Universidad de las Ciencias Informáticas

Facultad 6


Trabajo de diploma para optar por el título de

Ingeniero en Ciencias Informáticas

**Título: "Portlet para la gestión de información genética de los grupos de ortología"**

**Autores:**

Daneilys Pérez Villegas

Yonnis Turro Alfaro

**Tutores:**

MSc. Tonysé de la Rosa Martín

Ing. Adisley Reyes Crespo

La Habana, junio de 2013

"Año 55 de la Revolución"

# *Frase*



*"Si no existe organización, las ideas, después del primer impulso van perdiendo eficacia, van cayendo en la rutina, van cayendo en el conformismo y acaban por ser simplemente un recuerdo"*

*CHE*

# *Declaración de autoría*

**Declaración de autoría**

Declaramos ser autores del presente trabajo de diploma y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales del mismo, con carácter exclusivo. Autorizamos a dicho centro para que haga el uso que estime pertinente con este trabajo.

Para que así conste firmamos la presente a los ___ días del mes de _____ del año _____.

Daneilys Pérez Villegas

_____

Firma del autor

Yonnis Turro Alfaro

_____

Firma del autor

Ing. Adisley Reyes Crespo

_____

Firma del tutor

MSc. Tonysé de la Rosa Martín

_____

Firma del tutor

**Datos de contacto**

**MSc. Tonysé de la Rosa Martín**: El tutor es Máster en Bioinformática graduado de Ingeniero en Ciencias Informáticas, en 2010 obtuvo la categoría de Instructor. Cuenta con 5 años de experiencia trabajando en Bioinformática. Se Graduó en el año 2008 en la Universidad de las Ciencias Informáticas. Correo electrónico: tdelarosa@uci.cu.

**Ing. Adisley Reyes Crespo**: La tutora es Ingeniera en Ciencias Informáticas, es profesora asistente. Cuenta con 6 años de experiencia trabajando en Bioinformática. Se Graduó en el año 2007 en la Universidad de las Ciencias Informáticas. Correo electrónico: areyesc@uci.cu.

# Agradecimientos

*Agradecimientos de Daneilys Pérez Villegas.*

*Hoy es un día especial para mí, ya han pasado cinco cursos de sacrificios, sueños, alegrías y ha llegado el momento de enfrentarme al mundo como una profesional, son innumerables las personas a las que tengo que agradecerles por ayudarme a llegar.*

*Quisiera agradecer en primer lugar a mi madre por brindarme apoyo, confianza y seguridad para continuar superándome profesionalmente cada día más, a quien le debo todo lo que soy y los logros alcanzados porque ha sabido dar todo de ella para mi bienestar de manera incondicional desde que nací y aún más durante mi carrera. Recuerda que eres mi razón de ser. Se me hace difícil hablar de tanto en tan pocas palabras pero te estaré agradecida toda mi vida. ¡ Te amo !*

*A mi hermanita que ha sido la persona por la cual me he esforzado para que ella tome mi ejemplo de profesional, Rosi te amo eres lo más grande de mi vida.*

*A mis abuelos que han sido las personas que me lo han dado todo en la vida, quienes me infundieron la ética y el rigor que guían mi transitar por la vida, por su ejemplo de superación incasable, por su comprensión y confianza, por su amor incondicional. A mi abuela que me ha soportado mis 23 años de vida con todas mis pesadeces y malas crianzas, Nilda te quiero. A mi abuelo que ha sido ese hombre ejemplar que siempre me ha enseñado la cara de la vida y como enfrentar las cosas Te amo pipa.*

*A mi tía la loca que ha sido muy importante en mi vida, me ha guiado por el buen camino y aunque siempre hemos estado fajadas la quiero con la vida.*

*A mi familia de aquí de la habana por haberme dado todo el apoyo y confianza en estos 5 años de vida, en especial a mi prima Ely y a Edenia la loca, muchas gracias por todo, les estaré agradecida toda la vida.*

*A todas las personas maravillosas que he conocido durante estos cinco cursos, saber que han estado ahí me hace entender la parte linda de la vida. Gracias a ustedes: Kirenia, Grey, Lisandra, Dreidys, Yanet Miji, Dayatni, Yuraysi, Ennis, Mari, Yude, Anay, Niuvis, Katy, Ariannys, Claudia, La Chiqui, Edwin, Henry, Fito, Yoandry, Yordan, Yosbel, Migue, Dasley, El Flaco, al Pollo, Alejandro, Jose, Alfre, el Hermano, Fuki, Arian, Los Frank, Bazan, Omar. En fin muchas gracias a todos.*

*A todas mis amistades yusaf que siempre han estado ahí en los momentos que me han hecho falta, en especial a Leidanys y Haymee. Gracias por todos los momentos que me dieron, fueron muy lindos.*

*A mis tutores Adisley y Tonysé por compartir conmigo sus conocimientos y experiencias, por brindarme siempre sus consejos oportunos, por indicarme el mejor camino y por su ayuda incondicional durante este año de sacrificio.*

*A mi compañero de aula, de proyecto, de carrera, muchas gracias nunca podré pagarte todo lo que hiciste por mi te acuerdas cuando Grey decía Viltre yo creo que la Gorda te sabe algo, lo que pasaba es que tu querías que me graduara por eso simplemente siempre me ayudaste, mil gracias. Te quiero mi amigo incondicional.*

*A una pareja linda que siempre me estaban diciendo "gorda te vas a graduar cuando carlitín sea tu tutor", gracias a ella que pasó muchas noches de desvelo repasándome y a él que siempre me estaba invitando a hacer caldosa. A Olivia y a Carlos.*

# Agradecimientos

*A un chiquitín por soportarme siempre en el camino de aquí para allá y de allá para acá aunque a veces me daba unas quemá en la autopista jaja pero bueno eso queda entre amigos, gracias por ser mi amigo. Te quiero Héctor.*

*A una persona que Amo, que llegó en mi vida cuando pensé que todo estaba perdido, mil gracias por estos 7 meses que me has dado han sido muy hermosos. Gracias por ayudarme y entenderme en mis momentos difíciles. Te amo.*

*A mi amiga, esa inigualable compañera de locuras que me ha acompañado durante 5 años y a la que hoy le puedo llamar hermana, gracias por todo. A su novio por soportarme y cocinarme estos 5 cursos, esos son los dos monguitos: Jisel y Alejandro. ¡ Los quiero mucho !*

*A una amiga muy especial que apareció en mi vida en un momento muy difícil y supo darme la mano y decirme camina que ahora es que empieza tu vida, mil gracias te doy. Te quiero mucho aunque en algún momento de demostré que no era así. Mi amiga incondicional Dunia.*

*A una persona que ha sido mi ídolo, siempre he seguido sus consejos y siempre ha estado ahí tanto en momento buenos como en los malos, a ti te debo la vida, te quiero mucho Bismi eres esa hermana mayor que no tuve.*

*A mi duo de tesis por haberme soportado en este infinito curso, aguantarme a mí sé que no es fácil, gracias por toda la paciencia que tuviste conmigo, fue lindo compartir todos los momentos contigo incluso los más difíciles. Gracias*

*A mi santo y a dios que siempre han estado ahí apoyándome toda la vida. Gracias*

*No porque te mencione de ultima dejas de ser importante de hecho yo diría que eres la más importante en estos agradecimientos, por tu paciencia, tu confianza en mí aunque a veces me decía gorda tú no te vas a graduar, por tu preocupación cada vez que decía no puedo más aunque a veces me peleaba y se ponía en fase y no había quien la soportara, por cada granito de arena que pusiste en mi vida y en mi futuro, para ti es este regalo. Eres la persona a la cual de debo mi carrera sin tu apoyo no me hubiese podido graduar, mil gracias chuchi no me alcanzan las palabras para decirte lo agradecida y orgullosa que estoy de ti. Gracias por ser mi guía a seguir en estos cinco años.*

*Al tribunal y al oponente todas las críticas siempre fueron constructivas. Gracias.*

*A todos aquellos que me preguntaban en los pasillos ¿Cómo va la tesis? ¿Cuándo discutes? ¿Qué te falta para terminar?*

*A nuestro Comandante Fidel Castro Ruz y a la Revolución Cubana por darme la oportunidad de convertirme en profesional.*

# Agradecimientos

*Agradecimientos de Yonnis Turro Alfaro*

*En primer lugar a dios por haberme ayudado en estos 5 años infinitos de mi vida, siempre te estaré agradecido.*

*Agradezco a mi familia, que han dejado de tenerme por tanto tiempo para que logre mi preparación como persona y siempre me han dado amor y comprensión.*

*A mi mamá a la cual le debo todo lo que soy y a mi abuela*

*A mis hermanas Yoslaine y Yannis. Las quiero.*

*Agradezco a mi novia su apoyo y empuje me ayudaron mucho a terminar la tesis, su comprensión y amor me dieron fuerza.*

*¡ Te amo !*

*Agradezco a mis amigos que siempre estuvieron ahí incluso en los momentos más difíciles, muchos no se encuentran hoy por razones de la vida, pero siempre estarán en mi corazón. Especial a Ricardo.*

*A mis compañeros de cuarto no les agradezco porque no me dejaban dormir obligándome a que trabajara en la tesis: Andry y Daniel.*

*Cuando esta página se hace no vienen todos los nombres a la mente, pero siempre se sabe quién está dentro de ella aunque no se mencione. Muchas gracias a todos.*

*A mis tutores Adisley y Tonysé gracias por guiarme por el buen camino y por enseñarme sus conocimientos, por tantas noches de desvelo que pasaron conmigo. Muchas gracias.*

*A mi mejor amigo, ese que siempre sin duda estuvo ahí, no tengo como agradecerte, lo que soy hoy te lo debo a ti, gracias por todos los consejos, por siempre regañarme cuando lo llevaba, sin tu apoyo no hubiese podido llegar a lo que soy hoy. Mil gracias Frank.*

*A nuestro Comandante Fidel Castro Ruz y a la Revolución Cubana por darme la oportunidad de convertirme en profesional.*

# Dedicatoria

*Dedicatoria de Daneilys Pérez Villegas*

*A mi madre que con esfuerzo y desvelo ha dedicado su vida para cultivarme de valores humanos y forjarme como una mujer de bien. Ella que ha sentado sus esperanzas en verme convertida en profesional y nunca ha dudado de mis capacidades y espíritu de lucha, mami no te defraudé. Te amo.*

*A mis abuelos por darme tanto amor, confianza y apoyo en toda mi vida y principalmente en estos cinco años de carrera, sin su apoyo incondicional mi sueño que es el de ellos también no se hubiese realizado. Para ellos mi eterna gratitud.*

*A mi hermanita, mi chiquitica linda, eres el tesoro más preciado que tengo en esta vida y siempre ha estado ahí apoyándome, te amo Rosi y si no te tuviera en mi vida no hubiese logrado este sueño, esto va dedicado a ti. Eres lo más grande de mi vida nunca lo dudes.*

*Dedicatoria de Yonnis Turro Alfaro*

*A mi madre, por sus días y noches de entrega, por su apoyo total, demostrando su amor incondicional toda mi vida.*
*A mi padre, por haberme enseñado que la superación constante es el camino más agradable y excitante de ser cada día un ser humano mejor.*
*A mis hermanas porque significan mucho para mi.*

*Los quiero mucho son mi razón de ser*

**Resumen**

El presente trabajo forma parte del proyecto de investigación Plataforma de Servicios Bioinformáticos (PSBio) que desarrolla el departamento de Bioinformática del Centro de Tecnología de Gestión de Datos (DATEC). La misma tiene como propósito brindar servicios Bioinformáticos que puedan ser consumidos por aplicaciones o usuarios a través de un portal web. Como resultado de la investigación se obtuvo un portlet que consume servicios web para el manejo de datos ortológicos en formato FASTA, RoundUP, ARFF y SPSS el cual podrá ser desplegado en el contenedor de portlet del Portal. Para guiar el proceso de desarrollo de la aplicación se utilizó OpenUP como metodología de desarrollo, Liferay como contenedor de portlet y Vaadin como marco de trabajo y el protocolo SOAP para la comunicación con la capa de servicio. Una vez concluido el desarrollo del portlet se realizaron las pruebas funcionales las cuales permitieron validar el correcto funcionamiento del sistema.

**Palabras clave:** portlet, Bioinformática, genómica y ortología.

# *Índice de Contenido*

Índice de Contenido

# Índice de Contenido

# Índice de Contenido

Índice de tablas y figuras

# *Índice tablas y figuras*

## Introducción

Como consecuencia del creciente desarrollo tecnológico alcanzado en la actualidad, muchas instituciones y empresas han tenido la necesidad de automatizar sus procesos. Esto permite un aumento considerable en la calidad y rapidez de las actividades que se realizan en cada una de estas entidades, ya que se pueden almacenar, procesar y consultar grandes volúmenes de información.

Una de las ramas científicas más beneficiada con este desarrollo es la Biología, la cual utiliza la informática para recopilar, almacenar, analizar y combinar datos biológicos, dando lugar a una nueva disciplina: la Bioinformática, que constituye el área de investigación interdisciplinaria entre las ciencias biológicas y computacionales. (1)

La Bioinformática proporciona herramientas y recursos para favorecer las investigaciones relacionadas con las siguientes áreas: ciencias biomédicas, ciencias de la computación, matemáticas, física, estadística, biología molecular y genética. Tiene como objetivo final aumentar el conocimiento sobre la información biológica comprendida dentro del volumen de datos y obtener una percepción más clara de la biología fundamental de los organismos. (1)

Dentro de las ramas de la Bioinformática más abordadas por la comunidad científica se encuentra la homología biológica, la cual se encarga del estudio entre órganos determinados de dos especies diferentes, cuando ambos derivan del órgano correspondiente de su ancestro común. Los genes homólogos se clasifican en dos grupos principales: ortólogos y parálogos. Los genes parálogos son los que se encuentran separados luego de un proceso de duplicación en el mismo genoma. Los genes ortólogos son los que se encuentran en dos especies surgidas de un ancestro común luego de un evento de especiación y tienen una alta similitud tanto funcional como estructural. (2)

El estudio de los grupos de genes ortólogos es sumamente importante para el descubrimiento de funciones semejantes en diferentes especies y la comprensión de las mismas, cuyo fin es obtener patrones evolutivos necesarios en la predicción y prevención de enfermedades patológicas y genéticas. En dicho análisis se genera una enorme cantidad de datos los cuales son necesarios almacenar en una base de datos para una posterior consulta.

# *Introducción*

En la actualidad existe un gran número de especialistas que realizan estudios sobre la creación de algoritmos y técnicas para obtener grupos de ortologías entre especies, los cuales son almacenados en grandes bancos de datos. Algunas de las bases de datos que existen para el manejo de los grupos de ortología son: *Inparanoid*, que pertenece al centro de Bioinformática de *Stockholm* SBC (*Stockholm Bioinformatics Center* por sus siglas en ingles), el banco *Ensembl* Compara del Instituto de Bioinformática Europeo y *HomoloGene* del Centro Nacional para la Información Biotecnológica (NCBI). Estos bancos almacenan la información ortológica en el formato OrthoXML el cual, por su complejidad estructural, no es reconocido por las distintas aplicaciones de las disciplinas genómica y evolución molecular pues estas utilizan ficheros de menor complejidad como son los de tipo FASTA, ARFF, RoundUP y SPSS.

Teniendo en cuenta la importancia que representa para Cuba el desarrollo científico en áreas emergentes del conocimiento como la Bioinformática, se le dio la tarea de su estudio a un conjunto de especialistas de los principales polos científicos del país, como el Centro de Ingeniería Genética y Biotecnología (CIGB) y el Centro de Inmunología Molecular (CIM), unidos a universidades como la Universidad de la Habana (UH), Universidad Central Martha Abreu de las Villas (UCLV) y la Universidad de las Ciencias Informáticas (UCI).

En la UCI existen varios centros de desarrollo entre ellos el Centro de Tecnología de Gestión de Datos (DATEC) cuenta con el Departamento de Bioinformática. En este departamento existen varias herramientas enfocadas al procesamiento de datos biológicos entre las que se encuentra: BioSyS para la simulación de sistemas biológicos a través de ecuaciones diferenciales; T-Arenal, para realizar cálculos de manera distribuida y siRNADesign para el diseño de sondas que silencian genes patógenos humanos. Dicho departamento en conjunto con la UCLV se desarrolló Gregor, sistema que permite manejar información sobre grupos de ortología genética.

Para integrar estos sistemas, se crea la Plataforma de Servicios Bioinformáticos (PSBio), la cual tiene como propósito brindar servicios que puedan ser consumidos por aplicaciones o usuarios a través de un portal web, permitiendo a especialistas del país el acceso a las herramientas y bases de datos desarrolladas por el departamento; así como la formulación, composición y ejecución de problemas para la visualización y anotación de los resultados.

Gregor, al ser una herramienta de escritorio, no puede ser integrada a la Plataforma de Servicios Bioinformáticos, por lo que no está al alcance de los especialistas del país, la posibilidad de obtener datos de ortología en ficheros de menor complejidad estructural.

Por la situación antes expuesta se plantea como **problema de la investigación:** ¿Cómo obtener información de ortologías genéticas en ficheros FASTA, SPSS, ARFF y RoundUP de manera accesible para los especialistas a partir de base datos relacionales?

