



FACULTAD DE INFORMÁTICA

TRABAJO DE FIN DE GRADO
GRADO EN INGENIERÍA INFORMÁTICA

Mención en Computación

Desarrollo de una plataforma bioinformática para el almacenamiento y consulta de datos de secuenciación masiva en el ámbito de la biología marina y su aplicación en el procesado y análisis de los mismos

Proyecto de desarrollo en investigación

Autor: Daniel Ruiz Pérez
Director: Daniel Rivero Cebrián
Directora: Vanessa Aguiar Pulido

A Coruña, a 16 de junio de 2015

Resumen

El presente proyecto pretende ayudar a comprender las relaciones existentes entre los niveles de toxicidad en el agua marina producidos por la presencia del ácido okadaico (AO) y la información genética obtenida de moluscos bivalvos, que resultan unos excelentes indicadores.

Consta de dos partes diferenciadas:

- Desarrollo de una plataforma bioinformática que permita almacenar y gestionar la gran cantidad de datos obtenidos a través de técnicas de secuenciación masiva e integrar diferentes herramientas bioinformáticas para su manejo. Dicha plataforma constituirá una herramienta de gran utilidad para investigadores del campo de la biología marina y supondrá una sustancial mejora respecto a su anterior versión (<http://chromevaloa.udc.es>), desarrollada en Perl hace años.
- Análisis de datos de naturaleza biológica almacenados en dicha plataforma, con el objetivo de encontrar relaciones entre perfiles de expresión génica y los niveles de biotoxina en el agua de mar. Además se combinarán con la aplicación de vocabularios especiales denominados ontologías, mediante las cuales se pueden enriquecer los datos de tal forma que permiten extraer más conocimiento útil. Esto derivará en la obtención de potenciales biomarcadores genotóxicos.

Por lo tanto, este proyecto contribuirá a la producción de tests moleculares para la detección y evaluación de los efectos genotóxicos del AO en zonas costeras, llevándose a cabo con la colaboración de la Florida International University.

Palabras clave:

- ✓ Aprendizaje máquina
- ✓ BigData
- ✓ Bioinformática
- ✓ Clustering
- ✓ Django
- ✓ Ontologías
- ✓ Secuenciación Masiva
- ✓ Scrum

Índice general

	Página
1. Introducción	1
1.1. Motivación	1
1.2. Objetivos	2
1.3. Estructura del documento	3
2. Planificación	5
2.1. Estimación inicial	7
2.2. Desarrollo final	7
3. Desarrollo de la herramienta	11
3.1. Antecedentes	12
3.1.1. Análisis de la situación actual	12
3.1.2. Extracción del diseño de la base de datos	13
3.2. Especificación de requisitos iniciales	16
3.3. Metodología	16
3.3.1. Scrum	19
3.3.1.1. Ciclo de vida	20
3.3.1.2. Roles	20
3.3.1.3. Documentos	21
3.3.1.4. Reuniones	22
3.3.1.5. Adaptación a la metodología	23
3.3.2. Definiciones	24
3.3.2.1. UML	24
3.3.2.2. Diagrama de Casos de Uso	24
3.3.2.3. Diagrama de clases	24
3.3.2.4. Diagrama E-R	25

3.3.2.5. Diagrama de secuencia	25
3.4. Elección de la tecnología	25
3.4.1. Lenguajes y librerías	25
3.4.1.1. Python	25
3.4.1.2. HTML5	26
3.4.1.3. CSS3	26
3.4.2. <i>Frameworks</i>	27
3.4.2.1. Django	27
3.4.3. Herramientas	28
3.4.3.1. Trello	28
3.4.3.2. Jalview	30
3.4.3.3. Clustal Omega	30
3.4.3.4. BLAST	30
3.4.3.5. yEd	30
3.4.4. Sistemas de gestión de bases de datos	31
3.4.4.1. MySQL	31
3.5. Desarrollo iterativo	31
3.5.1. Sprint 1	31
3.5.1.1. Captura de requisitos	31
3.5.1.2. Análisis	32
3.5.1.3. Diseño	38
3.5.1.4. Implementación	40
3.5.1.5. Pruebas	42
3.5.1.6. Cambios en los requisitos	45
3.5.2. Sprint 2	45
3.5.2.1. Cambios en los requisitos	45
3.5.2.2. Análisis	45
3.5.2.3. Diseño	47
3.5.2.4. Implementación	48
3.5.2.5. Pruebas	49
3.5.3. Sprint 3	53
3.5.3.1. Captura de requisitos	53
3.5.3.2. Análisis	55
3.5.3.3. Diseño	80
3.5.3.4. Implementación	85
3.5.3.5. Pruebas	85

