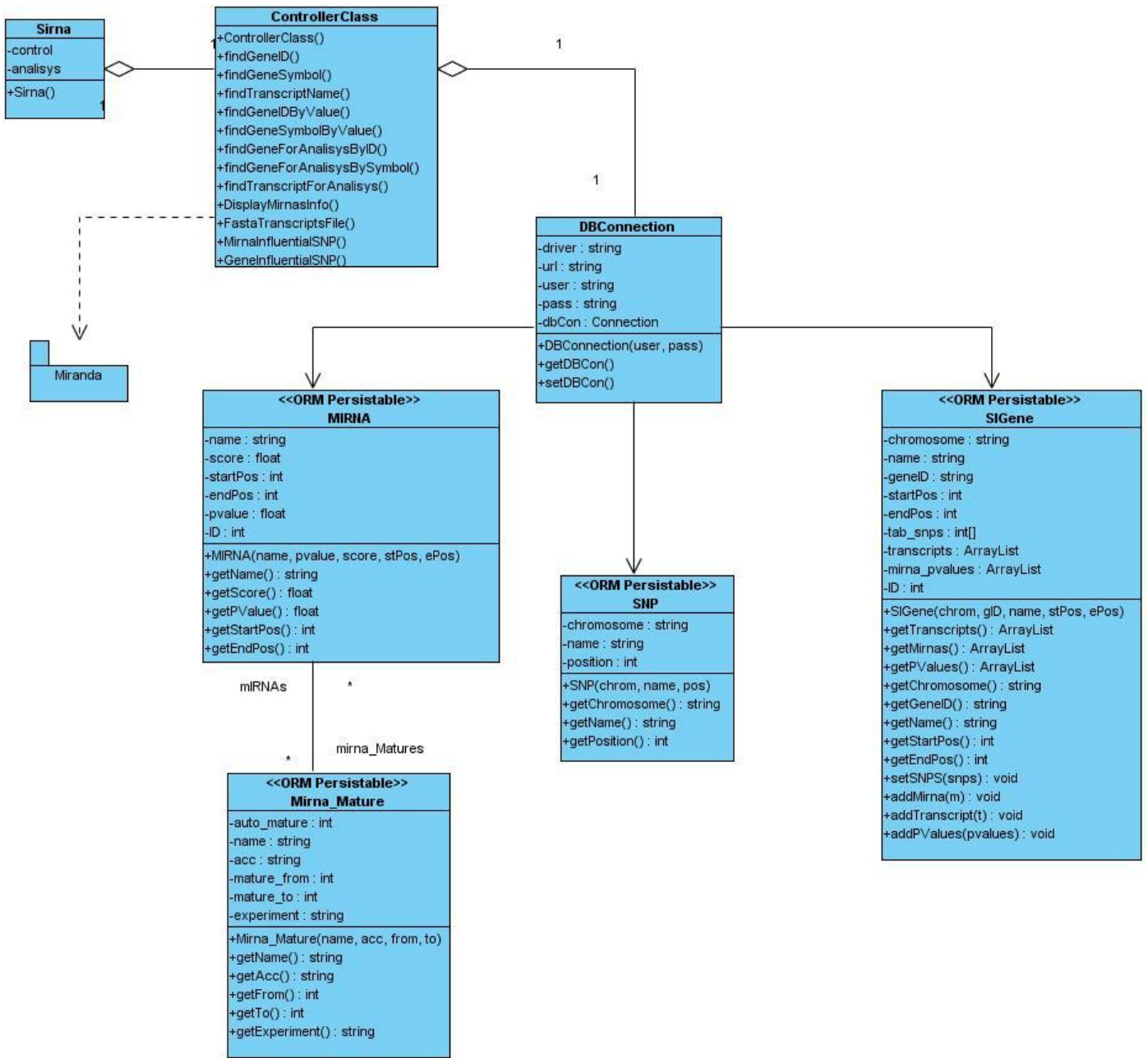


Fig. 8 Diagrama de clases del diseño CU Buscar información de miRNAs.

2.11 Diseño de la base de datos (clases persistentes)

Se mantiene el diseño de la base de datos existente aunque para el trabajo con los miRNAs se añaden nuevas entidades.

A continuación se presenta el diagrama de clases persistentes que está compuesto por dichas clases y las correspondientes relaciones entre ellas y el modelo físico de datos que es una representación de las tablas de la base de datos y sus relaciones.

Modelo lógico de datos

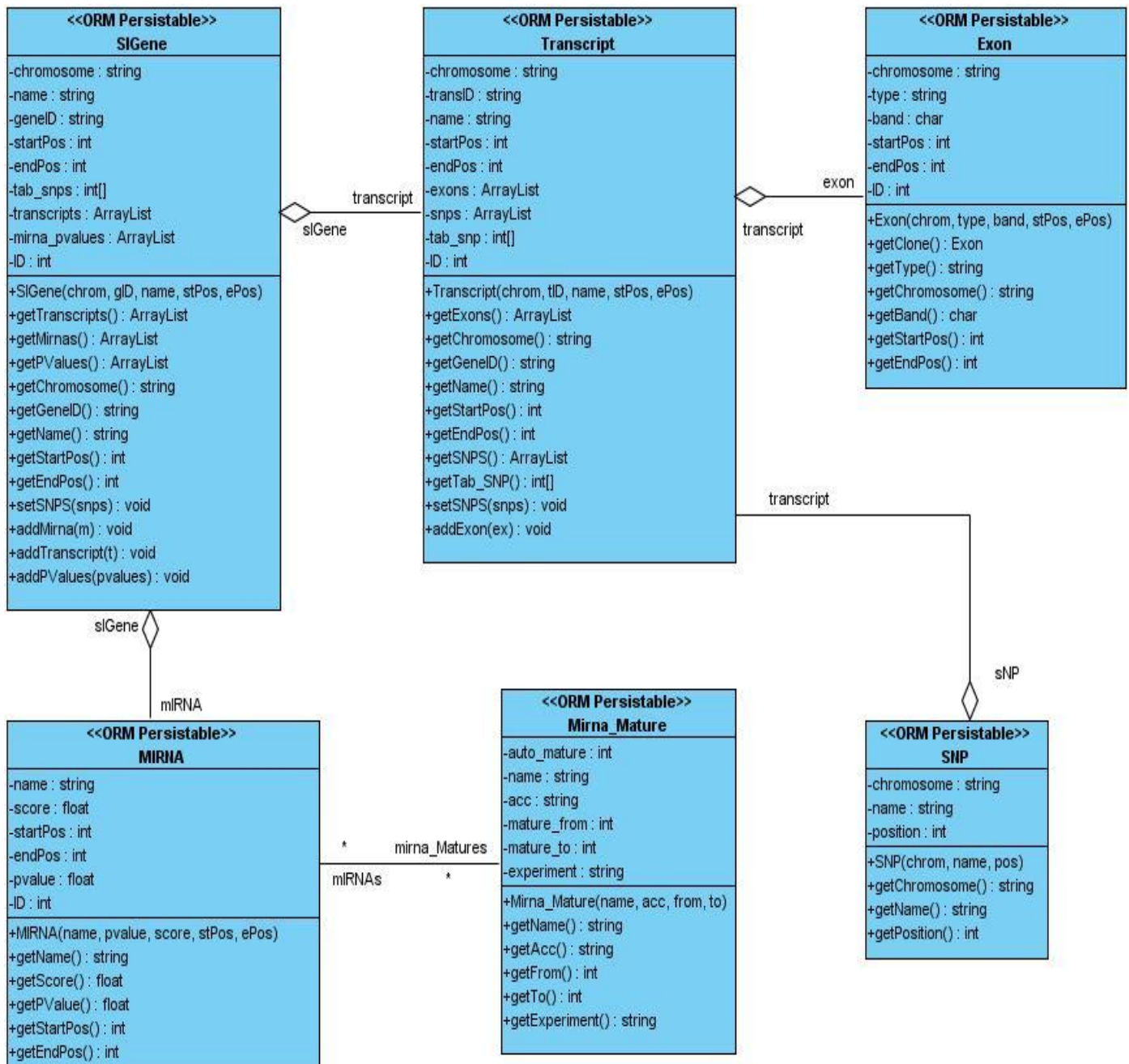


Fig. 9 Diagrama de clases persistentes.