Se define como **objeto de estudio** de la investigación: las aplicaciones informáticas para la gestión de información para la ortología genética. El **campo de acción** ha sido enmarcado en: las aplicaciones que consumen servicios web para la gestión de información de ortología genética.

El **objetivo general** de la investigación es**:** Desarrollar un portlet para la Plataforma de Servicios Bioinformáticos que permita consultar información genética de los grupos de ortología almacenados en bases de datos relacionales. En correspondencia con el objetivo general se desglosan los siguientes **objetivos específicos:**

1. Elaborar el marco teórico de la investigación sobre las herramientas y tecnologías existentes a nivel nacional y mundial para la gestión de información de ortología genética.
2. Realizar el diseño del portlet para la gestión de información genética de los grupos de ortología.
3. Implementar el portlet para la gestión de información genética de los grupos de ortología.
4. Validar el portlet implementado a partir de pruebas funcionales.

Para dar cumplimiento al objetivo planteado se proponen las siguientes **tareas de investigación**:

1. Revisión del estado del arte acerca de las aplicaciones en Internet que manejan información sobre ortología genética así como el consumo de servicios web en Internet.
2. Definición de las funcionalidades con las que debe contar la aplicación.
3. Elaboración de los diagramas de Casos de Uso del Sistema (CUS por sus siglas) para definir los componentes de la implementación de cada una de las funcionalidades del sistema.

4. Elaboración del diagrama de clases del diseño para definir las clases con que contará el sistema.
5. Descripción de la arquitectura.
6. Elaboración del diagrama de despliegue.
7. Elaboración del diagrama de componentes.
8. Implementación del portlet para la gestión de información genética de los grupos de ortología de diferentes especies.
9. Implementar los servicios que permitan interactuar con la Plataforma de Servicios Bioinformáticos.
10. Aplicación de pruebas de caja negra a la solución implementada para verificar el correcto funcionamiento de la misma.

La presente investigación está compuesta por cuatro capítulos, estructurados de la siguiente manera:

**Capítulo 1. Fundamentos Teóricos:** En este capítulo se presentan los elementos teóricos que sirven de base a la investigación del problema planteado, analizando los principales conceptos relacionados con el objeto de estudio. Se realiza un análisis de otras soluciones existentes y se presentan las herramientas y metodología a utilizar para el desarrollo del sistema propuesto.

**Capítulo 2. Características del sistema:** En este capítulo se profundiza sobre las funcionalidades que debe tener el sistema. Se presentan los requisitos funcionales y no funcionales. Además se muestran los casos de usos del sistema y su descripción, así como los actores que interactúan con el sistema.

**Capítulo 3. Diseño de la aplicación:** Este capítulo describe el diseño de la solución propuesta, se brinda una descripción de los estilos y patrones arquitectónicos más utilizados y la aplicación de los mismos para la confección de los diagramas de interacción de clases y el modelo del diseño del sistema.

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

## Capítulo 1: Fundamentos teóricos

El presente capítulo abarca los principales conceptos que comprenden la fundamentación teórica de la presente investigación y que permiten conformar una idea futura de los resultados esperados. Se exponen soluciones existentes en el ámbito universal sobre el manejo de datos ortológicos. Finalmente se justifica la selección de la metodología que guiará el proceso de desarrollo de la solución, mediante una caracterización de la misma y se incluyen las tecnologías y herramientas que servirán de sustento a la investigación.

### 1.1. Conceptos asociados al dominio del problema

A continuación se abordan algunos conceptos esenciales que permitirán un mayor entendimiento del dominio del problema planteado, lo cual facilitará la elaboración de la propuesta de solución a dicho problema.

#### 1.1.1 Base de datos relacional

Una base de datos relacional, es donde todos los datos visibles al usuario están organizados estrictamente como tablas de valores y todas las operaciones de la base de datos operan sobre estas tablas. El modelo relacional representa un sistema de bases de datos en un nivel de abstracción un tanto alejado de los detalles de la máquina subyacente. (3)

**Características principales de los archivos relacionales:** (3)

- Cada archivo contiene solo un tipo de registro.
- Los campos no tienen un orden específico.
- Los registros no tienen un orden específico.
- Cada campo tiene un solo valor.
- Los registros pueden poseer un campo identificador único (o combinación de campos) llamada clave primaria.

El estudio realizado permitió entender cómo funcionan y organizan los datos en las bases de datos diseñadas en el departamento de Bioinformática de la UCI, sobre las cuales va a trabajar el equipo del

presente trabajo, para consultar la información de los genes ortólogos que estas almacenan.

### 1.1.2 Plataforma de Servicios Bioinformáticos de la UCI

El proyecto PSBio es un sistema que integra un conjunto de herramientas de uso común en la Bioinformática, así como los productos desarrollados por el departamento, con el propósito de brindar servicios que puedan ser consumidos por aplicaciones o usuarios a través de un portal web. La solución informática a desarrollar tiene una arquitectura orientada a servicios y de cara al usuario mediante interfaces web, implementadas en forma de portlets, dentro del portal de gestión de contenidos Liferay. (4)

Entre sus principales ventajas se encuentra que los productos desarrollados por el departamento de Bioinformática están desagregados en forma de servicios, permitiendo que se acoplen e integren mejor las funcionalidades implementadas para dar respuesta a un flujo de trabajo de un investigador. Además, debido al gran volumen de datos que se manejan en el campo de la Bioinformática, cada servicio a implementar garantizará acceder a un clúster de estaciones de trabajo con el propósito de realizar cálculos intensos y aumentar la capacidad de almacenamiento. (4)

En PSBio igualmente se podrían implementar nuevas funcionalidades en forma de servicios que encapsulen la gran variedad de aplicaciones que existen en el mundo de la Bioinformática orientadas a: alineamiento de secuencias, análisis de evolución molecular, procesamiento de datos biológicos como proteínas, nucleótidos y genomas, por solo citar algunos ejemplos. Finalmente, todos estos servicios estarían a disposición de los investigadores de distintos centros científicos del país a través de un Portal Web, que permita y facilite la interacción entre las funcionalidades implementadas y los usuarios finales. (4)

### 1.1.3 Portal de Servicios Bioinformáticos de la UCI

El Portal de Servicios Bioinformáticos es un portal desarrollado sobre el contenedor de portlets, Liferay Portal, que provee un ambiente amigable para la definición y la ejecución de análisis en la esfera de la Bioinformática. (4)

Dentro de sus objetivos se encuentran desarrollar una capa de servicios bioinformáticos que encapsulen herramientas informáticas de uso común en el área de la Bioinformática y respondan a las funcionalidades

de los productos de software desarrollados por el departamento. Además a la capa de servicios bioinformáticos T-Arenal y el sistema de almacenamiento distribuido para garantizar el soporte tecnológico de la solución.

### 1.1.4 Liferay Portal 6.0.5

Es un portal de gestión de portlets, de código abierto e implementado en Java. Se puede desplegar en servidores como Apache Tomcat, es multiplataforma lo cual permite ejecutarse en cualquier sistema operativo. Liferay es un proyecto de código abierto que usa licencia LGPL. Provee la personalización del entorno de usuario mediante plantillas y temas que pueden ser instalados a partir de un archivo con extensión WAR. (5)

### 1.1.5 Portlet

Un portlet es un componente web basado en la tecnología Java, que brinda una vista de la información del sistema. Estos componentes producen fragmentos de lenguaje de marcado que los dispone para ser agregados dentro de páginas en portales. (6) Como se muestra en la Figura 1.


**Figura 1:** Anatomía de un portlet

Para que un portlet pueda mostrarse, debe existir un contenedor de portlets que cumpla con las especificaciones de peticiones implementadas por java (JSR, por sus siglas en inglés) 168. Un contenedor de portlets es el entorno de ejecución de un portlet, este maneja su ciclo de vida, sus características, como deben lucir y manipula las peticiones desde el portal. (6)

Una de las potencialidades de los portlets es su capacidad de transportarse sin necesidad de cambiar su

código. Estos se compilan en un Archivo de Aplicación Web (WAR por sus siglas en inglés), el cual puede ser desplegado en otro contenedor de portlets que sea compatible con las JSR 168. (6)

PSBio utiliza la tecnología de los portlets en las aplicaciones que lo integran. La finalidad del presente trabajo es posibilitar al portal obtener datos de ortologías, por lo que se hace necesario desarrollar un portlet para poder agregar esta funcionalidad al mismo. Este portal utilizará como contenedor de portlet el Liferay. (6)

### 1.1.6 Servicio Web

Los servicios web son tecnologías que utilizan un conjunto de protocolos y estándares que sirven para intercambiar datos entre aplicaciones. Permiten la intercomunicación entre sistemas de cualquier plataforma y se utilizan en varios escenarios de integración. Una vez que el servicio web se ha desplegado, este puede ser invocado por otras aplicaciones. (7)

El esquema de funcionamiento de los servicios web, requiere de tres elementos fundamentales: (7)

1. Un proveedor del servicio web se encarga de diseñar, desarrollar e implementar y además de la disponibilidad de uso, ya sea dentro de la misma organización o en público.

2. Un consumidor del servicio se encarga de acceder al componente para utilizar los servicios que éste presta.

3. Un agente de servicio, utilizado como enlace entre proveedor y consumidor para efectos de publicación, búsqueda y localización del servicio.

La descripción antes expuesta permitió entender la función que cumplen los servicios web y los beneficios que aportan, razón por la cual se decide desarrollar los mismos para el manejo de datos de genes ortólogos y para que puedan ser consumidos por cualquier sistema sin importar la plataforma en que estos se encuentren.

## 1.2 Repositorios que manejan información de genes ortólogos

En esta sección se presentarán diferentes repositorios que manejan información sobre genes y proteínas ortólogas. Se detallarán sus características, así como el tipo de información que contienen. En concreto

los repositorios analizados son: KOG, OrthoMCL e Inparanoid. (8)

**InParanoid** es una base de datos que almacena información sobre millones de secuencias de especies diferentes obtenidas de distintos repositorios como, Ensembl, WormBas, VectorBase, PlasmoDB, CryptoDB. La mayoría de sus técnicas para descubrir genes ortólogos tienen éxito cuando existe una copia de un gen en cada especie analizada. Mediante la distinción entre inparalogs y outparalogs, el algoritmo utilizado para la construcción del repositorio puede identificar operaciones entre genes ortólogos, de uno a muchos y de muchos a muchos. (8)

- Inparalogs: secuencias genómicas de una especie que han sido duplicadas después de un evento de especiación. Así pueden ser considerados como genes ortólogos de una o más secuencias en otra especie, ya que provienen de un ancestro común.

- Outparalogs: secuencias genómicas de una especie, que son duplicadas antes de un evento de especiación, por lo que no pueden ser considerados genes ortólogos.

InParanoid compara las secuencias de las distintas especies de dos en dos. Por lo tanto, el repositorio proporciona el resultado de comparar las secuencias de una especie con las de otra en ficheros separados. Estos ficheros indican en el nombre del fichero las especies comparadas. (8)

Por su parte el repositorio **KOG** está compuesto por una colección de grupos de genes ortólogos de entre siete especies eucariotas. En concreto, de las siete especies, tres son animales: Caenorhabditis elegans, Drosophila melanogaster y Homo sapiens; una planta: Arabidpsis thaliana; dos hongos: Saccharomyces cerevisiae y Schizosaccharomyces pombe; y un microorganismo intracelular parásito: Encephalitozoon cuniculi. La información de este repositorio está disponible en ficheros de texto semiestructurados. (8)

El fichero **KOG** contiene la información sobre los grupos de genes ortólogos. Cada grupo posee una clave única, una descripción, un conjunto de categorías funcionales identificadas cada una por un carácter y un conjunto de proteínas o genes junto con su especie. Los genes ortólogos están definidos mediante un código de tres letras que identifica a la especie y el identificador del ortólogo. (8)

Por último se describe **OrthoMCL**, el cual contiene datos de proteínas ortólogas. La primera versión de este repositorio fue publicada en 2005 y contenía 70 388 grupos de proteínas ortólogas de 55 especies

diferentes. La segunda versión del repositorio fue lanzada en 2008, y contenía información de 87 especies, así como mejoras en el proceso de actualización que permitía reducir el tiempo de la operación. La tercera versión contenía 128 especies, la cuarta 138 y, por último, la versión más reciente dispone de información de 150 especies. La última versión de OrthoMCL, contiene 124 740 grupos de genes ortólogos, con 1 192 387 proteínas ortólogas, lo que supone un gran aumento de información respecto a la primera versión. (8)

La información de este repositorio sobre los grupos de genes ortólogos está disponible en ficheros de texto semiestructurados. El fichero groups_OrthoMCL-5.txt.gz contiene la información sobre los grupos de genes ortólogos. Cada grupo posee una clave única y el conjunto de pares especie–proteína perteneciente al grupo ortólogo. La especie correspondiente a cada ortólogo está representada por cuatro caracteres. (8)

Aunque estas herramientas no dan solución al problema planteado en el presente trabajo, pues brindan su información en ficheros de tipo XML, conocidos como OrthoXML y SeqXML, aportaron una idea al equipo de trabajo de cómo implementar el sistema para dar solución al manejo de datos de genes ortólogos. (8)

## 1.3 Fichero

Un fichero o un archivo, en términos informáticos, es una colección de datos relacionados los cuales son almacenados y recuperados por un nombre asignado. Un fichero puede incluir información para iniciar un programa, contener texto o gráficos, o procesar una serie de comandos. En un término genérico para un tipo de objeto que hace alusión a un fichero de base de datos, un fichero de dispositivo, o un fichero almacenado. También se conoce como fichero a un nombrado grupo de registros almacenados o procesados como una unidad. (9) A continuación se describirán los ficheros que debe construir el sistema a desarrollar, y así entender su estructura, para crear algoritmos que construyan los mismos.

### 1.3.1 Fichero FASTA

Un fichero de secuencia bajo formato FASTA comienza con una descripción en una sola línea (línea de cabecera), seguida por líneas de datos de secuencia. La línea de descripción se distingue de los datos de secuencia por un símbolo '>' (mayor que) en la primera columna. La palabra siguiente a este símbolo es el identificador de la secuencia, y el resto de la línea es la descripción (ambos son opcionales). No debería

existir espacio entre el '>' y la primera letra del identificador. Se recomienda que todas las líneas de texto sean menores de 80 caracteres. La secuencia termina si aparece otra línea comenzando con el símbolo '>'; esto indica el comienzo de otra secuencia. (10) Un ejemplo simple de una secuencia en el formato FASTA se muestra en la Figura 2.

```
>read number 1
G C A T A A G C C A G G C G C C A C G C C T T G A C T A T A
A C C I P F G T G T C G T A T T A C A A C C G A C C A T G G
G C A T A A G C C A G G C G C C A C G C C T T G A C T A T A
C T T G T A C C T A T T G A A G A C T T A C G T A G C T T A
```

**Figura 2:** Estructura del formato FASTA

### 1.3.2 Fichero ARFF

Un ARFF es un archivo de texto de Código Estándar Estadounidense para el Intercambio de Información (ASCII por sus siglas en ingles) que describe una lista de casos que comparten un conjunto de atributos. Los Archivos ARFF tienen dos secciones distintas. La primera sección es la información de cabecera, y la segunda la información de los datos. Como se muestra en la Figura 3, el encabezado del archivo ARFF contiene el nombre de la relación, una lista de los atributos (las columnas de los datos), y sus tipos. El aspecto de los datos del archivo se muestra en la Figura 4. (11)

```
1.   % 1. Title: Iris Plants Database
2.       %
3.       % 2. Sources:
4.       %      (a) Creator: R.A. Fisher
5.       %      (b) Donor: Michael Marshall (MARSHALL%PLU@io.arc.nasa.gov)
6.       %      (c) Date: July, 1988
7.       %
8.       @RELATION iris
9.
10.      @ATTRIBUTE sepallength  NUMERIC
11.      @ATTRIBUTE sepalwidth   NUMERIC
12.      @ATTRIBUTE petallength  NUMERIC
13.      @ATTRIBUTE petalwidth   NUMERIC
14.      @ATTRIBUTE class        {Iris-setosa,Iris-versicolor,Iris-virginica}
```

**Figura 3:** Encabezado del archivo ARFF

```
1. @DATA
2.      5.1,3.5,1.4,0.2,Iris-setosa
3.      4.9,3.0,1.4,0.2,Iris-setosa
4.      4.7,3.2,1.3,0.2,Iris-setosa
5.      4.6,3.1,1.5,0.2,Iris-setosa
6.      5.0,3.6,1.4,0.2,Iris-setosa
7.      5.4,3.9,1.7,0.4,Iris-setosa
8.      4.6,3.4,1.4,0.3,Iris-setosa
9.      5.0,3.4,1.5,0.2,Iris-setosa
10.     4.4,2.9,1.4,0.2,Iris-setosa
11.     4.9,3.1,1.5,0.1,Iris-setosa
```

**Figura 4:** Datos del archivo ARFF

### 1.3.3 Fichero del SPSS

El programa SPSS (paquete estadístico aplicado a las ciencias sociales) constituye un programa modular que implementa gran variedad de temas estadísticos orientados al ámbito de las ciencias sociales desde hace más de 30 años. Actualmente, cubre casi todas las necesidades del cálculo estadístico de los investigadores y profesionales, no sólo del campo de las ciencias sociales sino también de las humanas y de las biomédicas. (12)

El SPSS trabaja con los datos previamente grabados en un fichero al que se denomina fichero de datos. Cuando se ejecuta el programa SPSS tal fichero deberá estar activo para que sobre él se lleven a cabo los cálculos oportunos. Una vez que el fichero está activo se puede ver en la ventana del editor de datos en forma de una rejilla en la que sus filas son los individuos o casos y las columnas son las variables objeto del estudio. (12) En la Figura 5 se muestran la estructura del fichero y como se muestra en la herramienta correspondientemente.