3.5.4. Desarrollos adicionales	89
3.5.4.1. Internacionalización	89
3.5.4.2. Logo renovado	90
3.5.4.3. Panel de acciones recientes	90
4. Análisis de datos	91
4.1. Introducción	91
4.1.1. Definiciones	92
4.1.2. Objetivos	92
4.1.3. Descripción de datos experimentales	93
4.1.3.1. Creación de la base de datos	93
4.1.3.2. Explicación de los datos	94
4.1.4. Trabajos relacionados	96
4.1.5. Elección de la tecnología	97
4.1.5.1. <i>K-Means</i>	97
4.1.5.2. Matlab	97
4.2. Análisis	98
4.2.1. Aproximación 1	99
4.2.2. Aproximación 2	100
4.2.3. Aproximación 3	102
4.3. Discusión	107
5. Conclusiones y trabajo futuro	111
5.1. Conclusiones	111
5.1.1. Objetivos	111
5.1.2. Comparación con trabajos relacionados	112
5.1.3. Metodología	113
5.1.4. Problemas encontrados	114
5.2. Trabajo futuro	114
A. Manual de usuario	119
A.1. Requisitos <i>Hardware</i> y <i>Software</i>	119
A.2. Descripción de pantallas visibles al usuario estándar	119
A.2.1. Pantalla inicial	120
A.2.2. <i>Browse</i>	122
A.2.2.1. Búsqueda de <i>Contigs</i>	124
A.2.2.2. Detalles del <i>Contig</i>	125

A.2.3. Jalview	125
A.2.4. <i>Expression</i>	126
A.2.5. BLAST	127
A.2.5.1. <i>Program Help</i>	130
A.2.5.2. <i>Database Help</i>	130
A.2.6. Errores	131
A.3. Descripción de pantallas visibles a los administradores	132
A.3.1. Pantalla de <i>login</i>	132
A.3.2. Pantalla de administración inicial	133
A.3.3. Pantalla de gestión de usuarios	133
A.3.4. Pantalla de gestión de grupos	135
A.3.5. Pantalla de gestión de <i>Reads</i>	136
B. Guía de instalación	139
B.1. Instalación	139
B.2. Ejecución	140
C. Glosario de acrónimos	141
D. Glosario de términos	143
Bibliografía	145

Índice de figuras

Figura	Página
2.1. Diagrama de Gantt de la planificación inicial	6
2.2. Diagrama de Gantt de la planificación final	9
3.1. Diagrama Entidad-Relación inicial	15
3.2. Esquema del modelo de espiral con prototipado.	17
3.3. Ciclo de vida y roles de <i>Scrum</i> particularizados a este proyecto	19
3.4. <i>Burndown chart</i>	22
3.5. Capturas de la herramienta Trello	29
3.6. Casos de uso iniciales	32
3.7. Diagrama de secuencia con las operaciones necesarias para obtener el fichero de entrada de Jalview	39
3.8. Diagrama de clases del primer <i>Sprint</i>	40
3.9. Diseño de plantillas HTML de Django	41
3.10. Capturas de la herramienta Trello del primer <i>Sprint</i>	43
3.11. Gráfico de trabajo pendiente del primer <i>Sprint</i>	43
3.12. Diagrama de casos de uso de las búsquedas en las tablas	46
3.13. Diagrama de secuencia de la búsqueda de <i>Contigs</i>	48
3.14. Capturas de la herramienta Trello del segundo <i>Sprint</i>	50
3.15. Diagrama de trabajo pendiente del segundo <i>Sprint</i>	51
3.16. Diagrama de casos de uso “Gestión de usuarios”	55
3.17. Diagrama de casos de uso “Gestión de grupos”	61
3.18. Diagrama de casos de uso “Gestión de <i>Reads</i> ”	65
3.19. Diagrama de casos de uso “Gestión de <i>Contigs</i> ”	70
3.20. Diagrama de casos de uso “Gestión de <i>Clusters</i> ”	76
3.21. Diagrama Entidad-Relación de las entidades principales	82
3.22. Diagrama Entidad-Relación de usuarios, grupos y permisos	83