Modelo físico de datos

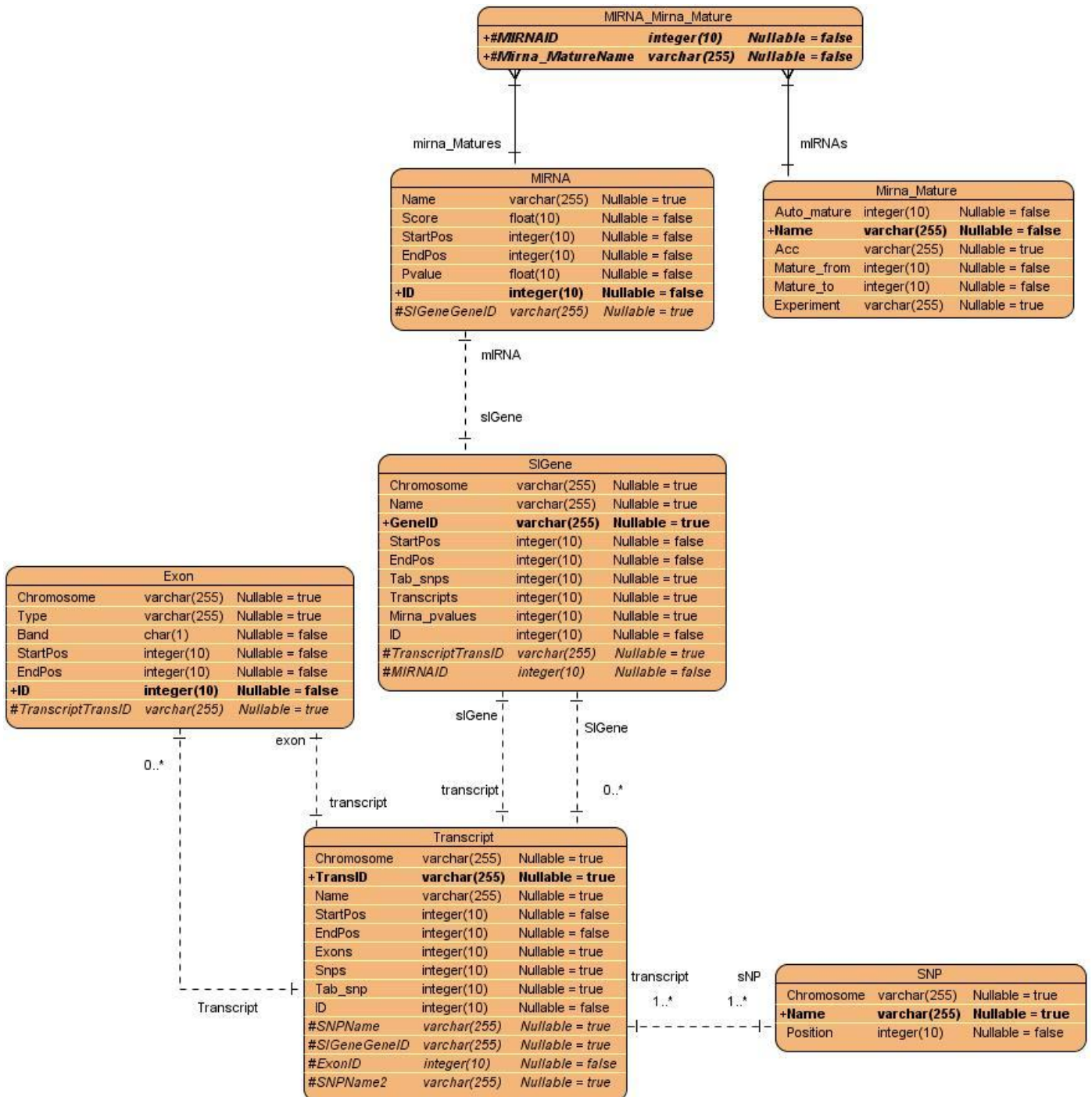


Fig. 10 Diagrama Entidad-Relación.

2.12 Diagramas de Interacción (Secuencia, Colaboración)

Los diagramas de interacción describen secuencias de intercambios de mensajes entre los roles que implementan el comportamiento de un sistema. Estos diagramas proporcionan una vista integral del comportamiento del sistema, muestran el flujo de control a través de muchos objetos.

Los diagramas de interacción se exhiben en dos diagramas centrados en distintos aspectos pero complementarios:

Los diagramas de secuencia para el diseño del sistema propuesto muestran la secuencia de mensajes entre objetos durante un escenario concreto (**Anexo 1**).

Los diagramas de colaboración son útiles en la fase exploratoria para identificar objetos. La distribución de los objetos en el diagrama permite observar adecuadamente la interacción de un objeto con respecto a los demás.

2.13 Diagrama de despliegue

Los diagramas de despliegue muestran la disposición física de los distintos **nodos** que componen un sistema y el reparto de los componentes sobre dichos nodos.

Un nodo es un elemento físico que existe en tiempo de ejecución y representa un recurso computacional, que generalmente tiene algo de memoria y, a menudo, capacidad de procesamiento. Estos se utilizan para modelar la topología del hardware sobre el que se ejecuta el sistema. Representa típicamente un procesador o un dispositivo sobre el que se pueden desplegar los componentes.

Los estereotipos permiten precisar la naturaleza del equipo:

Procesadores: Nodo con capacidad de procesamiento. Puede ejecutar un componente.

Dispositivos: Nodo sin capacidad de procesamiento. Representa cualquier otro dispositivo hardware.

Los nodos se relacionan mediante conexiones bidireccionales.

Las conexiones se modelan como asociaciones, con todas las características que implica. Pueden ser USB, HTTP, JDBC, ADO, HTTPS, entre otras.

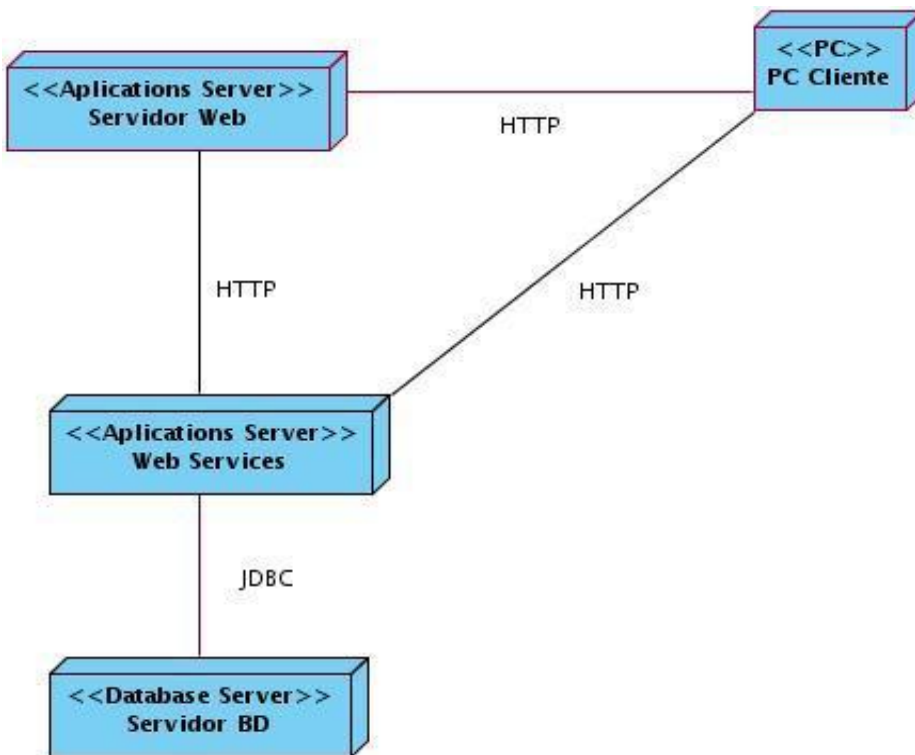


Fig. 11 Diagrama de despliegue.

2.14 Seguridad

La seguridad de la herramienta que se presenta se evidencia a través del servidor de base de datos PostgreSQL. A la información almacenada en este solo tendrá acceso el administrador del sistema, evitando así cambios no deseados en los datos.

2.15 Conclusiones

En este capítulo se ha brindado una definición de los requisitos principales del sistema así como su descripción, posibilitando concretar las funcionalidades que debe tener. Se determina el diseño apropiado para las nuevas funcionalidades en *SIRNA Web Services*, brindando una clara visión de los principios de diseño utilizados. Para esto se expone primeramente la vista lógica del sistema. Luego se presentan los diagramas de clases del diseño, los cuales siguen el mismo patrón de arquitectura de la versión anterior, además de los diagramas de interacción de dichas clases para el logro de una implementación más detallada y organizada de los procesos que llevará a cabo el sistema. Se obtuvo el modelo de despliegue con el cual quedó representada la distribución física del sistema.