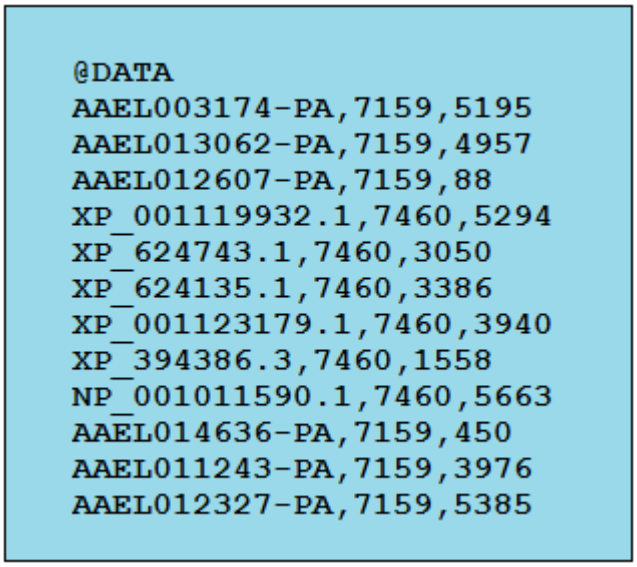



**Figura 5:** Estructura del formato SPSS

El estudio de la estructura de estos ficheros permitió comprender la forma en la que organizan la información contenida en ellos, además de brindar la posibilidad de crear algoritmos que soporten dichos ficheros.

### 1.4 Arquitectura del Software

La arquitectura del software es el conjunto de decisiones significativas sobre la organización del sistema, la selección de los elementos estructurales con los que se componen y sus interfaces, junto con su comportamiento tal como se especifica en las colaboraciones entre esos elementos, la composición de esos elementos estructurales y de comportamiento en subsistemas progresivamente más amplios, y el estilo de arquitectura que guía esta organización. (13)

Independientemente de la definición, el tema común en todas las definiciones de arquitectura del software es que tiene que ver con la gran escala, la organización, estilos, patrones, responsabilidades, colaboraciones, conexiones y motivaciones de un sistema y los subsistemas importantes. En el desarrollo de software, arquitectura se considera tanto un nombre como un verbo. (13)

Como nombre: la arquitectura comprende como indica la definición anterior la organización y estructura de los elementos importantes del sistema. Más allá de esta definición estática, incluye el comportamiento del sistema, especialmente en función de responsabilidades de gran escala de los sistemas y subsistemas, y sus colaboraciones. En cuanto a una descripción, la arquitectura comprende las motivaciones o fundamentos de por qué el sistema está diseñado de la forma que está. (13)

Como verbo: la arquitectura es parte investigación y parte trabajo de diseño; por claridad, el término es mejor que se califique como en investigación arquitectural o diseño arquitectural. (13)

La investigación arquitectural implica la identificación de aquellos requisitos funcionales y no funcionales que “influyen o deberían influir de manera” significativa en el diseño del sistema, como rendimiento, coste, mantenimiento, y puntos de evolución. Ampliamente, se trata del análisis de requisitos centrado en aquellos que tienen una influencia especial en las decisiones de diseño del sistema más importantes. (13)

El diseño arquitectural es la resolución de estas influencias y requisitos en el diseño del software, el hardware y las redes, operaciones, políticas, entre otras. (13)

El portlet para el manejo de datos ortológicos hace uso del patrón arquitectónico por capas y los patrones de diseño. También se guía por las arquitecturas Cliente-Servidor y Orientada a Servicios. La primera se manifiesta en el paradigma petición-respuesta, dado que es una aplicación web a la que se accede por el protocolo HTTP. La segunda está representada en los servicios web implementados que están disponibles a través del protocolo SOAP.

### 1.4.1 Patrón arquitectónico por Capas

Un patrón de arquitectura de software describe un problema particular y recurrente del diseño, surge en un contexto específico, y presenta un esquema genérico y probado de su solución. Define la estructura de los sistemas de software, los cuales a su vez se componen de subsistemas con sus responsabilidades. (14)

Para crear un buen diseño del sistema a desarrollar se decidió utilizar uno de los patrones arquitectónicos más conocidos para el diseño web, el patrón capas, el cual tiene como idea esencial organizar la estructura lógica de gran escala de un sistema en capas separadas de responsabilidades distintas y relacionadas. (14)

El patrón capas se relaciona con la arquitectura lógica; es decir, describe la organización conceptual de los elementos del diseño en grupos, independiente de su empaquetamiento o despliegue físico. Las capas definen un modelo general de N-niveles para la arquitectura lógica. (14)

La ventaja principal de este estilo es que el desarrollo se puede llevar a cabo en varios niveles y, en caso de que sobrevenga algún cambio sólo se ataca al nivel requerido sin tener que revisar entre código mezclado. Además, permite distribuir el trabajo de creación de una aplicación por niveles, de este modo, cada grupo de trabajo está totalmente abstraído del resto de los niveles. (14)

### 1.4.2 Arquitectura Orientada a Servicios.

A pesar de las grandes ventajas que poseen los servicios web tales como la usabilidad y la reutilización combinadas en la interoperabilidad, las cuales son las característica más importantes que proveen los mismos, estos presentan ciertas limitaciones y desventajas, como bajo nivel de abstracción, intercambio de grandes cantidades de datos sobre protocolos simples como HTTP, críticos problemas de seguridad y bajo rendimiento. Para aprovechar el potencial de los servicios web estos deben ser diseñados cuidadosamente bajo los conceptos y principios de la orientación a servicios propuestos por la Arquitectura Orientada a Servicios (SOA, por sus siglas en inglés). (14)

SOA, es un concepto de arquitectura de software que define la utilización de servicios para dar soporte a los requisitos del negocio. Permite la creación de sistemas de información altamente escalables que reflejan el negocio de la organización, a su vez brinda una forma bien definida de exposición e invocación de servicios, lo cual facilita la interacción entre diferentes sistemas propios o de terceros. (14)

SOA organiza las funcionalidades en servicios web que pueden ser consumidos por aplicaciones clientes, las cuales intercambian información con el proveedor de servicios mediante el Protocolo Simple de Acceso a Objeto (SOAP por sus siglas en inglés). El proveedor publica los servicios en un registro denominado UDDI (por sus siglas en inglés Descripción, Descubrimiento e Integración Universales) que es el catálogo de negocios de Internet, el cual especifica mediante el Lenguaje de Descripción de Servicios Web (WSDL por sus siglas en inglés) las funciones y los parámetros necesarios para poder utilizar el servicio web. (14) Los componentes básicos de SOA se representan en la Figura 6.

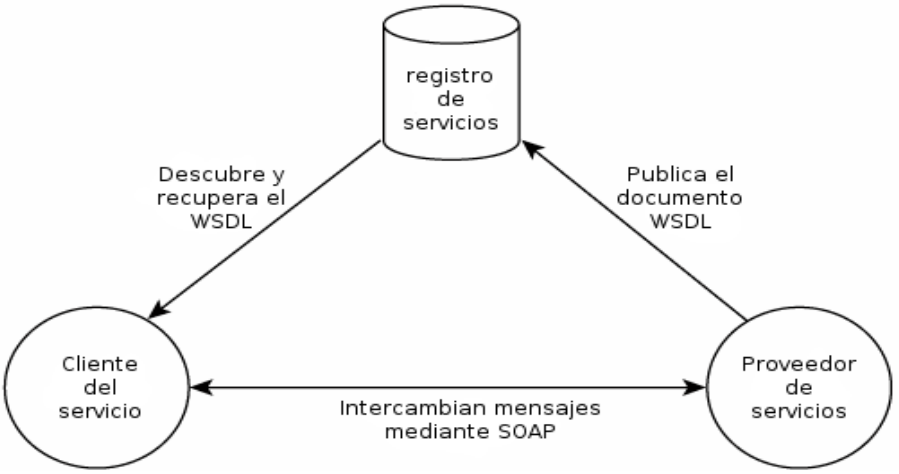



**Figura 6:** Componentes básicos de una arquitectura SOA

### 1.4.3 Patrones de diseño. GRASP

Un patrón de diseño es una solución repetible a un problema recurrente en el diseño de software. Esta solución no es un diseño terminado que puede traducirse directamente a código, sino más bien una descripción sobre cómo resolver el problema.

En la realización del diseño para la aplicación informática a implementar, se utilizaron los patrones generales de software para asignar responsabilidades (GRASP por sus siglas en ingles).

Los patrones GRASP constituyen un apoyo para la enseñanza que ayuda a entender el diseño de objetos esenciales, y aplica el razonamiento para el diseño de una forma sistemática, racional y explicable.

**Tabla 1** Los patrones GRASP y sus funciones

| | |
|---|---|
| **Experto** | Asignar una responsabilidad al experto en información: la clase que cuenta con la información necesaria para cumplir la responsabilidad. |
| **Creador** | Explica que clase es la encargada de crear objetos, en determinados escenarios de ejecución. |

| | |
|---|---|
| **Bajo acoplamiento** | Se basa en asignar una responsabilidad para mantener bajo acoplamiento. El bajo acoplamiento es cuando las clases del sistema tienen pocas dependencias entre ellas, solo las necesarias. |
| **Alta Cohesión** | Propone asignar la responsabilidad de manera que la complejidad se mantenga dentro de límites manejables asumiendo solamente las responsabilidades que deben manejar, evadiendo un trabajo excesivo. |
| **Controlador** | Es un objeto de interfaz no destinada al usuario que se encarga de manejar un evento del sistema a una clase que represente un sistema global. Define además el método de su operación. |

## 1.5 Metodología, Herramientas y Tecnologías

La solución propuesta requiere la selección de una metodología de desarrollo, que permita elaborar un marco de trabajo para la estructuración, planeación y control del proceso de desarrollo. Se seleccionan además las tecnologías y herramientas a utilizar para su construcción, teniendo en cuenta las características que debe tener el sistema y las restricciones de uso de las herramientas.

### 1.5.1 Metodología de desarrollo OpenUP

Para la realización de la solución se decide utilizar la metodología OpenUP pues es una metodología ágil apropiada para proyectos pequeños, permite disminuir las probabilidades de fracaso e incrementar las probabilidades de éxito. Posibilita detectar errores tempranos a través de un ciclo iterativo. Evita la elaboración de documentación, diagramas e iteraciones innecesarios. Tiene un enfoque centrado al cliente y con iteraciones cortas. Contiene los componentes básicos que pueden servir de base a procesos específicos y la mayoría de los elementos están declarados para fomentar el intercambio de información entre los equipos de desarrollo y mantener un entendimiento compartido del proyecto, sus objetivos, alcance y avances. (15) En la Figura 7 se muestran las fases de OpenUP.

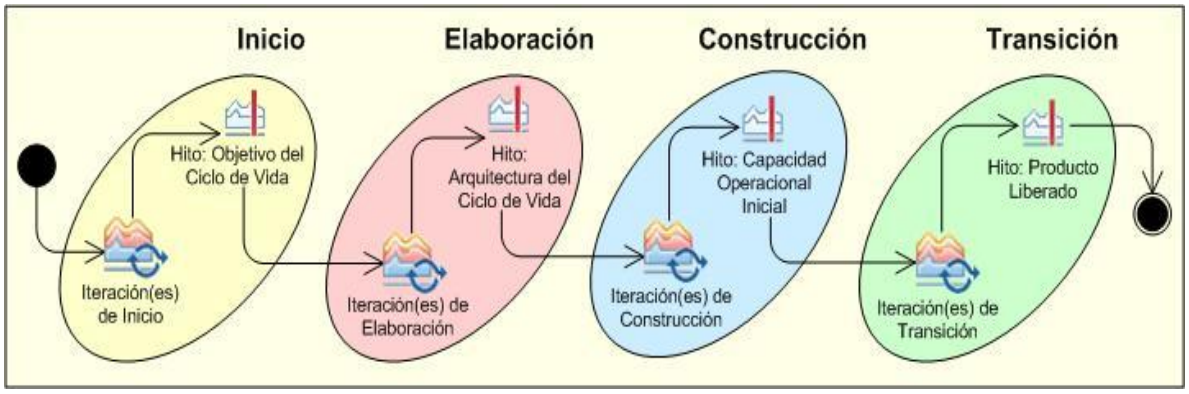



**Figura 7:** Fases de OpenUP

**Inicio:** El propósito de esta fase es establecer una visión común inicial de los objetivos del proyecto, determinar si es viable y decidir si merece la pena llevar a cabo algunas investigaciones serias en la fase de elaboración. (15)

**Elaboración:** Sus objetivos fundamentales son: obtener un entendimiento con mayor nivel de detalle de los requisitos, diseñar, implementar, mitigar riesgos y lograr estimaciones de costos y calendarios más precisos. La elaboración termina cuando se resuelven las cuestiones de alto riesgo, se completa el esqueleto de la arquitectura y se entiende la mayoría de los requisitos. (15)

**Construcción:** Su propósito es completar el desarrollo del sistema basado en la arquitectura definida y asegurar que el software está listo para ser entregado a la comunidad de usuarios. La construcción termina cuando se considera que el sistema está preparado para el despliegue operacional, y se complementan todos los materiales de soporte, como las guías de usuario y materiales de aprendizaje.

**Transición:** Sus objetivos fundamentales son realizar pruebas beta para determinar si se alcanzaron las expectativas de los usuarios, alcanzar la concordancia con los *stakeholders* de que el producto está terminado y mejorar la *performance* futura. (15)

OpenUP utiliza el Lenguaje Unificado de Modelado (UML por sus siglas en inglés) para el modelado de sistemas de software el cual es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema de software. OpenUP representa un número de modelos de desarrollo basados en componentes que han sido propuestos en la industria. Haciendo uso de UML define los componentes que se utilizarán para construir el sistema y las interfaces que conectarán los componentes. (15)

### 1.5.2 Lenguaje de Modelado: UML 2.1

UML es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema. UML ofrece un estándar para describir un plano del sistema (modelo), incluyendo aspectos conceptuales tales como procesos de negocio, funciones del sistema y aspectos concretos como expresiones de lenguajes de programación y esquemas de base de datos. Su principal objetivo es representar el conocimiento acerca de los sistemas que se pretenden construir y las decisiones tomadas durante su desarrollo.

Existen varias herramientas CASE que utilizan UML como lenguaje de modelado para generar los artefactos correspondientes a las diferentes etapas de desarrollo. Una de ellas es Visual Paradigm, la cual permite la construcción de aplicaciones de calidad a través de un proceso de desarrollo bien documentado.