3.23. Diagrama de clases Final	84
3.24. Capturas de la herramienta Trello del tercer <i>Sprint</i>	86
3.25. Diagrama de trabajo pendiente del tercer <i>Sprint</i>	87
3.26. Tiempo invertido por los usuarios para completar todas las tareas	88
3.27. Tiempo promedio de cada tarea	89
3.28. Logo	90
4.1. Estructura del ácido okadaiko.	93
4.2. Proceso de extracción génica de las dos poblaciones de moluscos.	94
4.3. Diagrama de muestra de la estructura de Gene Ontology.	95
4.4. Ancestros del GOTerm Nucleus.	96
4.5. Figuras resultado de la segunda aproximación	101
4.6. Figuras resultado de la tercera aproximación	104
4.7. Número de <i>clusters</i> en función de la distancia.	106
4.8. Número de elementos de cada <i>clusters</i> en función de la distancia.	106
A.1. Panel de navegación	120
A.2. Pantalla inicial	121
A.3. Pantalla Browse: visualizar los <i>Contigs</i>	123
A.4. Pantalla de descarga de la tabla completa	124
A.5. Pantalla de los resultados de la búsqueda de <i>Contigs</i>	124
A.6. Pantalla que muestra la secuencia del <i>Contig</i> seleccionado	125
A.7. Pantalla de visualización de Jalview	126
A.8. Pantalla <i>Expression</i> : visualizar la expresión de los <i>Unigenes</i>	126
A.9. Pantalla de los resultados de BLAST	128
A.10. Pantalla de configuración de BLAST	129
A.11. Pantalla de ayuda sobre el tipo de BLAST a utilizar	130
A.12. Pantalla de ayuda sobre los contenidos de cada base de datos	131
A.13. Error 404	131
A.14. Error 500	132
A.15. Pantalla de identificación	132
A.16. Pantalla inicial de administrador	133
A.17. Pantalla de gestión de usuarios	134
A.18. Pantalla de editar usuarios 1	134
A.19. Pantalla de editar de usuarios 2	135
A.20. Pantalla de cambio de contraseña de un usuario	135
A.21. Pantalla de gestión de grupos	136

A.22.Pantalla de editar grupos	136
A.23.Pantalla de gestión de <i>Reads</i>	137
A.24.Pantalla de editar <i>Reads</i>	137

Índice de cuadros

Tabla	Página
2.1. Tabla de costes inicial	8
2.2. Tabla de costes final	8
3.1. Caso de uso “Consultar <i>Contigs</i> ”	33
3.2. Vistas usadas por el caso de uso “Consultar <i>Contigs</i> ”	33
3.3. Caso de uso “Consultar <i>Unigenes</i> ”	33
3.4. Vistas usadas por el caso de uso “Consultar <i>Unigenes</i> ”	34
3.5. Caso de uso “Consultar expresión <i>Unigenes</i> ”	34
3.6. Vistas usadas por el caso de uso “Consultar expresión <i>Unigenes</i> ”	35
3.7. Caso de uso “Obtener detalles”	35
3.8. Vistas usadas por el caso de uso “Obtener detalles”	35
3.9. Caso de uso “Alinear secuencias”	36
3.10. Vistas usadas por el caso de uso “Alinear secuencias	36
3.11. Caso de uso “Visualizar alineamientos”	37
3.12. Vistas usadas por el caso de uso “Visualizar alineamientos”	37
3.13. Caso de uso “BLAST”	37
3.14. Vistas usadas por el caso de uso “BLAST”	38
3.15. Pruebas de valores frontera del primer <i>Sprint</i>	44
3.16. Caso de uso “Búsqueda de <i>Contigs</i> por campo”	46
3.17. Vistas usadas por el caso de uso “Búsqueda de <i>Contigs</i> por campo” . .	46
3.18. Caso de uso “Búsqueda de <i>Unigenes</i> por campo”	47
3.19. Vistas usadas por el caso de uso “Búsqueda de <i>Unigenes</i> por campo” . .	47
3.20. Pruebas de valores frontera del segundo <i>Sprint</i>	52
3.21. Pruebas realizadas sobre BLAST	52
3.22. Caso de uso “Crear usuario”	56
3.23. Vistas usadas en el caso de uso “Crear usuario”	57