CAPÍTULO 3: IMPLEMENTACIÓN Y PRUEBA

Introducción

En el presente capítulo se realiza el modelo de implementación a partir de los resultados del flujo de trabajo anteriormente descrito (Diseño). Se obtienen todos los algoritmos necesarios para la implementación de las nuevas funcionalidades. Además se describen las pruebas que se le han realizado al sistema, y como evidencia de ello se tendrán en cuenta esencialmente los casos de prueba de caja negra.

3.1 Diagrama de componentes

Los diagramas de componentes describen los elementos físicos del sistema y sus relaciones, muestran las opciones de realización incluyendo código fuente, binario y ejecutable.

Los componentes representan todos los tipos de elementos que entran en la fabricación de aplicaciones informáticas. Pueden ser simples archivos, paquetes, bibliotecas cargadas dinámicamente, cada uno de los cuales tiene un estereotipo diferente.

Son utilizados para modelar la vista estática de un sistema. Muestra la organización y las dependencias que existen entre un conjunto de componentes.

3.1.1 Diagrama de componentes del sistema

Los web services no están ligados a una arquitectura de componentes en particular. Los protocolos facilitan a nivel base la comunicación entre las distintas infraestructuras de objetos distribuidos. Por este motivo, únicamente es necesario preocuparse del desarrollo y utilización de Servicios Web. Por tanto en este diagrama de componentes se agrupan las clases en dependencia de la utilidad de estas en la aplicación.

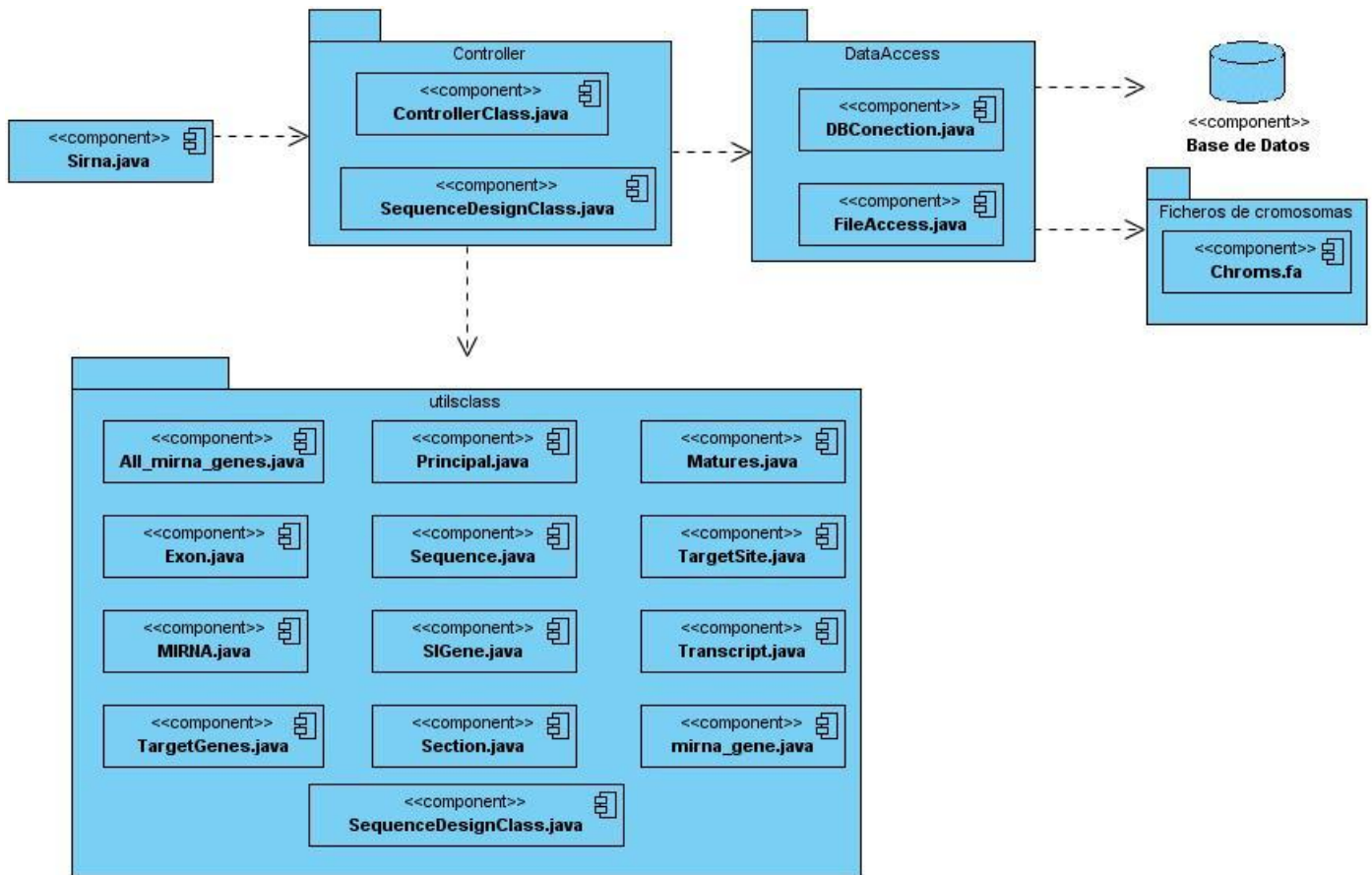


Fig. 12 Diagrama de componentes de *SiRNA Web Services*. Se observa el paquete que maneja la interacción con los datos, tanto los que se encuentran en la base de datos como en los ficheros, así como los demás paquetes que son utilizados por el Controlador para brindar los servicios.

3.2 Diseño de algoritmos

En este epígrafe se describen los principales algoritmos implementados en el componente *SiRNA Web Service*.

3.2.1 Algoritmo para la obtención de SNPs influyentes en un miRNA

MirnaInfluentialSNP (ID del miRNA)

```

{
    Búsqueda del ID del miRNA en la BD y validación del mismo.
    Creación de un arreglo de miRNAs maduros (Matures) contenidos en el miRNA especificado.

    Creación del 2do archivo de entrada al software miRanda (F2) con todos los genes almacenados en la BD.

    Ciclo (Matures. Longitud)
    {

```

Creación de un listado de genes silenciados (Original)=**RunMiranda** (F2, ID del miRNA maduro que se está analizando, secuencia sin cambios del miRNA maduro).

Obtención de los SNPs presentes en el miRNA maduro.

Modificación de los SNPs influyentes en el miRNA maduro=**FindInfluentialSNPInMature** (F2, Original, SNPs del miRNA maduro, ID del miRNA maduro, secuencia del miRNA maduro).

}

Devuelve un arreglo de los miRNAs maduros (con sus características modificadas) contenidos en el miRNA seleccionado.

}

RunMiranda (Archivo, ID del miRNA maduro, Secuencia)

{

Creación del 1er archivo de entrada al software miRanda (F1) con el ID y la secuencia del miRNA maduro.

Creación del directorio del archivo de salida (FSalida) que devuelve la ejecución del miRanda.

Se manda a ejecutar el miRanda con los parámetros correspondientes (F1, F2, FSalida).

Se analiza el fichero de salida para extraer los genes que fueron silenciados.

Se devuelve un listado de los genes silenciados.

}

FindInfluentialSNPInMature (Archivo, listado de genes silenciados por el miRNA maduro sin cambios (Original), SNPs, ID y secuencia del miRNA maduro)

{

Creación de un arreglo (Bases) con las cuatro bases de la secuencia de un miRNA ("A", "C", "U", "G").

Creación de un arreglo donde se guardarán los SNPs influyentes.

Ciclo (SNPs. Longitud)

{

//nota: un arreglo de SNP es un listado de ceros y unos, en caso de existir un uno indica la presencia de un SNP en esa posición de la secuencia del miRNA maduro.

Si (elemento del arreglo de SNPs=1)

{

Se guarda la secuencia del miRNA maduro en una variable temporal (SecTemp).

Ciclo (Bases. Longitud)

{

Si (elemento de Bases != elemento de la secuencia)

{

Se modifica esa base de la secuencia.

Se ejecuta el método **RunMiranda** (F2, ID entrado por parámetro, secuencia modificada).