### 1.5.3 Herramienta de modelado Visual Paradigm 8.0

Visual Paradigm es una herramienta de modelado, fácil de usar, con soporte multiplataforma y que proporciona excelentes facilidades de interoperabilidad con otras aplicaciones. Está desarrollada para cubrir todo el ciclo del desarrollo de software, permitiendo la captura de requisitos, análisis, diseño e implementación. Tiene la capacidad de crear el esquema de clases a partir de una base de datos y crear la definición de base de datos a partir del esquema de las clases. (16)

Permite invertir código fuente de programas, archivos ejecutables y binarios en modelos UML al instante, creando de manera simple toda la documentación. Está diseñada para usuarios interesados en sistemas de software con el uso del acercamiento orientado a objeto, además apoya los estándares más recientes de las notaciones de Java y de UML. Incorpora el soporte para trabajo en equipo, que permite que varios desarrolladores trabajen a la vez en el mismo diagrama y vean en tiempo real los cambios hechos por sus compañeros. (16)

Se selecciona esta herramienta debido a que es la utilizada por el departamento de Bioinformática, por lo que su uso permite lograr la compatibilidad entre todos los diagramas generados por el desarrollo de la Plataforma de Servicios Bioinformáticos, además cuenta con versiones para el sistema operativo GNU/Linux, el cual es el utilizado por el equipo de desarrollo. Se encuentra disponible bajo varias licencias, cada una con diferentes funcionalidades, desde una versión Edición Comunitaria y con licencia

de software libre hasta una versión Empresarial con licencia comercial, la cual fue pagada por la universidad, lo que permite su uso sin incurrir en violaciones de licenciamiento de software. También está disponible para integrarse en los principales entornos de desarrollo (IDEs por sus siglas en inglés) como Eclipse. (16)

### 1.5.4 Entorno de desarrollo integrado. Eclipse Juno

Se decide utilizar Eclipse pues brinda un entorno de desarrollo amigable, automatiza muchas funciones a la hora de completar el código. Es un producto libre y gratuito sin restricciones de uso. Además es fácil de usar y de integrar sus distintos componentes y herramientas. Consume pocos recursos de hardware comparado con otros IDEs, por lo que se presenta con buen rendimiento y rapidez y brinda la posibilidad de desarrollar aplicaciones web. (17)

### 1.5.5 Plataforma Java 2 Enterprise Edition

Java 2 Enterprise Edition (J2EE) es la extensión de Java para el desarrollo web, la misma permite que las aplicaciones desarrolladas bajo ella puedan ser desplegadas en cualquier sistema operativo. Es una unidad de software funcional, ensamblada dentro de las aplicaciones Java con sus clases relacionadas y archivos que lo comunican con otros componentes. Dispone de varias herramientas de código abierto como IDEs, servidores y marcos de trabajo. (18)

Esta plataforma también brinda una serie de librerías que facilitan el trabajo de los desarrolladores, tales como Java Database Connectivity (JDBC), Enterprise Java Beans (EJB), Java Server Pages (JSP), Java Tag Library (JSTL) y Java Message Service (JMS). Cuenta con varios marcos de trabajo que facilitan el desarrollo de las aplicaciones, tales como Spring, Vaadin y Java Server Faces. Además tanto para la plataforma J2EE y para el lenguaje Java en general existen múltiples bibliografías en varios idiomas y se han creado comunidades que promueven la utilización de esta tecnología. (18)

Teniendo en cuenta lo planteado se decide utilizar la plataforma J2EE para el desarrollo de la solución, destacando como ventaja la reutilización de código. (18)

### 1.5.6 Marco de trabajo Vaadin 6.0

Vaadin es un marco de desarrollo de aplicaciones web del lado del servidor que permite a los desarrolladores crear interfaces de usuario de alta calidad con Java. Proporciona una biblioteca de listas para utilizar los componentes de interfaz de usuario y un marco para crear sus propios componentes. La atención se centra en la facilidad de uso, la reutilización, la extensibilidad y el cumplimiento de los requisitos de las grandes aplicaciones empresariales. (19)

Con el modelo de programación basado en el servidor, Vaadin se encarga de la gestión de la interfaz de usuario en el navegador y las comunicaciones entre el navegador y el servidor. Con el enfoque de Vaadin, no se necesita aprender y depurar tecnologías del navegador, tales como HTML o JavaScript. (19)

La Figura 8 ilustra la arquitectura básica de las aplicaciones web hechas con Vaadin, el cual consiste en un framework del lado del servidor y un motor del lado del cliente que se ejecuta en el navegador como un programa de JavaScript, renderizando la interfaz de usuario y entregando la interacción del usuario con el servidor. (19)

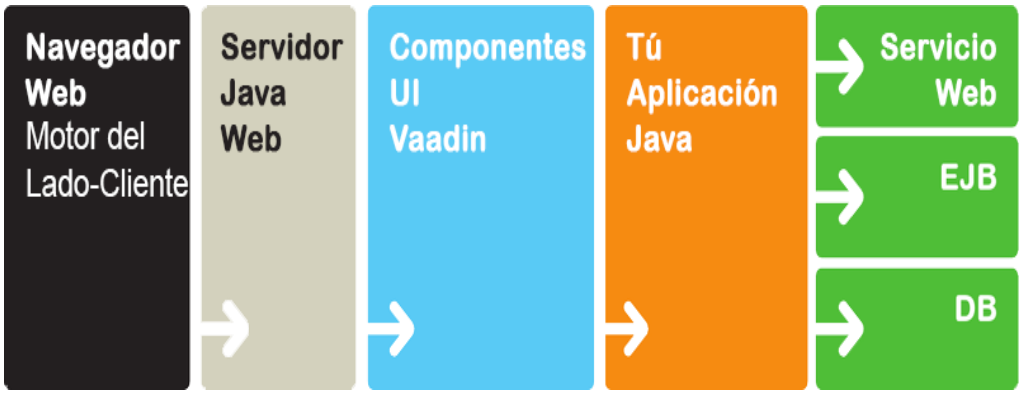



**Figura 8:** Arquitectura General de Vaadin

Este marco permite desarrollar componentes propios, que van desde botones, etiquetas, tablas y campos de texto hasta formas completas, campos para captura de fecha, notificaciones de errores y árboles

jerárquicos. Permite el desarrollo de aplicaciones web vistosas e interactivas sin requerir para ello plugins en el servidor web. Permite llevar nuestra aplicación desarrollada directamente como portlet a Liferay. (19)

Se escoge este marco de trabajo para la implementación del portlet, por las ventajas antes mencionadas además que su desarrollo se centra en hacer aplicaciones solamente programando en Java, no es necesario manejar Java Script ni HTML, permitiendo a los desarrolladores enfocarse principalmente en la lógica y dejar el HTML, Java Script y XML al Vaadin. (19)

### 1.5.7 Marco de trabajo Apache Axis 2

Apache Axis2 fue desarrollado con el objetivo de facilitar la implementación de servicios web. Este proyecto es una aplicación basada en Java, tanto del lado del cliente como del lado del servidor. Proporciona un completo modelo de objetos, para el procesamiento de mensajes, el cual es extensible y conveniente para el desarrollador. Tiene una arquitectura modular que facilita la tarea de añadir funcionalidad y dar soporte para nuevos servicios web. (20)

Además está equipado con la capacidad de despliegue de servicios web y manejadores mientras el sistema está activo y ejecutándose. En otras palabras, los nuevos servicios se pueden añadir al sistema sin tener que apagar el servidor. Simplemente se coloca el servicio web en el archivo de los servicios de directorio en el repositorio, y el modelo de despliegue se ocupa de desplegar automáticamente el servicio y hacer que el mismo esté disponible para su uso. (20)

Por todas las características descritas sobre este marco de trabajo, se decide utilizar para la implementación de los servicios del sistema a desarrollar.

### 1.5.8 Sistema gestor de base de datos PostgreSQL 8.4

PostgreSQL es un Sistema Gestor de Bases de Datos (SGBD) relacional orientado a objeto, distribuido bajo la licencia BSD. Constituye un potente gestor de base de datos de código abierto muy avanzado. Ofrece un control de concurrencia que mejora las operaciones de bloqueo y las transacciones en sistemas multiusuario. Soporta casi toda la sintaxis SQL, incluyendo subconsultas, transacciones, tipos y funciones definidas.

PostgreSQL se destaca por una amplia lista de prestaciones que le permiten competir con cualquier SGBD comercial entre las que se incluyen:

1 Cuenta con un conjunto de tipos de datos, permitiendo además su extensión mediante tipos y operadores definidos y programados por el usuario.

2 Su administración se basa en usuarios y privilegios.

3 Sus opciones de conectividad abarcan TCP/IP, sockets Unix y sockets NT, además de soportar completamente ODBC.

4 Los mensajes de error pueden estar en español y hacer ordenaciones correctas con palabras acentuadas o con la letra 'ñ'.

5 Es altamente confiable en cuanto a estabilidad se refiere.

6 Puede extenderse con librerías externas para soportar encriptación, búsquedas por similitud fonética.

7 Soporte para vistas, claves foráneas, integridad referencial, disparadores, procedimientos almacenados, subconsultas y casi todos los tipos y operadores soportados en SQL92 y SQL99.

8 Implementación de algunas extensiones de orientación a objetos. En PostgreSQL es posible definir un nuevo tipo de tabla a partir de otra previamente definida.

Todas estas características hacen que PostgreSQL ofrezca hoy en día un amplio y muy útil conjunto de funciones. La conectividad, la velocidad y seguridad, además de su sencilla interacción con el lenguaje de programación a utilizar para el desarrollo del sistema propuesto.

## 1.6 Conclusiones

En el presente capítulo, se analizaron los conceptos esenciales que permitirán un mayor entendimiento del dominio del problema planteado, lo cual facilitará la elaboración de la propuesta de solución a dicho problema. Además se describieron varias de las bases de datos bioinformáticas más utilizadas actualmente para el manejo de datos ortológicos, para ver cómo se manejan los datos ortológicos en el mundo permitiendo tener una idea de cómo debe funcionar el portlet que se debe implementar.

Se definió el uso de servicios web siguiendo una Arquitectura Orientada a Servicios, mediante el lenguaje de programación Java, aprovechando las características de la plataforma J2EE y utilizando el Apache Axis 2 para facilitar la implementación de los mismos. El proceso de desarrollo del software se guiará por la metodología OpenUP, utilizando como lenguaje de modelado a UML, el Visual Paradigm 8.0 para construir los diagramas correspondientes con la misma y para la implementación del sistema se utilizara como entorno de desarrollo Eclipse Juno.

## Capítulo 2: Características del sistema

El presente capítulo muestra el modelo de dominio con la relación entre los principales conceptos del negocio. Se especifican los requisitos funcionales que definen las capacidades o condiciones que el sistema debe ser capaz de realizar, y los requisitos no funcionales que definen las restricciones y propiedades del sistema. A partir de los requisitos funcionales se definen los casos de uso del sistema, los cuales son representados mediante el diagrama de casos de uso del sistema así como la descripción de los casos de uso.

### 2.1 Modelo de dominio

El modelo de dominio es un artefacto de la fase de elaboración, se representa en UML con un diagrama de clases en el que se pueden mostrar conceptos u objetos del dominio del problema, asociaciones entre las clases conceptuales o atributos de las clases conceptuales. No contiene conceptos propios de un sistema de software sino de la propia realidad física. (21)

Este modelo puede utilizarse para capturar y expresar el entendimiento ganado en un área bajo análisis como paso previo al diseño de un sistema, ya sea de software o de otro tipo. Similares a los mapas mentales utilizados en el aprendizaje, el modelo de dominio es utilizado por el analista como un medio para comprender el sector industrial o de negocios al cual el sistema va a servir. (21)

Además puede ser tomado como el punto de partida para el diseño del sistema. Esto es así pues cuando se realiza la programación orientada a objetos, se supone que el funcionamiento interno del software va a imitar en alguna medida a la realidad, por lo que el mapa de conceptos del modelo de domino constituye una primera versión del sistema. (21)

En esta fase (Elaboración) se determinó que los procesos del negocio no están claramente definidos, por esta razón se decide representar los conceptos que definen la situación real del sistema mediante un modelo de dominio facilitando a través de un vocabulario común que ayude a usuarios, clientes, desarrolladores e interesados a comprender el argumento principal del sistema como muestra la Figura 9.

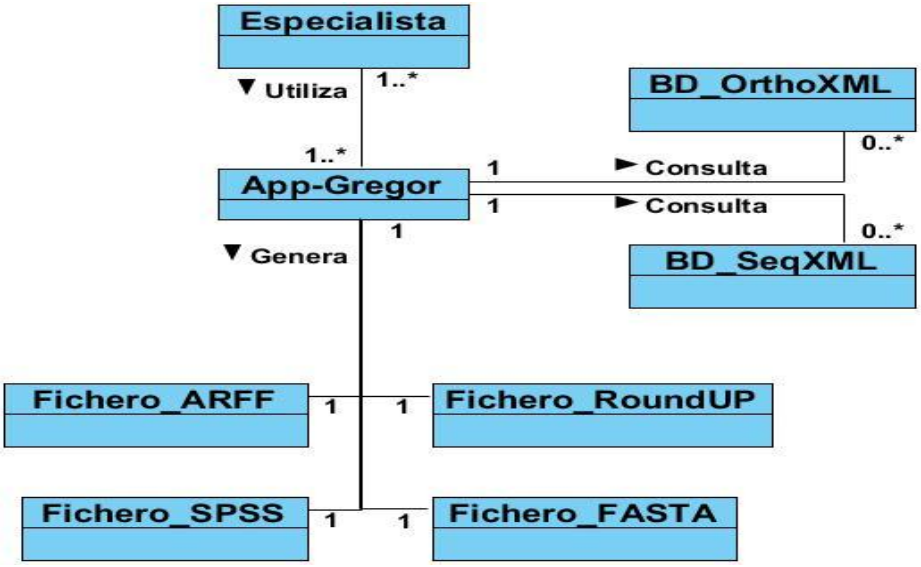



**Figura 9:** Diagrama de Clases del Modelo de Dominio de la aplicación

En este modelo de dominio se ven los conceptos asociados con la aplicación. El especialista utiliza la aplicación Gregor que es de escritorio la cual consulta dos base de datos para generar ficheros en distintos formatos como son ARFF, SPSS, RoundUP y FASTA.

**Tabla 2** Descripción de las clases del Modelo de Dominio

| | |
|---|---|
| **Especialista** | Persona que estudia los genes ortólogos. |
| **App-Gregor** | Herramienta de escritorio que brinda la información de datos ortológicos al especialista. |
| **BD_SeqXML** | Base de datos que guarda la información de los ficheros SeqXML. |
| **BD_OrthoXML** | Base de datos que guarda la información de los ficheros OrthoXML. |
| **Fichero_FASTA** | Fichero en formato FASTA con la información de los datos ortológicos |

| | |
|---|---|
| | generados por App-Gregor. |
| **Fichero_ARFF** | Fichero en formato ARFF con la información de los datos ortológicos generados por App-Gregor. |
| **Fichero_SPSS** | Fichero en formato SPSS con la información de los datos ortológicos generados por App-Gregor. |
| **Fichero_RoundUP** | Fichero en formato RoundUP con la información de los datos ortológicos generados por App-Gregor. |

## 2.2 Especificación de requisitos

El propósito general del flujo de trabajo requisitos es guiar el desarrollo hacia el sistema correcto. Dicho objetivo se consigue mediante una descripción de los requisitos del sistema suficientemente clara y exenta de ambigüedades, que permita llegar a un acuerdo entre los clientes y los desarrolladores sobre qué debe y qué no debe hacer el sistema. Los requisitos pueden dividirse en dos grupos: requisitos funcionales y requisitos no funcionales.

### 2.2.1 Requisitos Funcionales

Los requisitos funcionales definen las funciones, capacidades o condiciones que el sistema debe ser capaz de realizar, no son más que una descripción de las necesidades de un producto. Constituyen características expresadas que describen sus funcionalidades. Básicamente establecen los comportamientos del sistema. (22) A continuación se describen los requisitos funcionales identificados para el módulo a implementar, los cuales contienen un número de serie único, un nombre y un resumen:

RF1: Buscar especies en las entradas de los ficheros de tipo OrthoXML.

RF2: Mostrar listado de especie encontradas.

RF3: Buscar las entradas de ficheros de tipo OrthoXML.

RF4: Mostrar listado con las entradas de ficheros de tipo OrthoXML encontradas.

RF5: Buscar las entradas de ficheros de tipo SeqXML.

RF6: Mostrar listado con las entradas de ficheros de tipo SeqXML encontradas.

RF7: Buscar especies con genes ortólogos con la especie especificada.

RF8: Mostrar listado de especies con genes ortólogos encontradas.

RF9: Crear el fichero en formato ARFF.

RF10: Crear el fichero en formato SPSS.

RF11: Crea el fichero en formato RoundUP.

RF12: Crear el fichero en formato FASTA.

### 2.2.2 Requisitos no funcionales

Los requisitos no funcionales surgen con las necesidades del usuario y definen las restricciones y propiedades del sistema. Definen las propiedades o cualidades que el producto debe tener. De manera general estos tipos de requisitos se encuentran estrechamente vinculados a los requerimientos funcionales, atendiendo a que una vez que se conoce lo que el sistema debe hacer, es preciso determinar las cualidades que debe cumplir y cuán rápido debe ser. (23) Los requerimientos no funcionales que debe cumplir el sistema para garantizar su correcto funcionamiento son los siguientes:

**Hardware.**

Servidores (Servicios y Portlet):

- Memoria RAM al menos de 1GB.
- Procesadores Pentium 4 o superior.
- Espacio libre en el disco duro no menos de 1GB.

**Software.**

Servidores (Servicios y Portlet):

- Máquina virtual de Java v1.6 o superior.
- Apache Tomcat v6.0.

Cliente del Portlet:

- Navegador Web estándar con capacidad de interpretación de Java Script. Se recomienda Mozilla Firefox.

**Usabilidad.**

- El sistema podrá ser usado por aquellos usuarios que posean conocimientos básicos en el campo de la bioinformática y en el uso de aplicaciones web.

**Confidencialidad.**

- La información de las entradas de los ficheros de tipo SeqXML y OrthoXML, sólo será consultada por los usuarios que ingresen en la PSBio, los cuales son personas autorizadas para el trabajo con la misma.

### 2.3 Casos de uso del sistema (CUS)

Los casos de uso constituyen fragmentos de funcionalidades que el sistema ofrece para aportar un resultado de valor a sus actores. De manera un poco más precisa, un caso de uso especifica una secuencia de acciones que el sistema puede llevar a cabo interactuando con sus actores, incluyendo alternativas dentro de la secuencia. Cada forma en que estos usan el sistema se representa con un caso de uso. (23) A continuación se mencionan los casos de uso en que fueron agrupados los requisitos funcionales identificados para el desarrollo de la aplicación:

- **CUS1:** Listar especies y entradas de ficheros.

- **CUS2:** Crear fichero ARFF o SPSS.
- **CUS3:** Crear fichero FASTA o RoundUP.

### 2.4 Diagrama de casos de uso del sistema

Los diagramas de casos de uso del sistema (CUS) son ubicados dentro de los diagramas de comportamiento de UML y constituyen una representación gráfica de los procesos y su interacción con los actores, se utilizan para ilustrar los requisitos del sistema al mostrar cómo reacciona una respuesta a eventos que se producen en el mismo.

La Figura 10 muestra el diagrama de casos de uso del sistema propuesto para el desarrollo del sistema. El mismo incluye todos los casos de uso necesarios para satisfacer los requisitos funcionales especificados.

.