3.24. Caso de uso “Modificar usuario”	57
3.25. Vistas usadas en el caso de uso “Modificar usuario”	58
3.26. Caso de uso “Eliminar usuario”	58
3.27. Vistas usadas en el caso de uso “Eliminar usuario”	59
3.28. Caso de uso “Buscar usuario por nombre”	59
3.29. Vistas usadas en el caso de uso “Buscar usuario por nombre”	59
3.30. Caso de uso “Buscar usuario por e-mail”	60
3.31. Vistas usadas en el caso de uso “Buscar usuario por e-mail”	60
3.32. Caso de uso “Crear grupo”	61
3.33. Vistas usadas en el caso de uso “Crear grupo”	62
3.34. Caso de uso “Modificar grupo”	62
3.35. Vistas usadas en el caso de uso “Modificar grupo”	63
3.36. Caso de uso “Eliminar grupo”	63
3.37. Vistas usadas en el caso de uso “Eliminar grupo”	64
3.38. Caso de uso “Buscar grupo por nombre”	64
3.39. Vistas usadas en el caso de uso “Buscar grupo por nombre”	64
3.40. Caso de uso “Crear <i>Read</i> ”	65
3.41. Vistas usadas en el caso de uso “Crear <i>Read</i> ”	66
3.42. Caso de uso “Modificar <i>Read</i> ”	66
3.43. Vistas usadas en el caso de uso “Modificar <i>Read</i> ”	67
3.44. Caso de uso “Eliminar <i>Read</i> ”	68
3.45. Vistas usadas en el caso de uso “Eliminar <i>Read</i> ”	68
3.46. Caso de uso “Buscar <i>Read</i> por ID”	69
3.47. Vistas usadas en el caso de uso “Buscar <i>Read</i> por ID”	69
3.48. Caso de uso “Buscar <i>Read</i> por SEQ”	69
3.49. Vistas usadas en el caso de uso “Buscar <i>Read</i> por SEQ”	69
3.50. Caso de uso “Crear <i>Contig</i> ”	70
3.51. Vistas usadas en el caso de uso “Crear <i>Contig</i> ”	71
3.52. Caso de uso “Modificar <i>Contig</i> ”	71
3.53. Vistas usadas en el caso de uso “Modificar <i>Contig</i> ”	72
3.54. Caso de uso “Eliminar <i>Contig</i> ”	73
3.55. Vistas usadas en el caso de uso “Eliminar <i>Contig</i> ”	73
3.56. Caso de uso “Buscar <i>Contig</i> por ID”	74
3.57. Vistas usadas en el caso de uso “Buscar <i>Contig</i> por ID”	74
3.58. Caso de uso “Buscar <i>Contig</i> por descripción”	74
3.59. Vistas usadas en el caso de uso “Buscar <i>Contig</i> por descripción”	74

3.60. Caso de uso “Buscar <i>Contig</i> por SEQ	75
3.61. Vistas usadas en el caso de uso “Buscar <i>Contig</i> por SEQ”	75
3.62. Caso de uso “Crear <i>Cluster</i> ”	76
3.63. Vistas usadas en el caso de uso “Crear <i>Cluster</i> ”	77
3.64. Caso de uso “Modificar <i>Cluster</i> ”	77
3.65. Vistas usadas en el caso de uso “Modificar <i>Cluster</i> ”	78
3.66. Caso de uso “Eliminar <i>Cluster</i>	78
3.67. Vistas usadas en el caso de uso “Eliminar <i>Cluster</i> ”	79
3.68. Caso de uso “Buscar <i>Cluster</i> por ID	79
3.69. Vistas usadas en el caso de uso “Buscar <i>Cluster</i> por ID”	79
3.70. Caso de uso “Buscar <i>Cluster</i> por SEQ	80
3.71. Vistas usadas en el caso de uso “Buscar <i>Cluster</i> por SEQ”	80
3.72. Tareas de las pruebas de usuario	88
4.1. Tabla de distancias de cada <i>cluster</i>	108
4.2. Importancia de los atributos de cada <i>Cluster</i>	109

Capítulo 1

Introducción

Índice general

1.1. Motivación	1
1.2. Objetivos	2
1.3. Estructura del documento	3

1.1. Motivación

El ácido okadaico (AO) representa una de las biotoxinas de mayor relevancia producidas en los fenómenos de contaminación marina natural, conocidos como mareas rojas, lo cual constituye un serio problema con respecto al consumo humano de marisco. La ingesta de productos del mar cultivados en dichas condiciones de contaminación marina conlleva efectos nocivos para la salud humana, así como para los organismos marinos. Estos efectos se pueden categorizar en dos tipos principales: la patología aguda asociada a la intoxicación diarreica por ingesta de marisco o DSP(*Diarrhetic Shellfish Poisoning*); y los efectos derivados de la exposición crónica o a largo plazo debido al gran potencial genotóxico del AO. Es por ello que la monitorización de esta biotoxina en las zonas costeras ha sido un objetivo de interés en las últimas décadas, enfrentándose este problema a través de distintas estrategias. Sin embargo, no ha sido hasta hace poco que, gracias a la disponibilidad de información genética y molecular, así como al conocimiento en mayor profundidad de los mecanismos involucrados en el mantenimiento de la integridad del genoma, se ha comenzado a abrir un nuevo camino en cuanto a

métodos de biomonitorización del AO.