Si:

La lista de genes silenciados que devuelve el miRanda difiere en tamaño de Original o al menos no devuelve los elementos en el mismo orden, se añade un uno al arreglo de los SNPs influyentes.

```
        Sino:
        Se añade un cero al arreglo de los SNPs influyentes.
    }
}
}
Sino
Se añade un cero al arreglo de los SNPs influyentes.
}
Devuelve un arreglo de los SNPs influyentes de la misma longitud de los SNPs
del miRNA maduro pero solo con unos en las posiciones de la secuencia que
influyen en el proceso de silenciamiento.
}
```

3.2.2 Algoritmo para la obtención de SNPs influyentes en un gen

GeneInfluentialSNP (nombre de un transcrito)

```
{
    Búsqueda del nombre del transcrito en la BD y validación del mismo.
    Creación de un transcrito (T) a partir del nombre introducido por
    parámetro.
    Creación de un arreglo de SNPs (SNPTemp) influyentes=PredecirMirnas (T).

    Modificación del arreglo de los SNPs influyentes del transcrito a partir
    de SNPTemp.

    Devuelve el transcrito creado con sus propiedades modificadas.
}
```

PredecirMirna (Transcrito)

```
{
    Obtención de la secuencia del transcrito.
    Creación de un archivo (File) con el nombre y la secuencia del transcrito.

    Creación de un arreglo de SNPs influyentes=SNPInfluenceGen (T, File).
    Devuelve el arreglo de los SNPs influyentes que tiene un transcrito.
}
```

SNPInfluenceGen (Transcrito, Archivo)

```
{
    Creación de un arreglo (Bases) con las cuatro bases de la secuencia de un
    transcrito ("A", "C", "U", "G").
    Creación de un arreglo donde se guardarán los SNPs influyentes.
    Creación de un arreglo de miRNAs maduros (Original)= GenerarMirnasCoincidentes
    (File).

    Ciclo (arreglo de SNPs del transcrito. Longitud)
    {
        //nota: un arreglo de SNP es un listado de ceros y unos, en caso de
        existir un uno indica la presencia de un SNP en esa posición de la
        secuencia del transcrito.

        Si (elemento del arreglo de SNPs=1)
        {
            Se guarda la secuencia del transcrito en una variable temporal
            (SecTemp).
            Ciclo (Bases. Longitud)
```

```
{
    Si (elemento de Bases !=elemento de la secuencia)
    {
        Se modifica esa base de la secuencia.
        Se crea un nuevo archivo (F1) con la secuencia
        modificada, el nombre del transcrito y el directorio.
        Creación de un arreglo de miRNAs maduros =
        GenerarMirnasCoincidentes (F1).
        Si:
        El arreglo de miRNAs maduros obtenidos luego de
        modificada la secuencia del transcrito difiere en
        tamaño de Original o al menos no devuelve los
        elementos en el mismo orden, se añade un uno al
        arreglo de los SNPs influyentes.
        Sino:
        Se añade un cero al arreglo de los SNPs influyentes.
    }
}
Sino
Se añade un cero al arreglo de los SNPs influyentes.
}
Devuelve un arreglo de los SNPs influyentes del transcrito.
}
```

GenerarMirnasCoincidentes (Archivo)

```
{
    Guarda en un arreglo (Total) todos los miRNAs maduros almacenados en la BD.
    Creación de un arreglo con los miRNAs silenciadores (MiRNAsCoincidentes).

    Ciclo (Total. Longitud)
    {
        Se crea un archivo (F1) con los datos del miRNA maduro.
        Se crea el archivo de salida (FSalida).
        Se manda a ejecutar el miRanda con los parámetros correspondientes (F1,
        Archivo, FSalida).
        Luego de analizar el archivo de salida, si el miRNA maduro silencia al
        transcrito, se añade a MiRNAsCoincidentes.
    }
    Devuelve un arreglo con los miRNAs maduros que silencian al transcrito
    (MiRNAsCoincidentes).
}
```

3.2.3 Algoritmo para la creación de un archivo fasta

FastaTranscriptsFile (Nombre de archivo)

```
{
    Creación de un archivo con el nombre introducido por parámetro.
    Creación de un arreglo con todos los transcritos (Trans) existentes en la BD=
    all_Transcripts ().

    Ciclo (Trans. Longitud)
    {
        Creación de un encabezado con el ID del gen al que pertenece y el nombre
        del transcrito.
    }
}
```



```
        Se escribe en el fichero el encabezado y la secuencia del transcrito
        correspondiente.
    }
    Se cierra el fichero.
}

AllTranscripts ()
{
    Creación de un arreglo de transcritos.
    Se realiza una consulta a la BD para obtener todos los transcritos
    almacenados.
    Por cada transcrito se almacena cada uno de los exones que contiene.
    Se concatena la secuencia de los exones pertenecientes a un transcrito para
    formar la secuencia de este último.
    Se devuelve un arreglo de transcritos.
}
```

3.2.4 Algoritmo para la visualización de informaciones referentes a un miRNA

```
DisplayMirnasInfo (ID del miRNA)
{
    Búsqueda del Id del miRNA en la BD y validación del mismo.
    Creación de un arreglo de miRNAs maduros contenidos en el miRNA especificado.
    SetSNPinMirnaMature (arreglo de miRNAs maduros creado, ID del miRNA).
    SetTargetGenesInMature (arreglo de miRNAs maduros creado, ID del miRNA).
    Devuelve el arreglo de miRNAs maduros con los atributos modificados.
}

SetSNPinMirnaMature (arreglo de miRNAs maduros (Maduros), ID del miRNA)
{
    Se busca en la BD el cromosoma, posición inicial y final del miRNA según el ID
    introducido por parámetro.
    Ciclo (Maduros. Longitud)
    {
        Se busca posición inicial y final del maduro en el miRNA al que
        pertenece.
        Se buscan los SNPs en la BD que coinciden con ese intervalo.
        Se modifica el listado de SNPs del miRNA maduro.
    }
}

SetTargetGenesInMature (arreglo de miRNAs maduros (Maduros), ID del miRNA);
{
    Creación del 2do archivo de entrada al software miRanda (F2) con todos los
    genes almacenados en la BD.
    Ciclo (Maduros. Longitud)
    {
        Se guarda el ID del miRNA maduro en una variable (IDtemp).
        Se guarda la secuencia del miRNA maduro en una variable (SecTemp).
        Se ejecuta el método RunMiranda (F2, IDtemp, SecTemp).
        Se modifica la lista de genes silenciados del miRNA maduro con los
        resultados del método anterior.
    }
}
```

3.3 Tratamiento de errores

A nivel de servidor se tratan los errores 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 lanzamiento de excepciones impide que cada vez que se realiza una búsqueda, esta desencadena dos o tres búsquedas más, si la primera existe. Con esto no solo se evita búsquedas innecesarias sino también demora de tiempo en los procedimientos. (Fig. 13)

The screenshot displays the 'alasSiRNA-Design' web application. The header includes the site name and tagline 'Analysis System of mRNA sequences for siRNA Design', along with navigation links for 'Home Pag', 'Contact Us', and 'About'. The main content area is divided into three sections: 'SEARCH', 'ANALYZE', and 'OTHER SITES'. The 'SEARCH' section features a search input field, radio buttons for 'By gene symbol' (selected), 'By gene id', and 'By transcript name', and a 'Search' button. The 'ANALYZE' section lists various analysis options such as 'Paste your own sequence', 'Common sequence of the transcripts', 'Specific sequence of a transcript', 'Display information of miRNAs', 'Create Fasta file', 'Visualize influential SNPs of a Gene', and 'Visualize influential SNPs of a miRNA'. The 'OTHER SITES' section lists external resources like CIGB, NCBI, EBI, Heber Biotec S.A, Bussines Group, OMS, GAVI, and EMBnet. A 'Web service information' section on the right shows the date 'Tuesday, Juny 14th, 2011' and a blue error message: 'Results: The transcripts which name is 'XR_1' doesn't exist. Check that the letters are capitalized.'

Fig. 13 Mensajes para evitar búsquedas innecesarias en la base de datos. (Prueba de la funcionalidad Buscar SNPs influyentes en un gen).

3.4 Modelo de prueba

Las pruebas son una medida de la calidad del software. Para determinar la calidad de un producto de software se deben efectuar, por tanto, medidas y desarrollar actividades que permitan comprobar el grado de cumplimiento de las especificaciones iniciales del sistema, y es aquí donde las pruebas de software desempeñan un papel fundamental. Estas verifican el desarrollo que va alcanzando el producto durante todas sus etapas, identificando posibles fallos de implementación, calidad, o usabilidad de un programa.

3.4.1 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.

Para la realización de las pruebas se implementaron dos de las funcionalidades (Crear archivo Fasta y Buscar SNPs influyentes en un gen) en la aplicación cliente del sistema *alasSiRNA-Design* y las dos restantes (Buscar información de miRNAs y Buscar SNPs influyentes en un miRNA) fueron implementadas en una aplicación de escritorio con el fin de visualizar los resultados obtenidos.

Se diseñaron los casos de prueba siguientes:

A nivel de desarrollador:

- ✓ Buscar SNPs influyentes en un gen.
- ✓ Crear un archivo Fasta.