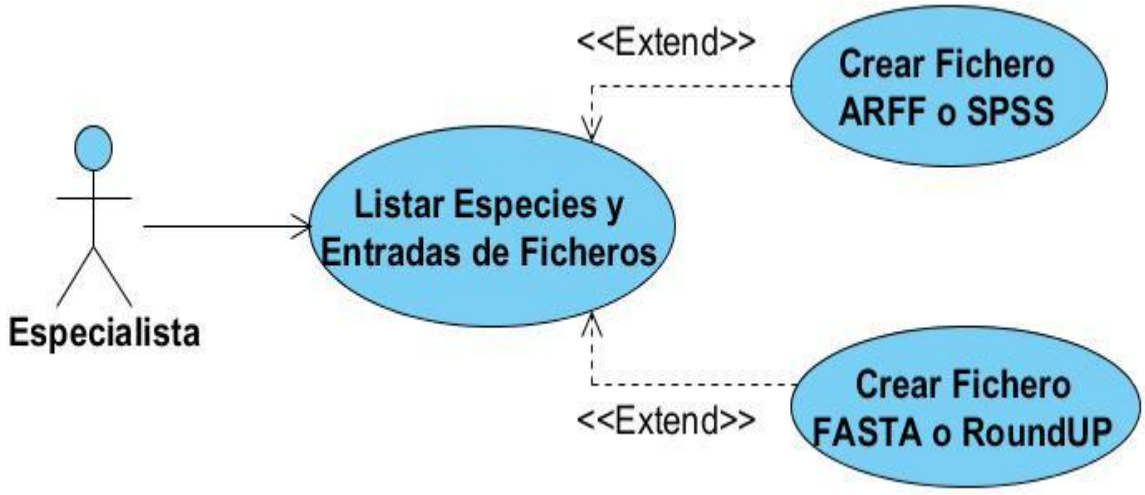



**Figura 10:** Diagrama de casos de uso del sistema

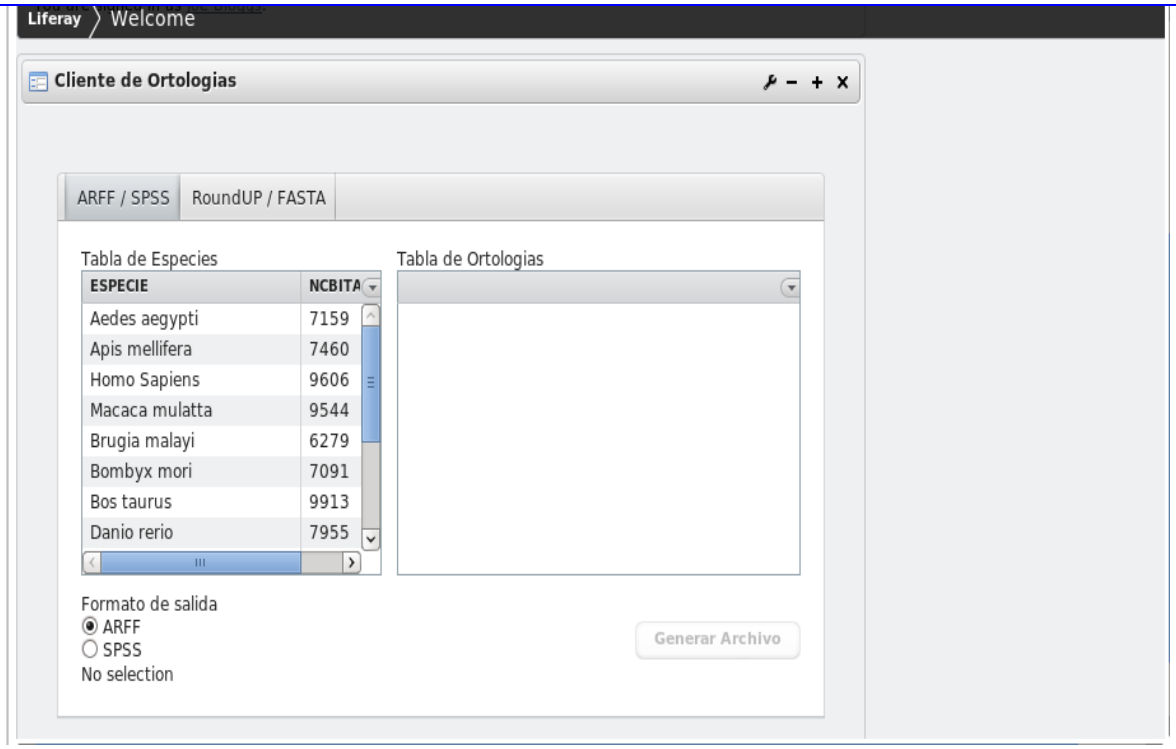
### 2.5 Descripción de los casos de uso del sistema

Una vez identificados los casos de uso del sistema, se realiza una descripción de lo que el sistema debe hacer cuando interactúa con sus actores. En ella se detalla paso a paso el flujo de eventos que ocurren en dicha interacción y se especifican las acciones alternas de cada uno de los escenarios. A continuación se muestra la descripción textual correspondiente a los casos de uso definidos.

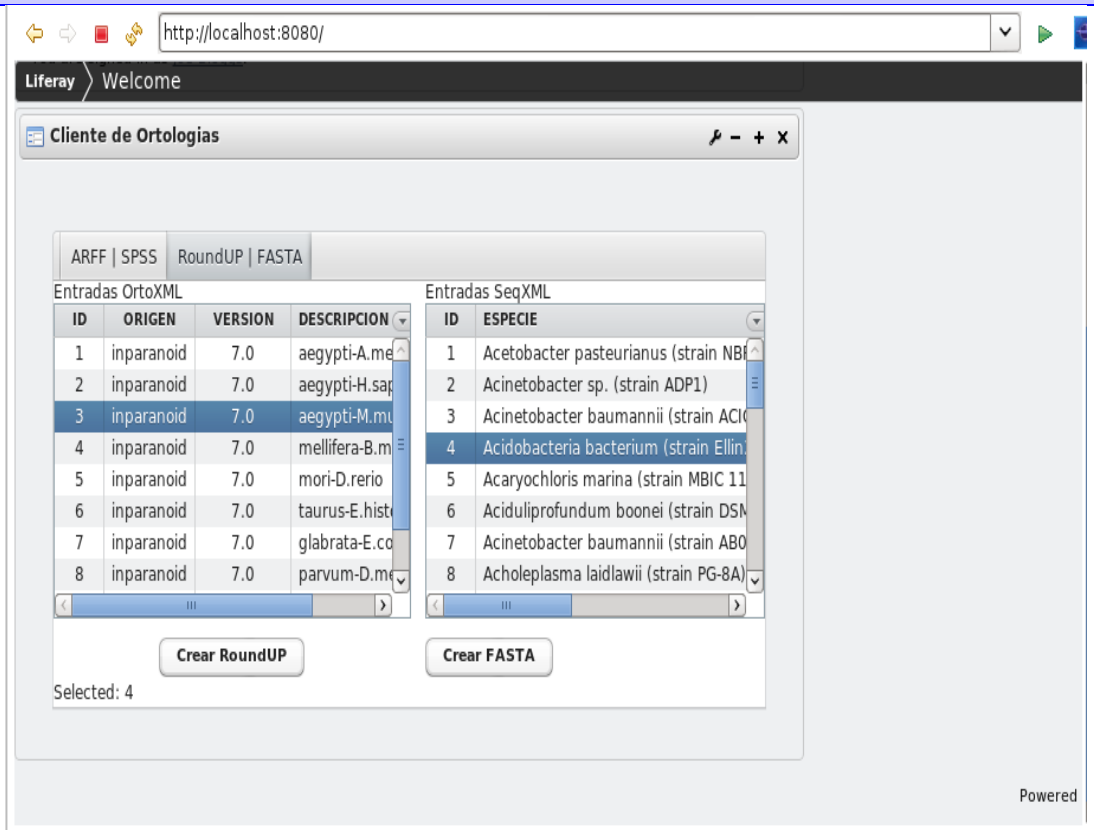
**Tabla 3** Descripción del Caso de Uso "Listar Especies y entradas de ficheros"

| | |
|---|---|
| **Caso de Uso** | Listar especies y entradas de ficheros. |
| **Actores** | Especialista. |
| **Resumen** | El caso de uso inicia cuando el especialista indica trabajar con el sistema de información ortóloga, iniciándose así la búsqueda y posterior visualización de las especies y entradas de ficheros OrthoXML y SeqXML. Termina el caso de uso. |
| **Referencias** | RF1, RF2, RF3, RF4, RF5, RF6. |
| **Complejidad** | Medio. |
| **Prioridad** | Crítico. |
| **Precondiciones** | Que exista conexión con la Base de Datos BD_OrthoXML.<br>Que exista conexión con la Base de Datos BD_SeqXML. |
| **Poscondiciones** | Se visualiza la interfaz con el listado de especies.<br>Se visualiza la interfaz con los listados de entradas de ficheros OrthoXML y SeqXML. |
| **Casos de Uso asociados** | Crear fichero RoundUP o FASTA (Extendido).<br>Crear fichero ARFF o SPSS (Extendido). |
| **Flujo normal de eventos** | |
| **No.** | **Actor** / **Sistema** |
| 1 | Selecciona la opción para trabajar con el sistema de información ortóloga. |
| 2 | Busca las especies en las entradas de los ficheros de tipo OrthoXML. |

| | | |
|---|---|---|
| 3 | | Busca las entradas de ficheros de tipo OrthoXML y SeqXML existentes. |
| 4 | | Muestra una interfaz que contiene un listado con las especies existentes y permite seleccionar una.<br>Campos del listado:<br>➢ name: Nombre de la especie.<br>➢ ncbitaxid: Identificador de la especie. |
| 5 | Selecciona una especie del listado. | |
| 6 | | Ejecuta el caso de uso “Crear fichero ARFF o SPSS”. |
| 7 | | Termina el caso uso. |
| **Prototipo de Interfaz (Flujo normal de eventos)** | | |




| Flujos alternos 1 | | |
|---|---|---|
| 5.1 | Selecciona la interfaz para la creación de ficheros RoundUP/FASTA. | |
| 5.2 | | Muestra una interfaz que contiene un listado con las entradas OrthoXML y otro con las SeqXML encontradas permitiendo seleccionar una en cada listado.<br>Campos del listado de entradas OrthoXML:<br>id_orthoxml: Identificador de la entrada del fichero de tipo OrthoXML. |

| | | |
|---|---|---|
| | | Campos del listado de entradas SeqXML:<br>id_seqxml: Identificador de la entrada del fichero de tipo SeqXML. |
| 5.3 | | Ejecuta el caso de uso “Crear fichero RoundUP o FASTA”. |
| 8 | | Termina el caso de uso. |
| **Prototipo de Interfaz (Flujos Alternos 1)** | | |
|  | | |
| **Flujos alternos 2** | | |
| 5.1.1 | Selecciona la interfaz para la creación de ficheros ARFF/SPSS. | |
| 5.1.2 | | Ejecuta la acción 4 del flujo normal de |

| | | |
|---|---|---|
| | | eventos. |

**Tabla 4** Descripción del Caso de Uso "Crear Fichero ARFF o SPSS"

| **Caso de Uso** | Crear Fichero ARFF o SPSS. | |
|---|---|---|
| **Actores** | Especialista. | |
| **Resumen** | El caso de uso inicia cuando el sistema busca las especies con genes ortólogos con la seleccionada previamente. A partir de la selección de las especies con genes ortólogos, crea un fichero de tipo ARFF o SPSS y muestra la ubicación del fichero. Termina el caso de uso. | |
| **Referencias** | RF7, RF8, RF9, RF10. | |
| **Complejidad** | Alta. | |
| **Prioridad** | Crítico. | |
| **Precondiciones** | Que exista conexión con la base de datos BD_OrthoXML. | |
| **Poscondiciones** | Se crea el fichero con el formato ARFF.<br>Se crea el fichero con el formato SPSS. | |
| **Flujo normal de eventos** | | |
| **No.** | **Actor** | **Sistema** |
| 1 | | Busca las especies que contengan genes ortólogos con la especie seleccionada. |
| 2 | | Muestra un listado de las especies encontradas y permite seleccionar una.<br>Campos del listado:<br>➢ name: Nombre de la especie.<br>➢ ncbitaxid: Identificador de la especie. |
| 3 | Selecciona la especie y el formato para crear el fichero (ARFF o SPSS). | |

| | | |
|---|---|---|
| 4 | | Se activa la opción “Generar archivo”. |
| 5 | | Verifica si el formato que escogió el Especialista fue ARFF. |
| 6 | | Crea el fichero en el formato ARFF. |
| 7 | | Abre el diálogo de descargar ficheros del navegador para que el especialista pueda guardarlo. |
| 8 | | Termina el caso de uso. |
| **Flujos alternos** | | |
| 3.1 | No selecciona una especie con genes ortólogos. | |
| 4.1 | | Permanece desactivada la opción “Generar archivo”. |
| 3.1 | Selecciona el formato SPSS para crear el fichero. | |
| | | Crea el fichero en el formato SPSS. Se ejecuta la acción 7. |

**Tabla 5** Descripción del Caso de Uso “Crear Fichero RoundUP o FASTA”

| | |
|---|---|
| **Caso de Uso** | Crear Fichero RoundUP o FASTA. |
| **Actores** | Especialista. |
| **Resumen** | El caso de uso inicia cuando el especialista selecciona una entrada de fichero OrthoXML o SeqXML. A partir de la selección de una entrada de ficheros de tipo OrthoXML o SeqXML se crea un fichero en formato RoundUP o FASTA. Termina el caso de uso. |
| **Referencias** | RF11, RF12. |

| | |
|---|---|
| **Complejidad** | Alta. |
| **Prioridad** | Crítico. |
| **Precondiciones** | Que exista conexión con la base de datos BD_OrthoXML.<br>Que exista conexión con la base de datos BD_SeqXML. |
| **Poscondiciones** | Se crea el fichero en formato RoundUP.<br>Se crea el fichero en formato FASTA. |

| **Flujo normal de eventos** | | |
|---|---|---|
| **No.** | **Actor** | **Sistema** |
| 1 | Selecciona una entrada de fichero de tipo OrthoXML. | |
| 2 | | Permite crear archivo con formato RoundUP. |
| 3 | Indica crear el fichero a formato RoundUP. | |
| 4 | | Crea el fichero en el formato RoundUP. |
| 5 | | Abre el diálogo de descargar ficheros del navegador para que el especialista pueda guardarlo. |
| 6 | | Termina el caso de uso. |
| **Flujos alternos** | | |
| 1.1 | Selecciona una entrada de fichero de tipo SeqXML. | |
| 1.2 | | Permite crear archivo con formato FASTA. |
| 1.3 | Indica crear el fichero a formato FASTA. | |
| 1.4 | | Crea el fichero en el formato FASTA. |
| 1.5 | | Abre el diálogo de descargar ficheros del navegador para que el especialista |

| | | |
|---|---|---|
| | | pueda guardarlo. |
| 1.6 | | Termina el caso de uso. |
| **Flujos alternos 1** | | |
| 1.1.1 | No selecciona una entrada de fichero. Termina el caso de uso. | |

### 2.6 Conclusiones

Después de realizar un estudio de los conceptos fundamentales relacionados con el contexto del sistema se elaboró el modelo de dominio el cual permitió un mejor entendimiento del problema a resolver. Se identificaron 12 requisitos funcionales agrupados en 3 casos de uso los que fueron descritos textualmente para obtener una visión más detalla del flujo de actividades a implementar en la aplicación. Se confeccionó el diagrama de CUS el cual permitió representar las interacciones entre los actores y casos de uso del sistema a implementar.

## Capítulo 3: Diseño

Este capítulo está dedicado a la descripción de los patrones de arquitectura y diseño usados en el desarrollo de la herramienta. Además se identifican las clases que implementan las funcionalidades identificadas. Se realizan los diagramas de clase del diseño y de secuencia respectivamente.

### 3.1 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 y tecnologías de interfaz de usuario. Además crea un punto de partida para las actividades de implementación. Es el punto de mayor importancia al final de la fase de elaboración y el comienzo de las iteraciones de construcción. Permite facilitar una arquitectura estable y sólida, además de crear un plano muy cercano del modelo de implementación. Es necesario mantener el modelo de diseño a través de todo el ciclo de vida del software.

### 3.2 Diagrama de clases del diseño

Los diagramas de clases son utilizados durante el proceso de análisis y diseño de los sistemas, donde se crea el diseño conceptual de la información que se manejará en el sistema y los componentes que se encargarán del funcionamiento y la relación entre uno y otro.