El presente proyecto pretende estudiar las relaciones existentes entre los niveles de toxicidad en el agua marina producidos por la presencia del AO y la información genética obtenida de organismos indicadores como los moluscos bivalvos. La información obtenida a partir de técnicas de secuenciación masiva será procesada e interpretada utilizando métodos bioinformáticos.

Por lo tanto, será necesario:

- Desarrollar un recurso que permita almacenar la gran cantidad de datos obtenidos a partir de este tipo de estudios biológicos.
- Facilitar su acceso a través de un portal o aplicación web.
- Integrar diferentes herramientas bioinformáticas para su manejo.

Partiendo de este recurso, se llevarán a cabo distintos tipos de análisis de datos combinados con la aplicación de vocabularios especiales denominados ontologías, los cuales permiten enriquecer los datos de tal forma que sea posible extraer más conocimiento útil, con el objetivo último de encontrar potenciales biomarcadores de la contaminación por AO. El presente proyecto contribuirá a la producción de tests moleculares para la detección y evaluación de los efectos genotóxicos del AO en zonas litorales. Este proyecto se realizará en colaboración con el grupo *Bioinformatics Research Group* (BioRG) del Departamento de Computación y Ciencias de la Información y con el grupo *Chromatin Structure and Evolution Research Group* (CHROMEVOL) del Departamento de Ciencias Biológicas, ambos pertenecientes a la Florida International University.

1.2. Objetivos

Al tener el presente proyecto dos partes diferentes, los objetivos que persiguen cada una difieren ampliamente. El desarrollo de la web pretende lo siguiente:

- Centralizar en una web las bases de datos biológicas del grupo CHROME-VOL, para que los investigadores del mismo puedan acceder y actualizar sus datos de manera rápida.

- Además, esta web estará accesible a todo el mundo debido a que las bases de datos que enlaza son de interés general. Esto propiciará el aumento del conocimiento disponible sobre las biotoxinas marinas.

Por otra parte, el análisis de datos pretende ayudar a la comprensión de las relaciones existentes entre las diferentes secuencias genéticas extraídas de los organismos centinela.

1.3. Estructura del documento

EL resto del documento cubre los siguientes apartados:

- Capítulo 2 (Planificación): Este capítulo integra la planificación económica y temporal de todo el proyecto, tanto de la parte de desarrollo como la de análisis. Así mismo, también integra el desarrollo final del proyecto, con su coste real en tiempo y dinero.
- Capítulo 3 (Desarrollo de la herramienta): Este capítulo contempla toda la parte de desarrollo de la herramienta web. Desde la contextualización, metodología y elección de tecnología, pasando por un análisis de requisitos y diseño de la herramienta hasta un seguimiento del proceso de implementación de la misma y las pruebas realizadas.
- Capítulo 4 (Análisis de datos): Este capítulo contiene una exhaustiva definición de la problemática presentada y de todas las aproximaciones realizadas hasta alcanzar un resultado satisfactorio.
- Capítulo 5 (Conclusiones y trabajo futuro): Este capítulo contiene las conclusiones extraídas, tanto del desarrollo de la herramienta web como del análisis de datos realizado. Además, contiene las posibles extensiones que se podrían realizar en un futuro de las dos vertientes del proyecto.

Capítulo 2

Planificación

Índice general

2.1. Estimación inicial	7
2.2. Desarrollo final	7

El objetivo de la planificación de un proyecto es obtener estimaciones para calcular de una forma razonable los costes y los plazos utilizados para su elaboración. Ésta planificación ha tenido en cuenta la parte de la herramienta web, el análisis de los datos y la realización de la memoria.

La dedicación estimada al presente proyecto fue de 4 horas al día. Sin embargo, no ha sido posible cumplirla siempre por causas ajenas al mismo.

Para la planificación del proyecto se ha utilizado la herramienta OpenProj, que es libre e intuitiva [1]. Se la conoce como el sustituto de Microsoft Project, herramienta que no se usó por razones económicas. OpenProj gestiona recursos y tareas, permitiendo su visualización en un Diagrama de Gantt. Su objetivo es exponer de manera gráfica el tiempo previsto para la realización de diferentes tareas [2]. Será la herramienta que se usará para visualizar la planificación del proyecto.



Figura 2.1: Diagrama de Gantt de la planificación inicial

2.1. Estimación inicial

La planificación inicial del proyecto se puede observar en la Figura 2.1, comenzándose el día 3/11/2014 y con una fecha de finalización prevista para el día 6/3/2014. Podría parecer un proyecto de reducido tamaño en tiempo, pero al haber manifestado el cliente en reiteradas ocasiones su intención de cambiar los requisitos, se ha dejado un amplio margen de maniobra para futuros imprevistos. Es por la elevada propensión a cambios por lo que se decidió cambiar la metodología inicialmente establecida (espiral) por otra que permita gestionar los cambios con un menor impacto (ver Sección 3.3).