A nivel de aplicación:

- ✓ Buscar SNPs influyentes en un miRNA.
- ✓ Buscar información de miRNAs.

Caso de prueba: Buscar SNPs influyentes en un gen.

Flujo central:

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

- ✓ El especialista:
 1. Visualiza la interfaz para mostrar los SNPs influyentes de un gen.
 2. Introduce el nombre del transcrito del cual desea conocer los SNPs influyentes.

3. Comienza el análisis.

✓ El sistema:

4. Valida que el nombre del transcrito sea correcto.

5. Busca los SNPs influyentes en el transcrito.

6. Muestra los SNPs influyentes así como la posición de estos en la secuencia del transcrito.

Flujo alterno:

2.1 El especialista introduce un nombre de un transcrito incorrecto.

2.1.1 El sistema muestra un mensaje de error.

Iteraciones:

Tabla 10: Resultado del caso de prueba Buscar SNPs influyentes en un gen

Clases válidas	Resultado esperado	Resultado de la prueba
<Se selecciona la opción “Visualize influential SNP of a gene” y se introduce el siguiente dato: XR_017611>	<El sistema verifica que el nombre del transcrito sea correcto, busca los SNPs influyentes en el transcrito y muestra los SNPs influyentes así como la posición de estos en la secuencia del transcrito.>	<Satisfactorio>
<Se selecciona la opción “Visualize influential SNP of a gene” y se introduce el siguiente dato: XR_1>	<El sistema verifica que el dato entrado es incorrecto y muestra el siguiente mensaje de error: “Results: The transcript which name is: “XR_1” doesn’t exist. Check that the letters are capitalized”>	<Satisfactorio>

Caso de prueba: Buscar SNPs influyentes en un miRNA.

Flujo central:

El desarrollador selecciona la opción “Visualize influential SNP of a miRNA” para visualizar los SNPs.

✓ El desarrollador:

1. Introduce el ID del miRNA del cual desea conocer los SNPs influyentes.

2. Comienza el análisis.

✓ El sistema:

3. Valida que el ID del miRNA sea correcto.
4. Busca los SNPs influyentes en todos los miRNAs maduros contenidos en el miRNA seleccionado.
5. Muestra los SNPs influyentes por cada uno de los miRNAs maduros.

Flujo alterno:

1.1 El desarrollador introduce un ID de un miRNA incorrecto.

1.1.1 El sistema muestra un mensaje de error.

Iteraciones:

Tabla 11: Resultado del caso de prueba Buscar SNPs influyentes en un miRNA

Clases válidas	Resultado esperado	Resultado de la prueba
<Se selecciona la opción “Visualize influential SNP of a miRNA” y se introduce el siguiente dato: cel-mir-40>	<El sistema verifica que el ID del miRNA sea correcto, busca los SNPs influyentes en todos los miRNAs maduros contenidos en el miRNA seleccionado y muestra los SNPs influyentes por cada uno de los miRNAs maduros.>	<Satisfactorio>
<Se selecciona la opción “Visualize influential SNP of a miRNA” y se introduce el siguiente dato: mir-1>	<El sistema verifica que el dato entrado es incorrecto y muestra el siguiente mensaje de error: “Results: MiRNA which ID is “mir-1” doesn’t exist.”>	<Satisfactorio>

Caso de prueba: Buscar información de miRNAs.

Flujo central:

El desarrollador selecciona la opción “Display information of miRNAs” para obtener los datos.

- ✓ El desarrollador:
 1. Introduce el ID del miRNA del cual desea conocer todos los datos.
 2. Ordena la búsqueda.
- ✓ El sistema:
 3. Valida que el ID del miRNA sea correcto.
 4. Busca los todos los miRNAs maduros presentes en el miRNA seleccionado.

5. Busca los genes que silencian así como los SNPs de cada uno de los miRNAs maduros.
6. Muestra los datos de cada uno de los miRNAs maduros.

Flujo alterno:

- 1.1 El desarrollador introduce un ID de un miRNA incorrecto.
 - 1.1.1 El sistema muestra un mensaje de error.

Iteraciones:

Tabla 12: Resultado del caso de prueba Buscar información de miRNAs

Clases válidas	Resultado esperado	Resultado de la prueba
<Se selecciona la opción “Display information of miRNAs” y se introduce el siguiente dato: cel-mir-40>	<El sistema verifica que el ID del miRNA sea correcto, busca los todos los miRNAs maduros presentes en el miRNA seleccionado, busca los genes que silencian así como los SNPs de cada uno de los miRNAs maduros, muestra los datos de cada uno de los miRNAs maduros.>	<Satisfactorio>
<Se selecciona la opción “Display information of miRNAs” y se introduce el siguiente dato: mir-1>	<El sistema verifica que el dato entrado es incorrecto y muestra el siguiente mensaje de error: “Results: MiRNA which ID is “mir-1” doesn’t exist.”>	<Satisfactorio>

Caso de prueba: Crear un archivo Fasta.

Flujo central:

El especialista selecciona la opción “Create Fasta file” para crear el archivo.

- ✓ El especialista:
 1. Visualiza la interfaz para crear el archivo.
 2. Introduce el nombre del archivo que desea crear.
 3. Ordena la ejecución.
- ✓ El sistema:
 4. Busca todos los genes almacenados en la base de datos.

5. Crea un fichero con información de los genes.
6. Muestra un mensaje en la aplicación de que el archivo fue creado satisfactoriamente.

Flujo alterno:

- 2.1 El especialista no introduce un nombre para el archivo.
 - 2.1.1 El sistema muestra un mensaje de error.

Iteraciones:

Tabla 13: Resultado del caso de prueba Crear un archivo Fasta

Clases válidas	Resultado esperado	Resultado de la prueba
<Se selecciona la opción “Create Fasta file” y se introduce el siguiente dato: genoma>	<El sistema busca todos los genes almacenados en la base de datos, crea un fichero con información de los genes y muestra un mensaje en la aplicación de que el archivo fue creado satisfactoriamente.>	<Satisfactorio>
<Se selecciona la opción “Create Fasta file” y no se introduce ningún dato >	<El sistema muestra el siguiente mensaje de error: “Please, Type a name in the textbox.”>	<Satisfactorio>

3.5 Conclusiones

Se describieron detalladamente los diagramas de componentes que corresponden al modelo de implementación del sistema para un mejor entendimiento de las funcionalidades a implementar. Se realizó una descripción detallada de los algoritmos implementados. Se realizaron las pruebas de caja negra donde se pudo comprobar las funcionalidades de la aplicación. Durante la etapa de prueba se comprobó la existencia de algunos errores que estaban en contra del buen funcionamiento del software, los cuales fueron rectificadas a medida que se fueron realizando pruebas posteriores.

CONCLUSIONES

Con la presente investigación se logró la implementación de nuevas funcionalidades en el componente *SiRNA Web Service* de la aplicación *alasSiRNA-Design*. Para ello se realizó el diseño de los algoritmos obteniéndose los principales artefactos definidos por la metodología *OpenUp*, que en conjunto con la realización de las tareas establecidas, dieron cumplimiento al objetivo general y los objetivos específicos.

De forma general se desarrollaron nuevos algoritmos para *SiRNA Web Service* vinculados a los miRNAs y las mutaciones puntuales. Los algoritmos implementados para determinar qué influencia puede tener una mutación puntual en un miRNA o un gen brindan una opción no encontrada en otras aplicaciones disponibles en Internet.

RECOMENDACIONES

Con el desarrollo del trabajo se han cumplido los objetivos del mismo, a pesar de ello nuevas ideas han ido surgiendo, 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:

- ✓ Continuar el desarrollo del sistema con el objetivo de implementar otras funcionalidades que se desean para próximas versiones.
- ✓ Continuar la optimización de las búsquedas en ficheros y en base de datos.
- ✓ Realizar la implementación de los algoritmos en la plataforma de tareas distribuidas (T-Arenal) para minimizar el tiempo de ejecución.