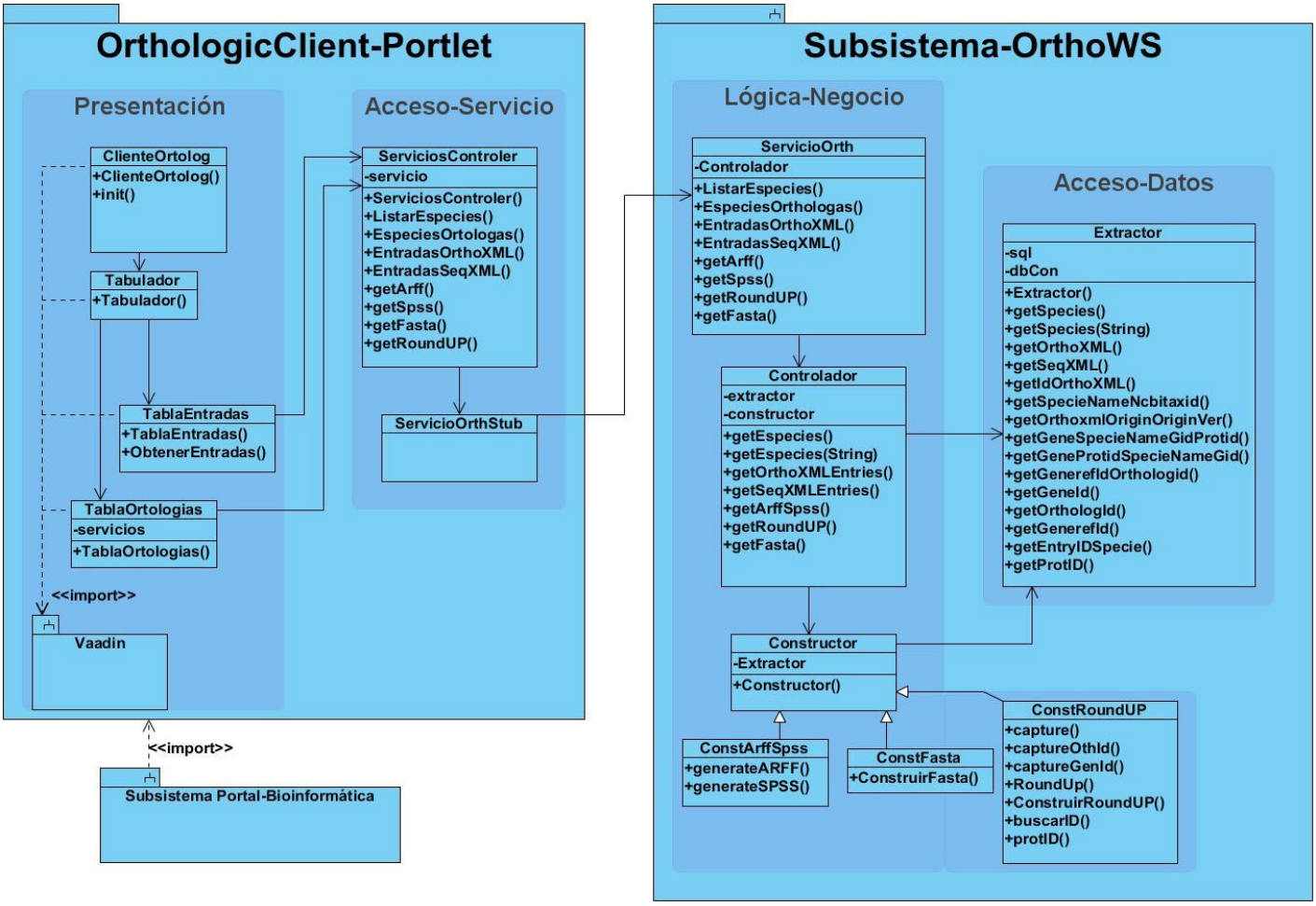



**Figura 11:** Diagrama de Clases del Diseño

Para una mejor comprensión del diseño del sistema se dividió este en dos componentes como muestra la Figura 11. El portlet en un paquete y en otro un subsistema que agrupa los servicios web, ambos se modelaron utilizando el patrón arquitectónico “Capas”. El paquete OrthologicClient-Portlet agrupa las clases que generan la vista dentro de la capa Presentación y en la capa Acceso-Servicio se agrupan las clases clientes de los servicios web. El Subsistema-OrthoWS incluye los servicios web y agrupa las clases que conforman la lógica del negocio en la capa Lógica-Negocio y la clase de acceso a datos en Acceso-Datos.

### 3.3 Patrones de diseño

**Patrón experto**: El problema que resuelve el patrón experto está referido al principio más básico mediante el cual las responsabilidades son asignadas en el diseño orientado a objetos.

En el sistema este patrón se evidencia en la clase constructor, la misma cuenta con la información necesaria para crear y modificar. Ver en la Figura 12.

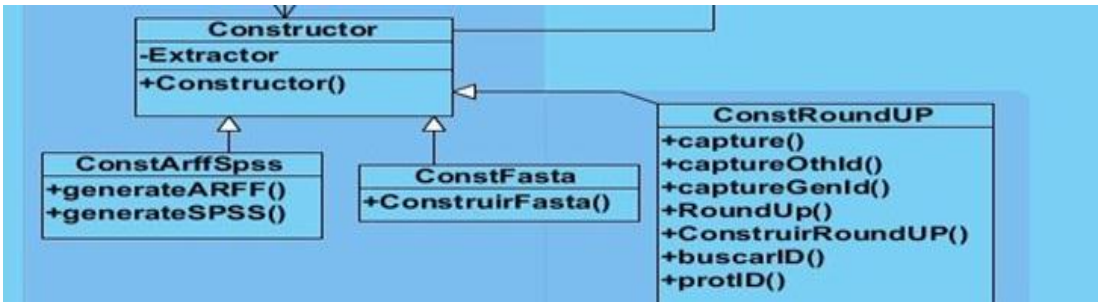



**Figura 12:** Aplicación del patrón Experto

**Patrón Creador:** La clase Controlador tiene la responsabilidad de crear y contener instancias de las clases Extractor y Constructor ya que posee la información necesaria o datos de inicialización de ambas. Ver Figura 13.


**Figura 13:** Aplicación del patrón Creador

**Patrón Bajo Acoplamiento:** Las clases que implementan la lógica de negocio y de acceso a datos se encuentran en el Modelo, estas clases no tienen asociaciones con las de la Vista o el Controlador por lo que la dependencia en este caso es baja.

El Bajo Acoplamiento soporta el diseño de clases más independientes que reducen el impacto de los cambios, y también más reutilizables, acrecentando la oportunidad de una mayor productividad. Este patrón es un principio que asigna la responsabilidad de controlar el flujo de eventos del sistema a clases específicas. Esto facilita la centralización de actividades. El controlador no realiza estas actividades, las delega en otras clases con las que mantiene un modelo de Alta Cohesión.

**Patrón Alta Cohesión:** Asignar a las clases responsabilidades que trabajen sobre una misma área de la aplicación y que no tengan mucha complejidad, evitando así que una clase sea la única responsable de muchas tareas.

La cohesión es una medida de la fuerza con la que se relacionan las clases y el grado de focalización de las responsabilidades de un elemento. Cada elemento de diseño debe realizar una labor única dentro del sistema, no desempeñada por el resto de los elementos y auto-identificable. Una clase con baja cohesión hace muchas cosas no relacionadas o hace demasiado trabajo, radicando en estos inconvenientes la importancia de la correcta utilización de este patrón.

### 3.4 Diagramas de Secuencia

Los diagramas de secuencia ofrecen las interacciones entre clases. Describen el curso particular de los eventos de un caso de uso, proporcionando una realización física de comportamiento de los casos de uso en términos de clases del diseño. En las próximas Figuras se muestran los diagramas de secuencias de los escenarios que conforman el CUS “Crear fichero en formato FASTA”. El resto de los CUS pueden verse en los anexos.

**CUS** “Crear fichero FASTA o RoundUP”:

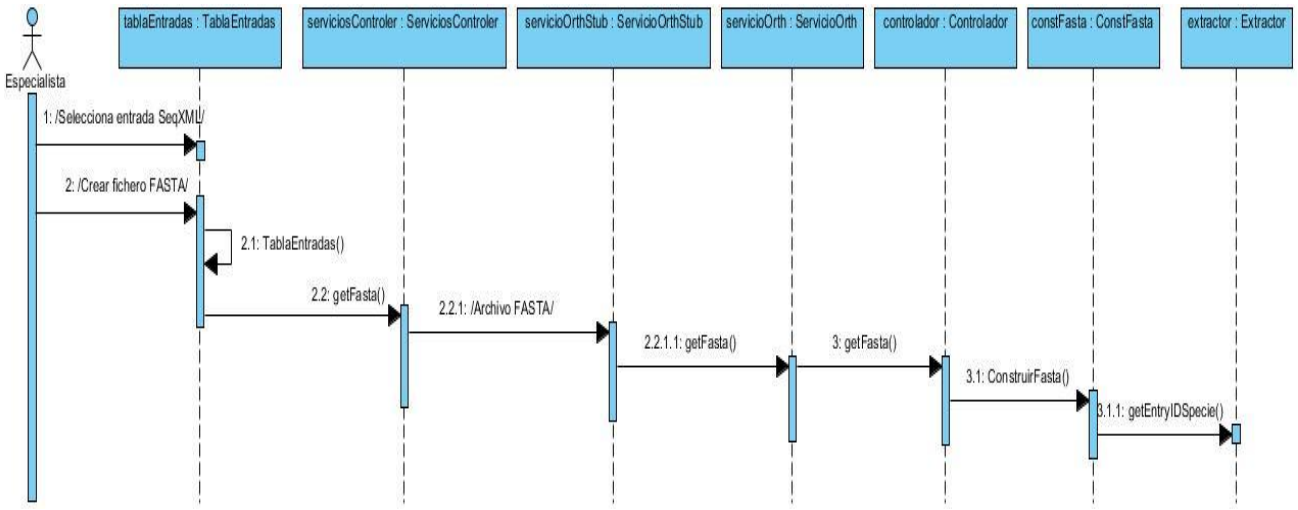



**Figura 14:** Diagrama de secuencia del escenario “Crear fichero en formato FASTA”

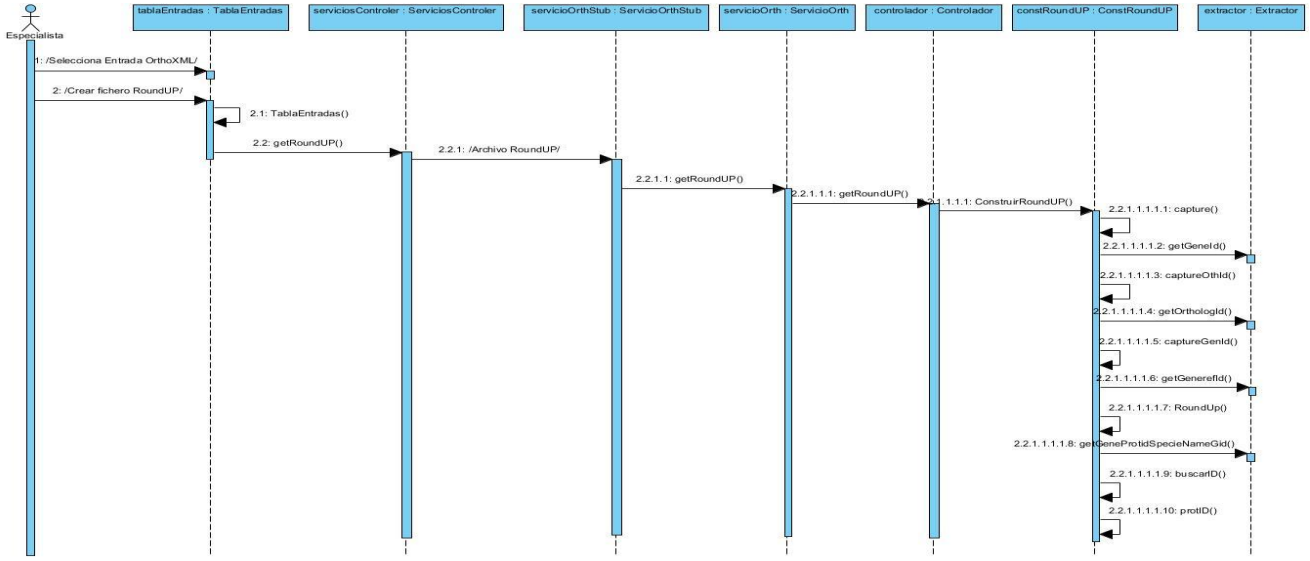



**Figura 15:** Diagrama de secuencia del escenario “Crear fichero en formato RoundUP”

### 3.5 Conclusiones

Una vez concluido el diseño de la solución se obtuvo como resultado el diagrama de clases del diseño el cual permitió comprender detalladamente las características del sistema. La elaboración del diagrama de secuencia permitió establecer un punto de partida para la posterior implementación del módulo propuesto.

Se seleccionó como patrón arquitectónico el patrón por Capas, el cual permitió separar la implementación del software en varias capas, posibilitando que los cambios realizados en una de ellas no tengan una gran repercusión en el resto.

**Capítulo 4: Implementación y pruebas**

En este capítulo se expondrá el diagrama de componentes, donde se describen los elementos físicos del sistema y sus relaciones. Además se representan los nodos necesarios para realizar el despliegue del sistema. Por último se describen detalladamente las pruebas que se le realizan al sistema, en busca de cualquier error que no se haya detectado durante la implementación.

## 4.1 Modelo de Implementación

El modelo de implementación describe cómo los elementos del modelo de diseño se implementan en términos de componentes, ficheros de código fuente y ejecutables. Describe también cómo se organizan los componentes de acuerdo a los mecanismos de estructuración disponibles en el entorno de implementación, lenguajes de implementación empleados y cómo dependen los componentes unos de otros.

## 4.2 Diagrama de Componentes

Un diagrama de componentes muestra las organizaciones y dependencias lógicas entre componentes de software: código fuente, binarios o ejecutables. Desde el punto de vista del diagrama de componentes se tienen en consideración los requisitos relacionados con la facilidad de desarrollo, la gestión del software, la reutilización, y las limitaciones imputadas por los lenguajes de programación y las herramientas utilizadas en el desarrollo. (24)

Un componente es una parte física y reemplazable de un sistema conformado por un conjunto de interfaces y proporciona la realización de dicho conjunto. Se usan para modelar los elementos físicos que pueden hallarse en un nodo, por lo que empaquetan elementos como clases, colaboraciones e interfaces. En la Figura 16 se muestra el diagrama de componentes del sistema. (24)

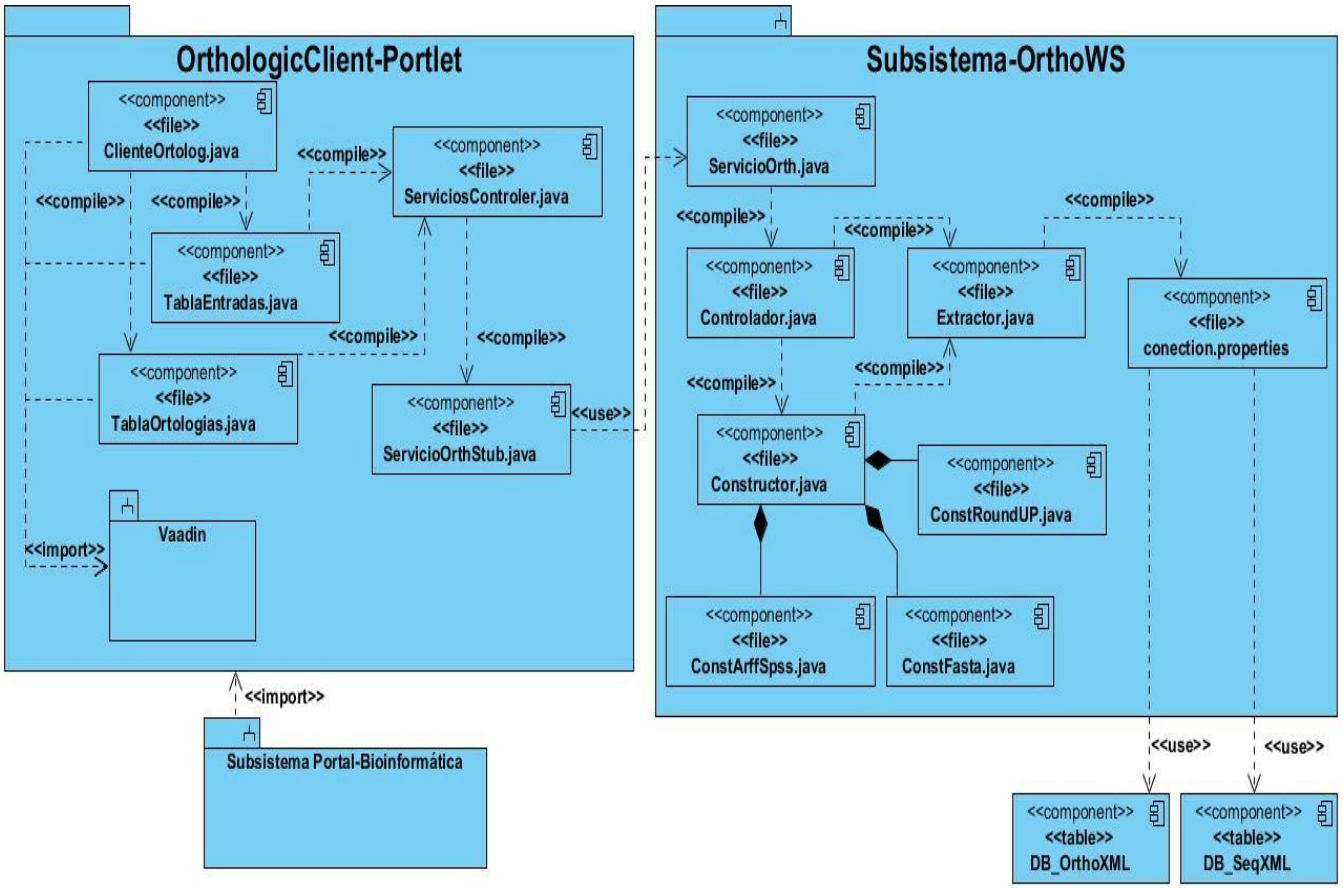



**Figura 16:** Diagrama de componentes del sistema

### 4.3 Diagrama de despliegue

El modelo de despliegue describe la arquitectura física del sistema durante su ejecución, en términos de procesadores, dispositivos y componentes de software. Describe además, la topología del sistema, es decir, la estructura de los elementos de hardware y software que ejecuta cada uno de ellos.