El proyecto se divide en dos partes perfectamente diferenciables, que se realizarán de forma paralela en la medida de lo posible. El desarrollo de la herramienta consta de unas fases previas necesarias y las fases de desarrollo propiamente dichas (análisis, diseño, implementación y pruebas). La fase de análisis de los datos biológicos también posee unas fases previas y una serie de aproximaciones siguiendo así mismo una metodología en espiral.

Esta planificación, desglosada para cada uno de los perfiles necesarios, se puede consultar en el Cuadro 2.1. Las tareas que realiza cada uno de estos perfiles son las siguientes:

- Consultor: Analiza problemas empresariales y recomienda acciones encaminadas a la mejora del producto.
- Analista: Es el encargado de realizar el diseño del sistema y de la obtención de algoritmos, además de obtener los requisitos del mismo.
- Diseñador: Es el encargado de definir la arquitectura del sistema.
- Programador: El encargado de escribir, depurar y mantener el código del programa.

2.2. Desarrollo final

En la Figura 2.2 se puede observar el coste real a fecha de cierre de proyecto. La fecha de inicio del mismo fue el día 3/11/2014, tal y como había sido planificado. En cambio el proyecto finalizó el día 17/06/2015.

Perfil	Tiempo dedicado (h)	Precio por hora (€/h)	Coste total (€)
Consultor	132	30	3.960
Analista	300	30	9.000
Diseñador	158	20	3.160
Programador	182	7	1.274
Total	—	—	17.394

Cuadro 2.1: Tabla de costes inicial

Como se puede observar, esta planificación tiene una estructura de metodología ágil (*Scrum*), en vez de en espiral como el inicial (ver Sección 3.3). El proceso de desarrollo de la herramienta propiamente dicha consta de una serie de *Sprints*, a su vez subdivididos en las clásicas fases del desarrollo *software*.

Ha sido imposible parallelizar completamente la fase de análisis de datos biológicos, debido a que sólo había una persona para realizar todo el trabajo. Ésto ha dado como resultado la espaciación en el tiempo de las tareas de análisis, ya que se ha dado una mayor prioridad al desarrollo completo de la herramienta web, de acuerdo con las preferencias del cliente. Se ha utilizado una metodología en espiral, como había sido planificado inicialmente.

El aumento de requisitos y el retraso en alguna de las tareas ha dado como resultado una desviación de tres meses y medio en la planificación inicial, con una casi duplicación del coste económico del proyecto. Este desvío se acumula sobre todo en la parte de desarrollo de la herramienta web, habiendo realizado tres *Sprints* de 21 días de duración cada uno. También se ha realizado una aproximación adicional en la parte de análisis de datos, aumentando su duración.

El coste final del proyecto puede consultarse en el Cuadro 2.2, desglosado para cada uno de los perfiles necesarios.

Perfil	Tiempo dedicado (h)	Precio por hora (€/h)	Coste total (€)
Consultor	192	30	5.760
Analista	598	30	17.940
Diseñador	244	20	4.880
Programador	388	7	2.716
Total	—	—	31.296