REFERENCIAS BIBLIOGRÁFICAS

1. Science Daily. [En línea] Rockefeller University, 26 de Julio de 2007. [Citado el: 1 de Diciembre de 2010.] <http://www.sciencedaily.com/releases/2007/07/070721201804.html>.
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., Febrero, 2010.
3. miRanda on-line. [En línea] [Citado el: 20 de Noviembre de 2010.] <http://cbio.mskcc.org/cgi-bin/mirnaviewer/mirnaviewer.pl>.
4. Pic Tar. [En línea] [Citado el: 20 de Noviembre de 2010.] <http://pictar.bio.nyu.edu/cgi-bin/PicTar Vertebrate.cgi>.
5. Miralign. [En línea] [Citado el: 20 de Noviembre de 2010.] <http://bioinfo.au.tsinghua.edu.cn/miralign/>.
6. miRBase. [En línea] [Citado el: 20 de Noviembre de 2010.] <http://microrna.sanger.ac.uk/>.
7. Targetscan. [En línea] [Citado el: 20 de Noviembre de 2010.] <http://www.targetscan.org/>.
8. Aplicación miRanda. [En línea] [Citado el: 20 de Noviembre de 2010.] http://www.microrna.org/miranda_new.html.
9. MiRscan. [En línea] [Citado el: 20 de Noviembre de 2010.] <http://genes.mit.edu/mirscan/>.
10. Ligeya Perezleo Solórzano. ACIMED. *Impacto de la Bioinformática en las ciencias biomédicas*. [En línea] 8 de Junio de 2003. [Citado el: 1 de Diciembre de 2010.] http://scielo.sld.cu/scielo.php?pid=S1024-94352003000400007&script=sci_abstract&lng=en.
11. Adjounian, María Fernanda Correa de. *Silenciamiento de genes mediante RNA interferencia*. Venezuela : s.n., 2007.
12. UVIGEN. [En línea] [Citado el: 18 de Enero de 2011.] <http://uvigen.fcien.edu.uy/utem/camgen/02mut.htm>.
13. SNP . [En línea] [Citado el: 25 de Noviembre de 2010.] http://www.ornl.gov/sci/techresources/Human_Genome/faq/snps.shtml.
14. Bernstein E, Caudy AA, Hammond SM, Hannon GJ. Pubmed.gov. *Role for a bidentate ribonuclease in the initiation step of RNA interference*. [En línea] 18 de Enero de 2001. [Citado el: 1 de Diciembre de 2010.] <http://www.ncbi.nlm.nih.gov/pubmed/11201747>.
15. Vicki Glaser. GEN. *Tapping miRNA-Regulated Pathways*. [En línea] 1 de Marzo de 2008. [Citado el: 25 de Noviembre de 2010.] <http://www.genengnews.com/gen-articles/tapping-mirna-regulated-pathways/2382/>.
16. Wikipedia. *Imagen miRNA*. [En línea] [Citado el: 20 de Noviembre de 2010.] <http://es.wikipedia.org/wiki/MIRNA>.
17. Enright, A. J., B. John, U. Gaul, T. Tuschl, C. Sander, and D. S. Marks. *microRNA targets in Drosophila*. *Genome Biol.* 2003.

18. wellcome trust sanger institute. [En línea] [Citado el: 1 de Diciembre de 2010.] <http://www.sanger.ac.uk>.
19. PHP. [En línea] [Citado el: 18 de Enero de 2011.] <http://www.php.net/>.
20. Python. [En línea] [Citado el: 18 de Enero de 2011.] <http://www.python.org/>.
21. Visual C# Developer Center. [En línea] [Citado el: 18 de Enero de 2011.] <http://msdn.microsoft.com/en-us/vcsharp/default>.
22. Scala. [En línea] [Citado el: 18 de Enero de 2011.] <http://www.scala-lang.org/>.
23. ciberaula. [En línea] [Citado el: 1 de Diciembre de 2010.] http://java.ciberaula.com/articulo/que_es_java.
24. UCI, Informatización. *Arquitectura para los Sistemas que Conforman la Intranet Universitaria*. Ciudad de La Habana : s.n., 2007.
25. *Desarrollo Orientado a Objetos con UML*. [En línea] [Citado el: 20 de Noviembre de 2010.] <http://www.clikear.com/manuales/uml/introduccion.aspx>.
26. *Netbeans vs Eclipse*. [En línea] [Citado el: 25 de Noviembre de 2010.] <http://www.juanjonavarro.com/masquecodigo/2003/12/09/netbeans-frente-a-eclipse>.
27. *Visual Paradigm*. [En línea] [Citado el: 25 de Noviembre de 2010.] [http://www.freedownloadmanager.org/es/downloads/Paradigma_Visual_para_UML_\(M%C3%8D\)_1472_0_p/](http://www.freedownloadmanager.org/es/downloads/Paradigma_Visual_para_UML_(M%C3%8D)_1472_0_p/).
28. desarrolloweb.com. [En línea] 22 de Junio de 2010. [Citado el: 1 de Diciembre de 2010.] <http://www.desarrolloweb.com/actualidad/netbeans-6-9-final-liberado-3613.html>.
29. Garcilaso Jordana. *Introducción Open UP*. [En línea] [Citado el: 1 de Diciembre de 2010.] http://www.mug.org.ar/Descargas/Jornadas/Downloads_GetFile.aspx?id=3136.
30. Y. Correa, Y. Diaz. *Sistema de análisis de secuencias mRNA para el diseño de siRNA*. UCI, Ciudad de La Habana : s.n., 2009.
31. *MySQL*. [En línea] [Citado el: 18 de Noviembre de 2010.] http://www.netpecos.org/docs/mysql_postgres/x57.html.
32. *PostgreSQL*. [En línea] [Citado el: 18 de Noviembre de 2010.] http://www.computerworld.com.au/article/62894/postgresql_affiliates_org_domain.
33. *Tomcat*. [En línea] [Citado el: 25 de Noviembre de 2010.] http://www.programacion.com/tutorial/tomcatintro/1/#1_intro.
34. desarrolloweb.com. [En línea] 2 de Julio de 2010. [Citado el: 1 de Diciembre de 2010.] <http://www.desarrolloweb.com/actualidad/tomcat-7-0-nueva-version-servidor-web-3673.html>.
35. W3C. [En línea] [Citado el: 1 de Diciembre de 2010.] http://www.w3c.es/Prensa/2007/nota070130_webcgm.html.

BIBLIOGRAFÍA

1. Adjounian, María Fernanda Correa de. *Silenciamiento de genes mediante RNA interferencia*. Venezuela : s.n., 2007.
2. Ahmed, H., et al. Prediction of guide strand of microRNAs from its sequence and secondary structure, in BMC Bioinformatics 2009. Chandigarh, India. Disponible en: <http://www.biomedcentral.com/1471-2105/10/105>.
3. Aplicación miRanda. [En línea] [Citado el: 20 de Noviembre de 2010.] http://www.microma.org/miranda_new.html.
4. Bernstein E, Caudy AA, Hammond SM, Hannon GJ. Pubmed.gov. *Role for a bidentate ribonuclease in the initiation step of RNA interference*. [En línea] 18 de Enero de 2001. [Citado el: 1 de Diciembre de 2010.] <http://www.ncbi.nlm.nih.gov/pubmed/11201747>.
5. Booch, G., Rumbaugh, J., Jacobson, I. "El Lenguaje Unificado de Modelado". Addison-Wesley. 1999
6. ciberaula. [En línea] [Citado el: 1 de Diciembre de 2010.] http://java.ciberaula.com/articulo/que_es_java.
7. *Desarrollo Orientado a Objetos con UML*. [En línea] [Citado el: 20 de Noviembre de 2010.] <http://www.clikear.com/manuales/uml/introduccion.aspx>.
8. desarrolloweb.com. [En línea] 22 de Junio de 2010. [Citado el: 1 de Diciembre de 2010.] <http://www.desarrolloweb.com/actualidad/netbeans-6-9-final-liberado-3613.html>.
9. desarrolloweb.com. [En línea] 2 de Julio de 2010. [Citado el: 1 de Diciembre de 2010.] <http://www.desarrolloweb.com/actualidad/tomcat-7-0-nueva-version-servidor-web-3673.html>.
10. Enright, A. J., B. John, U. Gaul, T. Tuschl, C. Sander, and D. S. Marks. *microRNA targets in Drosophila*. *Genome Biol*. 2003.
11. Garcilaso Jordana. *Introducción Open UP*. [En línea] [Citado el: 1 de Diciembre de 2010.] http://www.mug.org.ar/Descargas/Jornadas/Downloads_GetFile.aspx?id=3136.
12. Herramientas CASE [Consultado el 1 de Diciembre de 2010]. Disponible en: <http://www.cyta.com.ar/biblioteca/bddoc/bdlibros/proyectoinformatico/libro/c5/c5.html>.
13. IDT Scitools RNAi Design. [Consultado el: 20 de Noviembre de 2010]. Disponible en: <http://www.idtdna.com/scitools/applications/rnai/rnai.aspx>.
14. Ligeya Perezleo Solórzano. ACIMED. *Impacto de la Bioinformática en las ciencias biomédicas*. [En línea] 8 de Junio de 2003. [Citado el: 1 de Diciembre de 2010.] http://scielo.sld.cu/scielo.php?pid=S1024-94352003000400007&script=sci_abstract&lng=en.
15. Miralign. [En línea] [Citado el: 20 de Noviembre de 2010.] <http://bioinfo.au.tsinghua.edu.cn/miralign/>.