En la Figura 17 se muestra el modelo de despliegue de la aplicación desarrollada para la cual es necesario un servidor de portlet para incluir los componentes del portlet, el mismo debe conectarse al servidor de servicios a través del protocolo de red SOAP para consumir los servicios que este brinda.

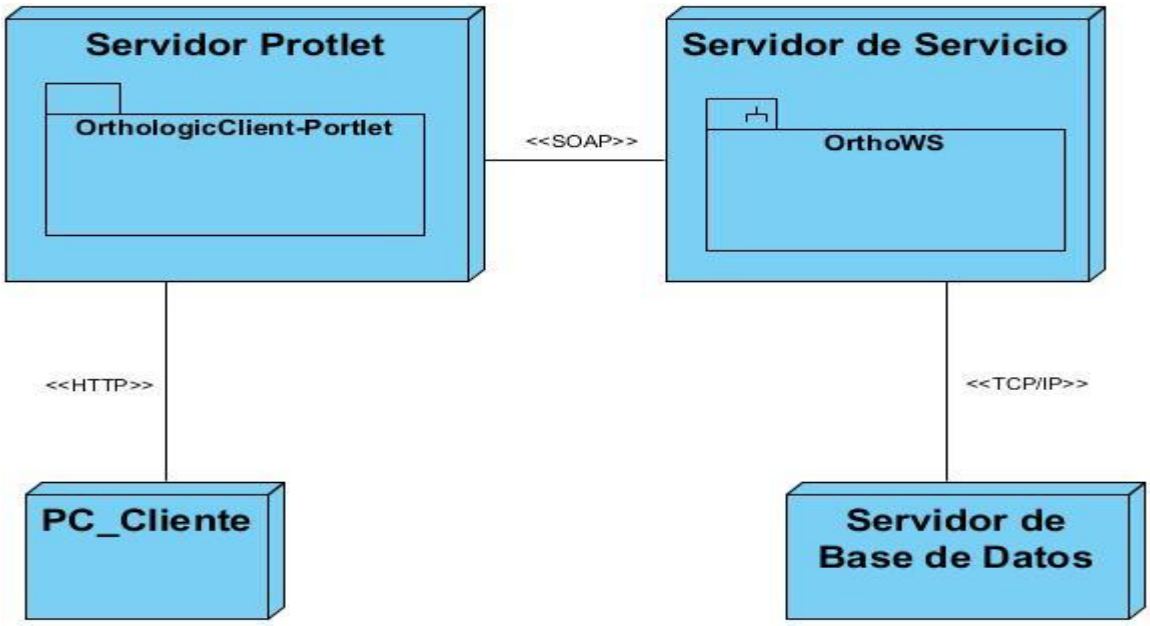



**Figura 17:** Diagrama de despliegue de la aplicación

### 4.4 Pruebas de Software

Cuando un sistema tiene terminado una serie de clases y componentes, el software debe ser probado para descubrir y corregir el máximo de errores posible antes de su entrega al cliente. El objetivo es diseñar una serie de casos de prueba que tengan una alta probabilidad de encontrar errores. Para ello se aplican una serie de técnicas de pruebas del software. Estas técnicas facilitan una guía sistemática para diseñar pruebas que comprueben la lógica interna de los componentes del software y verifiquen los dominios de entrada y salida del programa para descubrir errores en la funcionalidad, el comportamiento y en el rendimiento. (25)

Las pruebas definen el grado de aceptación del sistema que pueden tener los clientes del mismo, e incluso determinar si lo aceptan o no. Para ser más eficaces, las pruebas deberían ser realizadas por un equipo independiente, esto garantiza pruebas con alta probabilidad de encontrar errores, que es el objetivo principal de las pruebas. (25)

#### 4.4.1 Diseño de prueba de caja negra

Las pruebas de caja negra permiten identificar las posibles fallas en el funcionamiento del sistema. Estas

pruebas se centran principalmente en las características del sistema independientemente del código. Para llevarlas a cabo es necesario tener conocimiento sobre los escenarios de prueba y secciones que serán probadas.

En los casos de pruebas se combinan los posibles juegos de datos válidos y no válidos que son necesarios para verificar el correcto funcionamiento del sistema. Estas pruebas se realizan a nivel de sistema, el tipo de prueba es funcional, donde se emplea el método de caja negra, específicamente haciendo uso de la técnica de pruebas de partición equivalentes. Esta técnica es muy efectiva al realizar pruebas sobre la interfaz de la plataforma, pues prueban la validez de cada entrada de datos o información al sistema. Pretenden demostrar que las funciones de la plataforma son operativas. (26)

**Caso de prueba diseñado para el CUS Listar especies y entradas de ficheros**

| Nombre la sección | Escenario | Descripción de la funcionalidad | Flujo central |
|---|---|---|---|
| SC 1 Listar especies y entradas de ficheros. | EC 1.1: Mostrar listado de especies encontradas satisfactoriamente. | El sistema muestra una interfaz que contiene un listado con las especies existentes. Los campos que se muestran son:<br>➢ name: nombre de la especie.<br>➢ ncbitaxid: identificador de la especie. | Plataforma de Servicios Bioinformáticos / Portal de Portlets / Portlet para la Gestión de Genes Ortólogos |
| | EC 1.2: Mostrar listado con las entradas de ficheros de tipo OrthoXML y SeqXML encontradas. | El sistema muestra una interfaz que contiene un listado con las entradas OrthoXML y otro con las SeqXML encontradas.<br>Campos del listado de entradas | Plataforma de Servicios Bioinformáticos / Portal de |

| | | | |
|---|---|---|---|
| | | OrthoXML:<br>➢ id_orthoxml: Identificador de la entrada del fichero de tipo OrthoXML.<br>Campos del listado de entradas SeqXML:<br>➢ id_seqxml: Identificador de la entrada del fichero de tipo SeqXML. | Portlets / Portlet para la Gestión de Genes Ortólogos / Clic en Tab "RoundUP/ FASTA" |

**Caso de prueba diseñado para el CUS Crear fichero ARFF o SPSS.**

| Nombre la sección | Escenario | Descripción de la funcionalidad | Flujo central |
|---|---|---|---|
| SC 2: Crear fichero ARFF o SPSS. | EC 2.1: Mostrar listado de especies con genes ortólogos encontradas. Satisfactoriamente. | El especialista selecciona una especie del listado "Tabla de Especies". El sistema muestra un listado de las especies con genes ortólogos con la seleccionada previamente y permite seleccionar una.<br>Campos del listado:<br>➢ name: Nombre de la especie.<br>➢ ncbitaxid: Identificador de la especie. | Plataforma de Servicios Bioinformáticos / Portal de Portlets / Portlet para la Gestión de Genes Ortólogos |
| | EC 2.2: Seleccionar especie. Satisfactoriamente. | El especialista selecciona la especie del listado "Tabla de Ortologías". El sistema activa la opción "Generar Archivo". | Plataforma de Servicios Bioinformát |

| | | | |
|---|---|---|---|
| | | | icos / Portal de Portlets / Portlet para la Gestión de Genes Ortólogos |
| | EC 2.3: Seleccionar especie. Satisfactoriamente. | El especialista selecciona una especie del listado "Tabla de Ortologías" e intenta realizar una selección múltiple. El sistema mantiene seleccionada una sola especie, no permite la selección múltiple. Queda activada la ultima especie seleccionada y la opción "Generar Archivo". | Plataforma de Servicios Bioinformáticos / Portal de Portlets / Portlet para la Gestión de Genes Ortólogos |
| | EC 2.4: Seleccionar especie. Falla. | El especialista no selecciona ninguna especie del listado "Tabla de Ortologías". El sistema mantiene desactivada la opción "Generar Archivo". | Plataforma de Servicios Bioinformáticos / Portal de Portlets / Portlet para la Gestión de Genes Ortólogos |
| | EC 2.5: Crear fichero ARFF o SPSS. | El especialista selecciona el formato para crear el fichero (ARFF o SPSS) y la opción "Generar Archivo". El | Plataforma de Servicios Bioinformát |

| | | | |
|---|---|---|---|
| | | sistema muestra la URL donde se encuentra el archivo generado. | icos / Portal de Portlets / Portlet para la Gestión de Genes Ortólogos |

**Caso de prueba diseñado para el CUS Crear fichero FATA o RoundUP.**

| Nombre la sección | Escenario | Descripción de la funcionalidad | Flujo central |
|---|---|---|---|
| SC 3: Crear Fichero FASTA o RoundUP. | EC 3.1: Seleccionar entrada OrthoXML. Satisfactoriamente. | El especialista selecciona la especie del listado “Entradas OrthoXML”. El sistema activa la opción “Crear RoundUP”. | Plataforma de Servicios Bioinformáticos / Portal de Portlets / Portlet para la Gestión de Genes Ortólogos |
| | EC 3.2: Seleccionar entrada OrthoXML. Satisfactoriamente. | El especialista selecciona una especie del listado “Entradas OrthoXML” e intenta realizar una selección múltiple. El sistema mantiene seleccionada una sola especie, no permite la selección múltiple. Queda activada la ultima especie seleccionada y la opción | Plataforma de Servicios Bioinformáticos / Portal de Portlets / Portlet para la Gestión de Genes |

| | | | |
|---|---|---|---|
| | | "Crear RoundUP". | Ortólogos |
| | EC 3.3: Seleccionar entrada OrthoXML. Falla. | El especialista no selecciona ninguna especie del listado "Entradas OrthoXML". El sistema mantiene desactivada la opción "Crear RoundUP". | Plataforma de Servicios Bioinformáticos / Portal de Portlets / Portlet para la Gestión de Genes Ortólogos |
| | EC 3.4: Crear fichero RoundUP. | El especialista selecciona la opción "Crear RoundUP". El sistema muestra la URL donde se encuentra el archivo generado. | Plataforma de Servicios Bioinformáticos / Portal de Portlets / Portlet para la Gestión de Genes Ortólogos |
| | EC 3.5: Seleccionar entrada SeqXML. Satisfactoriamente. | El especialista selecciona la especie del listado "Entradas SeqXML". El sistema activa la opción "Crear FASTA". | Plataforma de Servicios Bioinformáticos / Portal de Portlets / Portlet para la Gestión de Genes |

| | | | |
|---|---|---|---|
| | | | Ortólogos |
| | EC 3.6: Seleccionar entrada SeqXML. Satisfactoriamente. | El especialista selecciona una especie del listado “Entradas SeqXML” e intenta realizar una selección múltiple. El sistema mantiene seleccionada una sola especie, no permite la selección múltiple. Queda activada la ultima especie seleccionada y la opción “Crear FASTA”. | Plataforma de Servicios Bioinformáticos / Portal de Portlets / Portlet para la Gestión de Genes Ortólogos |
| | EC 3.7: Seleccionar entrada SeqXML. Falla. | El especialista no selecciona ninguna especie del listado “Entradas SeqXML”. El sistema mantiene desactivada la opción “Crear FASTA”. | Plataforma de Servicios Bioinformáticos / Portal de Portlets / Portlet para la Gestión de Genes Ortólogos |
| | EC 3.8: Crear fichero RoundUP. | El especialista selecciona la opción “Crear FASTA”. El sistema muestra la URL donde se encuentra el archivo generado. | Plataforma de Servicios Bioinformáticos / Portal de Portlets / Portlet para la Gestión de Genes |

| | | | |
|---|---|---|---|
| | | | Ortólogos |

### 4.4.2 Resultados de las pruebas

La Figura 18 muestra que en la primera iteración se detectaron 6 no conformidades de las cuales fueron resueltas las 6, en la segunda iteración se detectaron 2 no conformidades y se resolvieron las 2 y en la cuarta iteración no se encontraron no conformidades. Cada una de las no conformidades detectas fueron resueltas inmediatamente por el equipo de desarrollo. La solución de las no conformidades contribuyó a que la aplicación alcanzara una mejor calidad y cumpliera con los requisitos planteados para satisfacer al usuario.

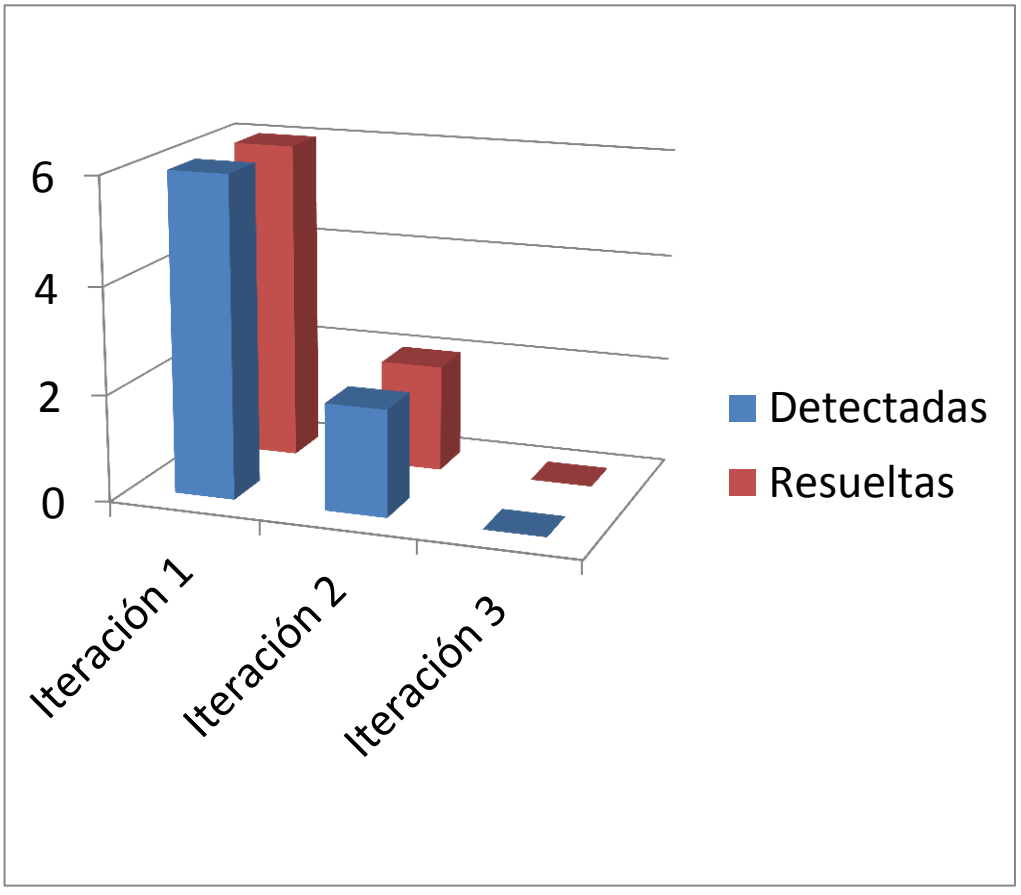



**Figura 18:** Resultados de las pruebas por iteración

### 4.5 Conclusiones del capítulo

En el desarrollo del capítulo concluido se elaboró el diagrama de componentes que permitió organizar y definir las dependencias entre ejecutables, códigos fuentes y librerías que se utilizarán en el sistema. Se define el estándar de programación, el cual nos permitió lograr una mejor organización del código fuente. Por su parte el diagrama de despliegue ofrece una representación visual de la disposición física de los nodos que componen el sistema, incluyendo el protocolo de comunicación. Por último se describieron las principales pruebas realizadas al sistema, indicando para cada una la respuesta de la aplicación. De forma general cada una de las pruebas arrojaron resultados satisfactorios, pues se obtuvo un número reducido de no conformidades, las cuales fueron solucionadas de forma rápida por el equipo de desarrollo, contribuyendo a obtener un producto de mayor calidad.

**Conclusiones generales**

Una vez terminada la investigación se arribó a las siguientes conclusiones:

- Se elaboró el marco teórico de la investigación definiéndose como metodología de desarrollo de software OpenUP, la cual permitió guiar el proceso de construcción del sistema propuesto. Las tecnologías y herramientas seleccionadas dieron soporte a la implementación de la aplicación.

- Se implementó un componente portlet que consume servicios web para el manejo de datos ortológicos en varios formatos el cual podrá ser desplegado en el contenedor de portlet del Portal de Servicios Bioinformáticos.

- Se aplicaron casos de prueba de caja negra de partición de equivalencia con el objetivo de validar las funcionalidades implementadas desde la interfaz del sistema. Se realizaron 4 iteraciones de pruebas, donde se identificaron algunas no conformidades que fueron solucionadas por el equipo de desarrollo.

**Recomendaciones**

Una vez concluido la investigación se recomienda:

- Optimizar los algoritmos que se emplean en la construcción de los archivos a partir de consultas a la base de datos que almacena información de genes ortólogos.
- Implementar un mecanismo de búsqueda o filtro que permita una búsqueda más eficiente por parte del especialista.

# *Referencias bibliográficas*

**Referencias Bibliográficas**


1. BiGRe. [En línea] 2011. [Citado el: 12 de 11 de 2012.] http://www.bigre.ulb.ac.be/.

2. *Integración de la Bioinformática en la investigación Genómica Cardiovascular: Aplicaciones en el Framingham Heart Study. Departamento de Lenguajes y Sistemas Informáticos.* España : Escuela Superior de Tecnología y Ciencias Experimental, 2011.