Cuadro 2.2: Tabla de costes final

2. Planificación

9

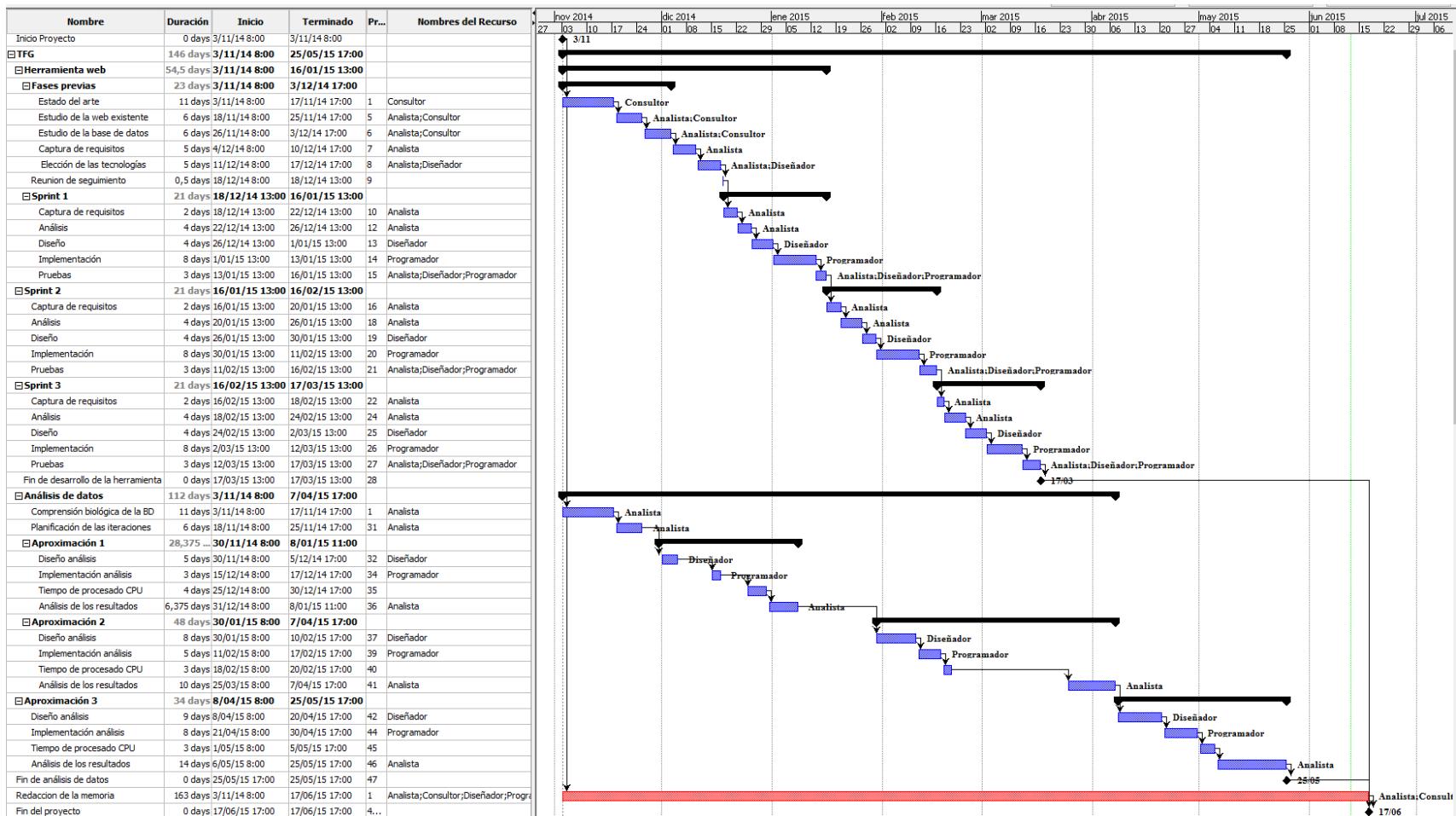


Figura 2.2: Diagrama de Gantt de la planificación final

Capítulo 3

Desarrollo de la herramienta

Índice general

3.1. Antecedentes	12
3.1.1. Análisis de la situación actual	12
3.1.2. Extracción del diseño de la base de datos	13
3.2. Especificación de requisitos iniciales	16
3.3. Metodología	16
3.3.1. Scrum	19
3.3.2. Definiciones	24
3.4. Elección de la tecnología	25
3.4.1. Lenguajes y librerías	25
3.4.2. <i>Frameworks</i>	27
3.4.3. Herramientas	28
3.4.4. Sistemas de gestión de bases de datos	31
3.5. Desarrollo iterativo	31
3.5.1. Sprint 1	31
3.5.2. Sprint 2	45
3.5.3. Sprint 3	53
3.5.4. Desarrollos adicionales	89

Neste capítulo se abordará todo el desarrollo de la herramienta web. Desde las definiciones tecnológicas importantes y un análisis del estado de la cuestión, pasando por la elección de la metodología y tecnología aplicada, hasta el seguimiento de la implementación del producto.

3.1. Antecedentes

En esta sección se pondrá al lector en situación, con las definiciones tecnológicas pertinentes y el actual estado de la cuestión.

3.1.1. Análisis de la situación actual

En este proyecto no será necesario un exhaustivo análisis de mercado ni usuarios, debido a que ya existe una plataforma que tomar como referencia. Dicha plataforma es la web CHROMEVALOA (*CHROMatin EVALuation of OA*) que se supone se sustituirá por el resultado de la primera parte de este proyecto.