16. miRanda on-line. [En línea] [Citado el: 20 de Noviembre de 2010.] <http://cbio.mskcc.org/cgi-bin/mirnaviewer/mirnaviewer.pl>.
17. miRBase. [En línea] [Citado el: 20 de Noviembre de 2010.] <http://microrna.sanger.ac.uk/>.
18. MiRscan. [En línea] [Citado el: 20 de Noviembre de 2010.] <http://genes.mit.edu/mirscan/>.
19. MySQL . [En línea] [Citado el: 18 de Noviembre de 2010.] http://www.netpecos.org/docs/mysql_postgres/x57.html.
20. Netbeans vs Eclipse. [En línea] [Citado el: 25 de Noviembre de 2010.] <http://www.juanjonavarro.com/masquecodigo/2003/12/09/netbeans-frente-a-eclipse>.
21. Parrado, Vladir Antonio. *Sistema informatico para el diseño de siRNAs mediante el analisis de secuencias de ARN mensajero*. Ciudad de La Habana : s.n., Febrero, 2010.
22. Perez, J. Web Services, in Universidad Politécnica de Valencia, Valencia: España. p. 15.
23. PHP. [En línea] [Citado el: 18 de Enero de 2011.] <http://www.php.net/>.
24. Pic Tar. [En línea] [Citado el: 20 de Noviembre de 2010.] <http://pictar.bio.nyu.edu/cgi-bin/PicTar Vertebrate.cgi>.
25. PostgreSQL. [En línea] [Citado el: 18 de Noviembre de 2010.] http://www.computerworld.com.au/article/62894/postgresql_affiliates_org_domain.
26. Python. [En línea] [Citado el: 18 de Enero de 2011.] <http://www.python.org/>.
27. Quiñones. E, A. Introducción a Postgres. Asociación Peruana de Software Libre
28. Scala. [En línea] [Citado el: 18 de Enero de 2011.] <http://www.scala-lang.org/>.
29. Science Daily. [En línea] Rockefeller University, 26 de Julio de 2007. [Citado el: 1 de Diciembre de 2010.] <http://www.sciencedaily.com/releases/2007/07/070721201804.html>.
30. SNP. [En línea] [Citado el: 25 de Noviembre de 2010.] http://www.ornl.gov/sci/techresources/Human_Genome/faq/snps.shtml.
31. TargetsCan. [En línea] [Citado el: 20 de Noviembre de 2010.] <http://www.targetscan.org/>.
32. Tirado, N. and K. Roman, SISTEMA DE ANÁLISIS DE SECUENCIAS MRNA PARA EL DISEÑO DE SIRNA. V 1.1, in Facultad 6. 2009, UCI: Ciudad Habana. p. 108.
33. Tomcat. [En línea] [Citado el: 25 de Noviembre de 2010.] http://www.programacion.com/tutorial/tomcatintro/1/#1_intro.
34. UCI, Informatización. *Arquitectura para los Sistemas que Conforman la Intranet Universitaria*. Ciudad de La Habana : s.n., 2007.
35. UVIGEN. [En línea] [Citado el: 18 de Enero de 2011.] <http://uvigen.fcien.edu.uy/utem/camgen/02mut.htm>.

36. Vicki Glaser. GEN. *Tapping miRNA-Regulated Pathways*. [En línea] 1 de Marzo de 2008. [Citado el: 25 de Noviembre de 2010.] <http://www.genengnews.com/gen-articles/tapping-mirna-regulated-pathways/2382/>.
37. Vienna RNA Package. [Consultado el 18 de Enero de 2011] Disponible en: <http://www.tbi.univie.ac.at/RNA/>.
38. Visual C# Developer Center. [En línea] [Citado el: 18 de Enero de 2011.] <http://msdn.microsoft.com/en-us/vcsharp/default>.
39. *Visual Paradigm*. [En línea] [Citado el: 25 de Noviembre de 2010.] [http://www.freedownloadmanager.org/es/downloads/Paradigma_Visual_para_UML_\(M%C3%8D\)_14720_p/](http://www.freedownloadmanager.org/es/downloads/Paradigma_Visual_para_UML_(M%C3%8D)_14720_p/).
40. W3C. [En línea] [Citado el: 1 de Diciembre de 2010.] http://www.w3c.es/Prensa/2007/nota070130_webcgm.html.
41. wellcome trust sanger institute. [En línea] [Citado el: 1 de Diciembre de 2010.] <http://www.sanger.ac.uk>.
42. Wikipedia. *Imagen miRNA*. [En línea] [Citado el: 20 de Noviembre de 2010.] <http://es.wikipedia.org/wiki/MIRNA>.
43. Y. Correa, Y. Diaz. *Sistema de análisis de secuencias mRNA para el diseño de siRNA*. UCI, Ciudad de La Habana : s.n., 2009.

ANEXOS

Anexo 1. Diagramas de secuencias

CU Buscar SNPs influyentes de un gen. *SiRNA Web Services*

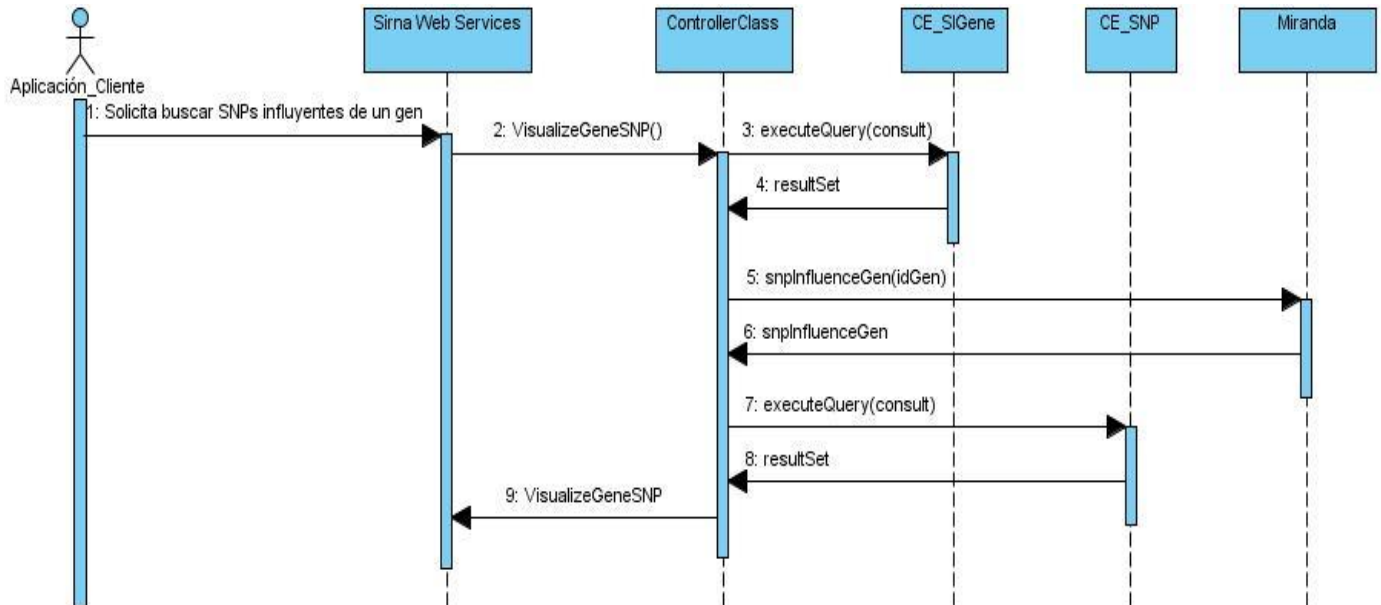


Fig. 14 Diagrama de secuencia CU Buscar SNPs influyentes de un gen.