3. **RIVERA, Fray León Osorio.** *Base de datos relacionales. Itm.* 2010.

4. **Clara Elena Brizuela Figueredo, Claudia García Suárez del Villar, Julio Cesar Brito Rodríguez.** *Mercado de datos para la Empresa de Mantenimiento de Grupos Electrógenos.* La Habana, Cuba : s.n., 2013.

5. **MIÑO VITERI, Pablo Miguel.** *Desarrollo del portal interno de GESTORINC SA mediante la creación extensión e implementación de portlets, utilizando liferay portal como gestor de contenidos.* 2010.

6. **LOJA CAJAS, Jorge Oswaldo.** *Diseño e implementación del Portal Web de la Universidad Politécnica Salesiana.* 2011.

7. **COLTELL, O.** *Servicio Web como soporte de Investigación de las redes de Investigación en Biomédica. La Información de la Salud: Punto de encuentro Disciplinas Sanitarias.* 2003.

8. **VIERA, Ivette Camayd.** *UN ACERCAMIENTO A LA ONTOLOGÍA DE GENES Y SUS APLICACIONES.* 2012.

9. **Dr. Pere Marquès Graells.** INTRODUCCIÓN A LA INFORMÁTICA. [En línea] 2010. [Citado el: 10 de 12 de 2012.] http://www.peremarques.net/INFMULTI.htm.

10. Cavsi. [En línea] 2012. [Citado el: 10 de 12 de 2012.] http://www.cavsi.com/espanol/TerminosComputacionAcronimosAbreviacionesSiglas.html.

11. **Morate, Diego García.** *MANUAL DE WEKA.* España : s.n., 2012.

12. Locualo. [En línea] 2013. [Citado el: 18 de 02 de 2013.] http://www.locualo.net/programacion/mineria-datos-weka-ficheros-arff/00000019.aspx.

13. **SERRANO, Juan Manuel.** *Arquitectura del software.*

# *Referencias bibliográficas*


14. **LARMAN, Craig.** *UML y patrones. Pearson.* 2010.

15. Academia.edu. [En línea] 2013. [Citado el: 18 de 02 de 2013.] http://www.academia.edu/1740166/Metodologias_de_desarrollo_agil.

16. Software.com.ar. [En línea] 2013. [Citado el: 19 de 02 de 2013.] http://www.software.com.ar/visual-paradigm-para-uml.html.

17. Desarrolloweb.com. [En línea] 2012. http://www.desarrolloweb.com/articulos/1692.php.

18. **ALLAMARAJU, Subrahmanyam, BEUST, Cedric y DAVIES, John.** *Programación Java Server con J2EE Edición.* 2002.

19. Woorank. [En línea] 2013. [Citado el: 19 de 02 de 2013.] http://www.woorank.com/es/www/vaadin.com.

20. Apache Axis2. [En línea] 2012. [Citado el: 10 de 12 de 2012.] http://maypun.blogspot.com/2009/09/apache-axis2.html.

21. **JACOBSON, Ivar, BOOCH, Grady y RUMBAUGH, James.** *El proceso unificado de desarrollo de software.* s.l. : Reading: Addison Wesley, 2000.

22. **DÍAZ, Isabel, SÁNCHEZ, Juan y PASTOR, Oscar.** *Metamorfosis: Un marco para el análisis de requisitos funcionales.* Portugal : s.n., 2005.

23. **ESCALONA, María José y KOCH, Nora.** *Ingeniería de Requisitos en Aplicaciones para la Web-Un estudio comparativo.* España : Universidad de Sevilla. 2002.

24. **STEVENS, Perdita y POOLEY, Rob.** *Utilización de UML en Ingeniería del Software con Objetos y Componentes.* s.l. : Addison Wesley, 2007.

25. **USAOLA, Macario Polo.** *Pruebas del Software.*

26. **SANZ, Luis Fernández y BERCIAL, Pedro J. Lara.** *CUADRADO-GALLEGO, Juan J. Mejora de la calidad en desarrollos orientados a objetos utilizando especificaciones UML para la obtención y precedencia de casos de prueba. Revista de Procesos y Métricas de las Tecnologías de la Información (RPM).* 2004.

**Bibliografía**

1BiGRehttp://www.bigre.ulb.ac.be/

2*Integración de la Bioinformática en la investigación Genómica Cardiovascular: Aplicaciones en el Framingham Heart Study. Departamento de Lenguajes y Sistemas Informáticos*EspañaEscuela Superior de Tecnología y Ciencias Experimental2011

3*Base de datos relacionales. Itm.*2010

4*Mercado de datos para la Empresa de Mantenimiento de Grupos Electrógenos.*La Habana, Cuba2013

5 *Desarrollo del portal interno de GESTORINC SA mediante la creación extensión e implementación de portlets, utilizando liferay portal como gestor de contenidos.*2010

6*Diseño e implementación del Portal Web de la Universidad Politécnica Salesiana*2011

7*Servicio Web como soporte de Investigación de las redes de Investigación en Biomédica. La Información de la Salud: Punto de encuentro Disciplinas Sanitarias*2003

8*UN ACERCAMIENTO A LA ONTOLOGÍA DE GENES Y SUS APLICACIONES.*2012

9**Dr. Pere Marquès Graells**INTRODUCCIÓN A LA INFORMÁTICAhttp://www.peremarques.net/INFMULTI.htm

10Cavsihttp://www.cavsi.com/espanol/TerminosComputacionAcronimosAbreviacionesSiglas.html

11*MANUAL DE WEKA*España2012

12Locualohttp://www.locualo.net/programacion/mineria-datos-weka-ficheros-arff/00000019.aspx

13*Arquitectura del software.*

14*UML y patrones. Pearson*2010

15Academia.eduhttp://www.academia.edu/1740166/Metodologias_de_desarrollo_agil

16Software.com.arhttp://www.software.com.ar/visual-paradigm-para-uml.html

17Desarrolloweb.comhttp://www.desarrolloweb.com/articulos/1692.php

# *Bibliografía*


18*Programación Java Server con J2EE Edición*2002

19Woorankhttp://www.woorank.com/es/www/vaadin.com

20Apache Axis2http://maypun.blogspot.com/2009/09/apache-axis2.html

21*El proceso unificado de desarrollo de software*Reading: Addison Wesley2000

22*Metamorfosis: Un marco para el análisis de requisitos funcionales*Portugal2005

23*Ingeniería de Requisitos en Aplicaciones para la Web-Un estudio comparativo.*EspañaUniversidad de Sevilla2002

24*Utilización de UML en Ingeniería del Software con Objetos y Componentes.*Addison Wesley2007

25*Pruebas del Software.*

26*CUADRADO-GALLEGO, Juan J. Mejora de la calidad en desarrollos orientados a objetos utilizando especificaciones UML para la obtención y precedencia de casos de prueba. Revista de Procesos y Métricas de las Tecnologías de la Información (RPM)*2004

*27. Pressman, Roger S.. Ingeniería del software. Un enfoque práctico. Lugar de publicación desconocido. Mc Graw Hill. 1998. 614.*

*28. Eclipse Project Framework. Concept: Use Case. [en línea], 2011. [Consultado el: 19 de marzo de 2013].Disponible en:[http://epf.eclipse.org/wikis/openup/core.tech.common.extend_supp/guidances/concepts/use_case_BB199D1B.html?nodeId=e37dcf94].*

*29. Larman, C.. UML y Patrones. Introducción al análisis y diseño orientado a objetos. México.Prentice Hall. 1999. 499.*

## Anexo 1

**CUS** “Listar Especies y Entradas de Ficheros”

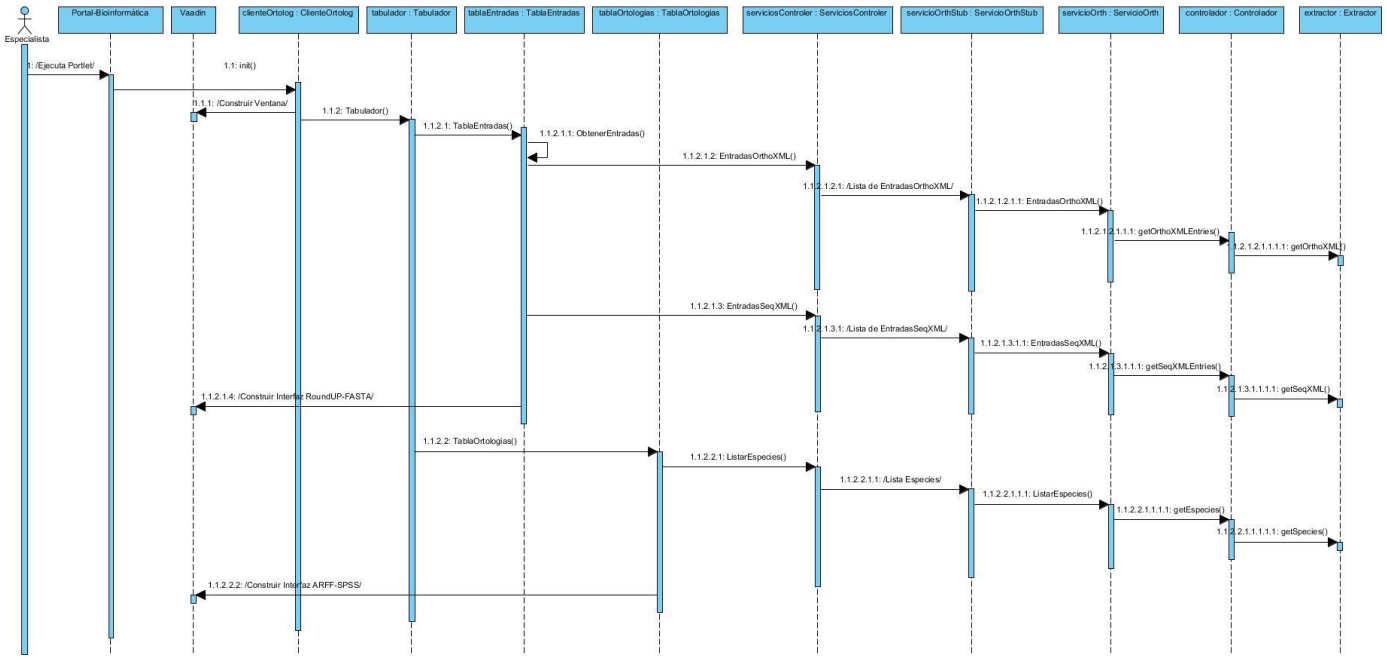



**Figura 19:** Diagrama de secuencia del escenario "Listar Especies y Entradas de Ficheros”

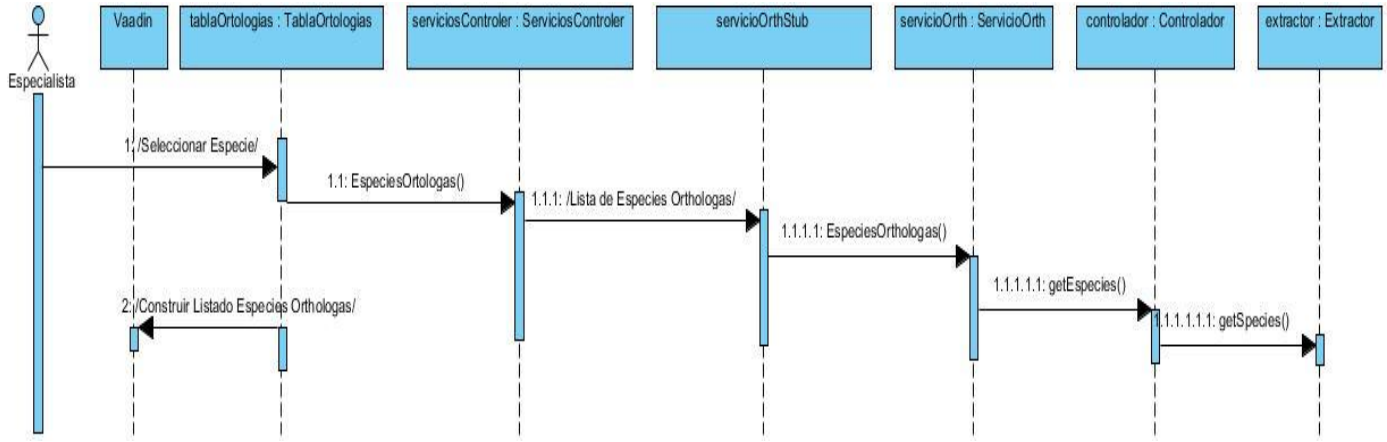



**Figura 20:** Diagrama de secuencia del escenario " Listar Especies Ortólogas”

**CUS** “Crear Fichero ARFF o SPSS”

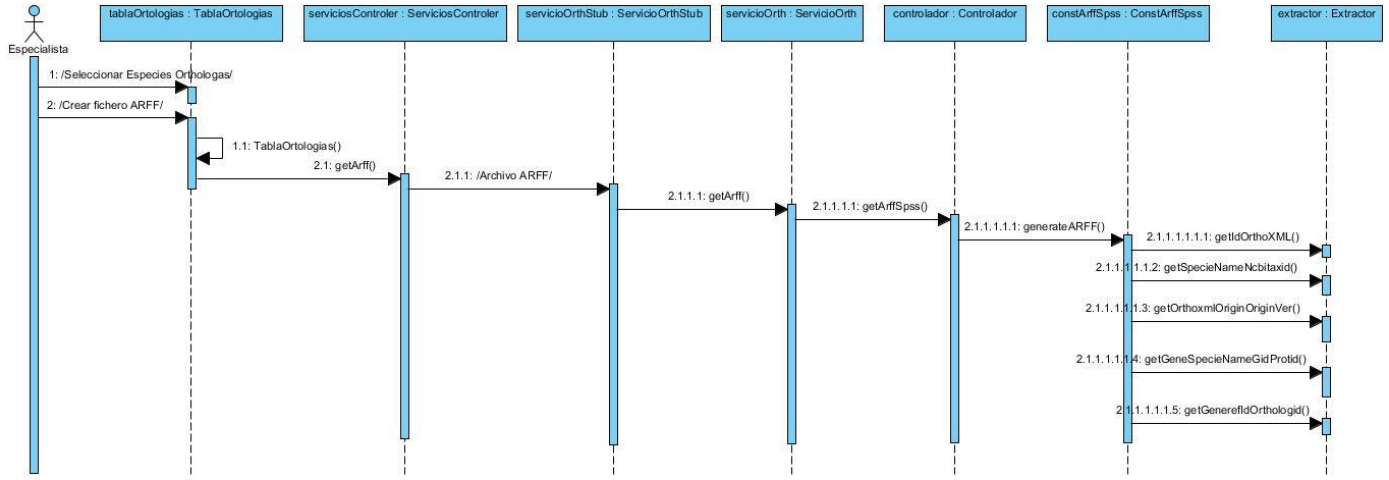



**Figura 21:** Diagrama de secuencia del escenario "Crear Fichero ARFF"

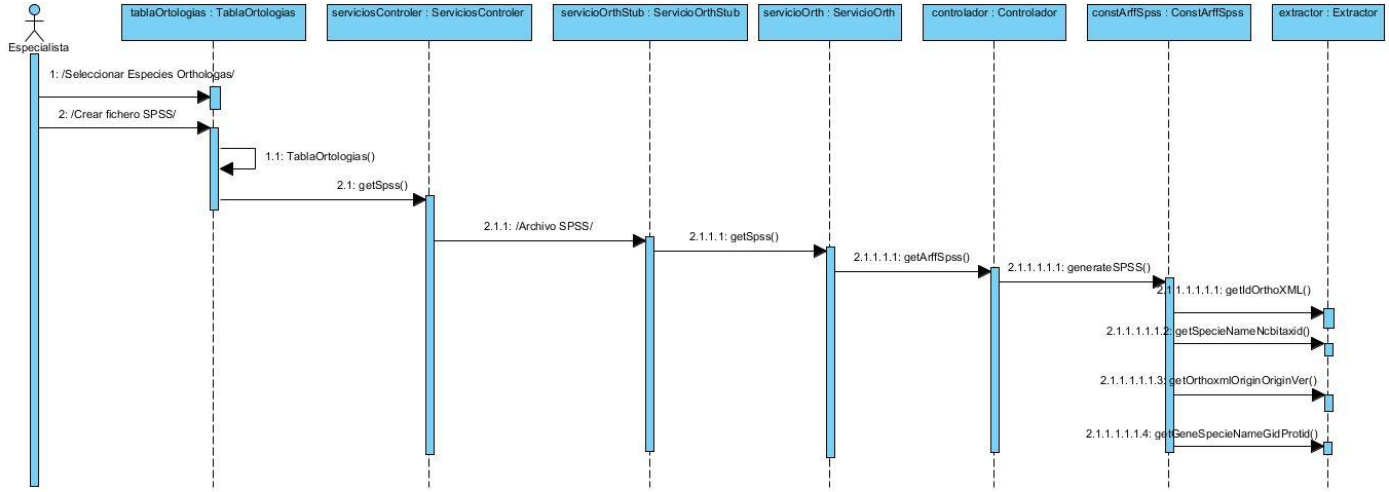



**Figura 22:** Diagrama de secuencia del escenario "Crear Fichero SPSS”