CU Buscar SNPs influyentes de un miRNA. *SiRNA Web Services*

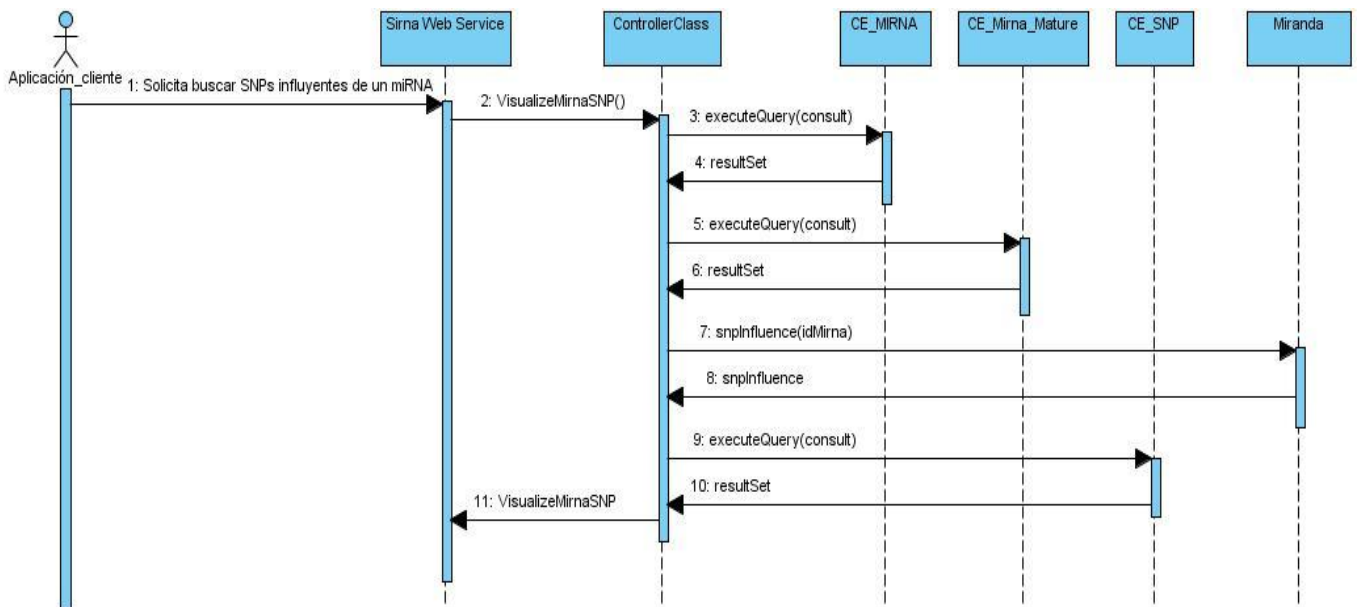


Fig. 15 Diagrama de secuencia CU Buscar SNPs influyentes de un miRNA.

CU Crear archivo Fasta con información del genoma humano. *SiRNA Web Services*

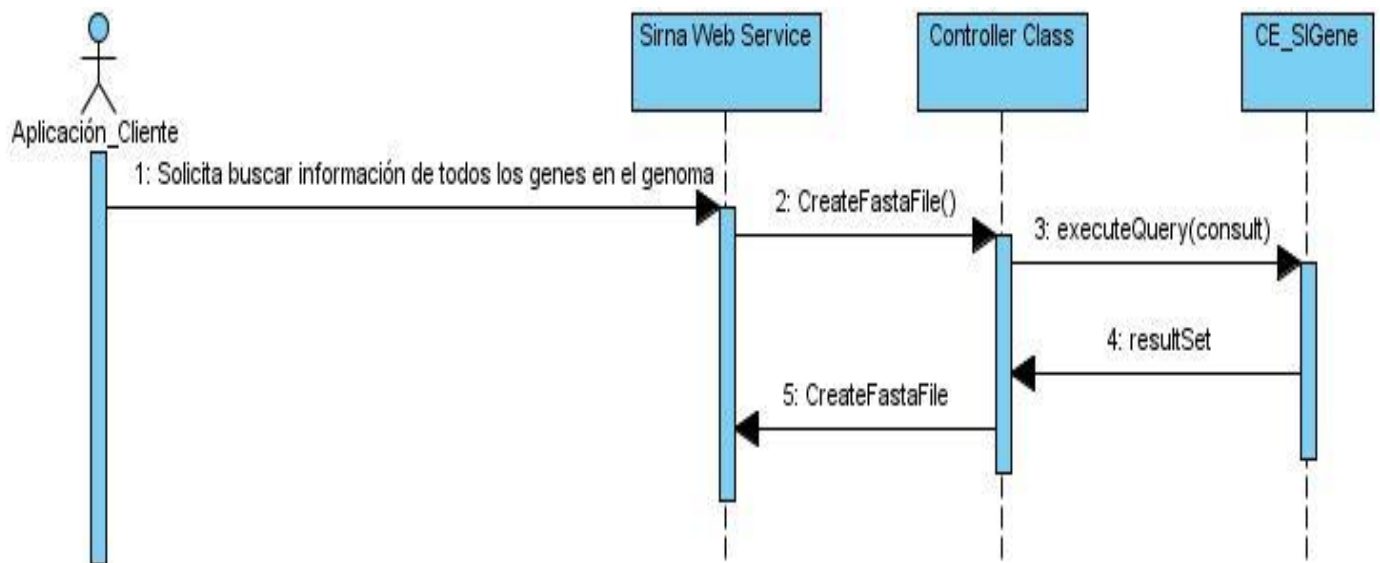


Fig. 16 Diagrama de secuencia CU Crear archivo Fasta con información del genoma humano.

CU Buscar información de miRNAs. *SiRNA Web Services*

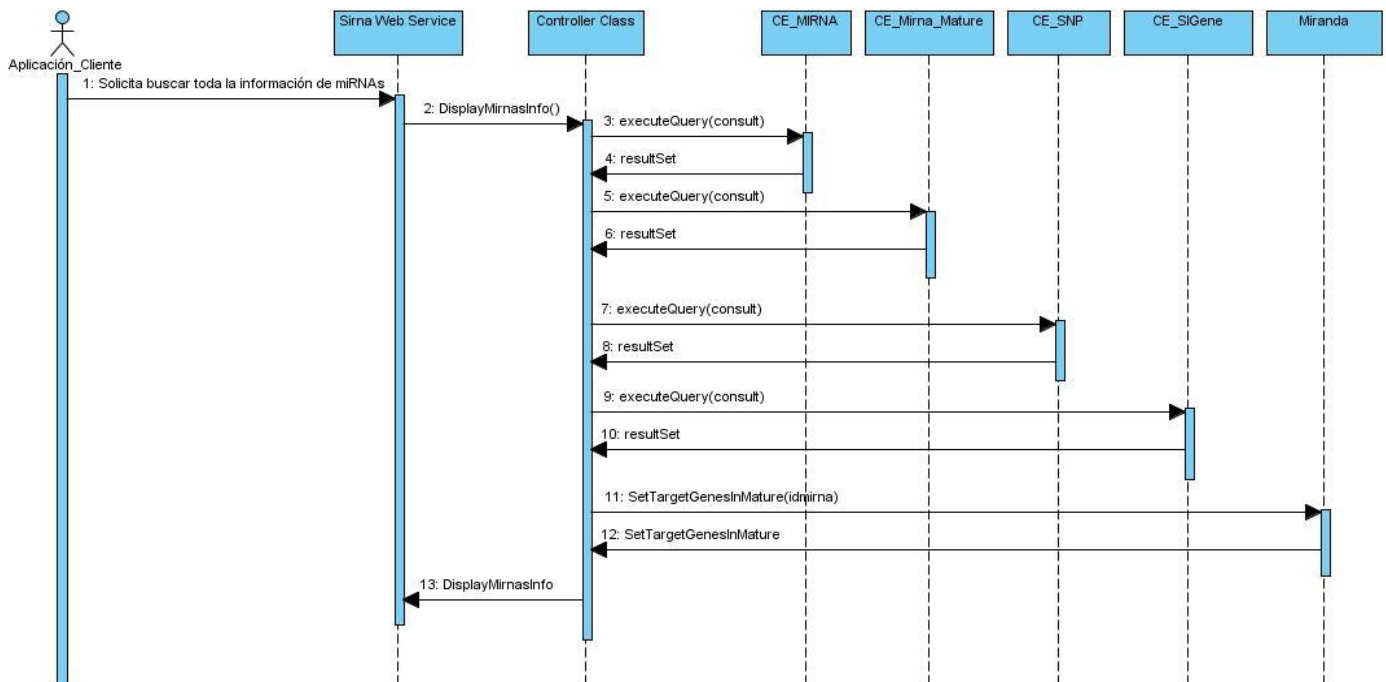


Fig. 17 Diagrama de secuencia CU Buscar información de miRNAs.

GLOSARIO

1. Delecciones: Pérdida de un segmento cromosómico que lleva consigo la desaparición de la información genética contenida en él.
2. Formato Fasta: es un formato de fichero informático basado en texto, utilizado para representar secuencias tanto de ácidos nucleicos, como de péptidos, y en el que los pares de bases o los aminoácidos se representan usando códigos de una única letra.