Esta actualización se debe a lo siguiente:

- **Obsolescencia tecnológica:** La antigua web estaba completamente realizada en Perl. Éste fue el lenguaje de programación por excelencia de los investigadores en bioinformática hace años. En cambio, hoy en día, sus características quedan algo anticuadas, por lo que Python ha ganado terreno y muchas herramientas ya no se actualizan más para Perl. Es por ello que los responsables de la misma vieron necesaria esta actualización.
- **Escasa escalabilidad:** Algo recurrente en la ingeniería informática es la escalabilidad de los sistemas. La antigua web CHROMEVALOA fue codificada sin la ayuda de ningún *framework*, ni utilizando ningún patrón ingenieril como puede ser el MVC (Model View Controller). Ésto quiere decir que en un mismo fichero CGI (*Common Gateway Interface*) estaba la conexión con la base de datos, el procesado de los mismos y la generación del HTML y CSS a partir de ellos. Además, los parámetros de conexión a la base de datos estaban en texto plano y el código destacaba por su es-

casa legibilidad y ausencia de comentarios. Ésto supuso un gran obstáculo debido a que el presente proyecto tomaba esa web como base.

- **Contenidos desactualizados:** Al seguir los enlaces disponibles en las bases de datos, muchos de ellos no funcionan como sería esperado; algunos deberían realizar una búsqueda en la página de Gene Ontology que no realizan, enlaces al NCBI rotos, atributos obsoletos, etc.
- **Escasa eficiencia:** El antiguo sistema dejaba mucho margen a la mejora debido a que el acceso a sus bases de datos era lento y tedioso. Ésto era ocasionado por muchos factores, entre los que cabe destacar que se mostraba el contenido completo de la base de datos, sin realizar ninguna clase de paginación ni optimización en las consultas, hasta el punto de seleccionar atributos que luego no se usaban en ningún sitio. El presente proyecto pretende atajar estos problemas de raíz.
- **Elevada propensión a errores:** Las escasas pruebas realizadas con el antiguo sistema llevaban a que fallara al utilizar sus herramientas. Un ejemplo de esto es que si se dejaba en blanco uno de los campos (concretamente el *e-valor*) en la herramienta BLAST, la web entera fallaba.
- **Interfaz antigua:** Una de las características que más destaca de una web es la interfaz gráfica. La vista de la antigua web CHROMEVALOA es contenido estático, con un estilo más propio de la década pasada.
- **Escasas funcionalidades:** El creciente grupo CHROMEVOL prevé precisar más funcionalidades en un futuro cercano, resultando atractivas características como una gestión de usuarios y tablas.

El coste de implantación de esta nueva tecnología será muy escaso debido a los servidores disponibles para la antigua versión. Además, al suponer una actualización, los usuarios sólo tendrán que aprender a usar las nuevas herramientas ofrecidas debido a que saben utilizar correctamente las demás.

3.1.2. Extracción del diseño de la base de datos

Al estar ya la web CHROMEVALOA hecha, su sistema de bases de datos también estaba montado. Ésto podría parecer una ventaja, pero no lo fue, entre otras

cosas porque no se disponía de ningún diagrama Entidad-Relación, y obtener su estructura supuso aplicar técnicas de ingeniería inversa. Esta tarea no fue sencilla debido a la gran cantidad de tablas superfluas e información redundante existentes, además de la escasa calidad del código disponible. Finalmente, y después de comprobarlo con la persona responsable de la antigua web, perteneciente al grupo CHROMEVOL, el diagrama de la base de datos existente sería el mostrado en la Figura 3.1.

Se procederá a explicar brevemente lo que representa cada entidad:

- **Read:** Fragmentos de ARN o tránscritos (ver Glosario C) que se obtienen al secuenciar el material genético de los moluscos.
- **Cluster:** *Reads* agrupados en base a un criterio de similaridad pre-establecido.
- **Contig:** Es la secuencia de consenso extraída como representante de cada *Cluster*. Cada letra de esta secuencia es la más frecuente de las letras en esa posición de todas las lecturas del *Cluster*.
- **ContigSEQ:** Posee, para cada *Cluster*, su cadena de ADN.
- **Fulllengther:** Indica si el tránscribo estaba entero o era solo un fragmento.
- **Expression:** Indica si con el tratamiento (exposición al AO) ese gen se estaba expresando mas o menos que la referencia (población de control, con condiciones normales). Ésto se denomina expresión diferencial. Es posible que los genes que aparecen con sus niveles de expresión alterados no necesariamente lo hacen por culpa del ácido okadaico, si no por cualquier otra razón que desconocemos. Sin embargo son buenos candidatos.
- **Unigenes:** Son el resultado de comparar los *Contigs* mediante BLAST (ver Sección 3.4.3.4) con el contenido de la base de datos UniGene del NCBI. Esta BD pretende proporcionar una vista organizada del transcriptoma (ver Glosario C).
- **Genes:** Describe qué genes se estaban expresando en las condiciones del experimento.

3. Desarrollo de la herramienta

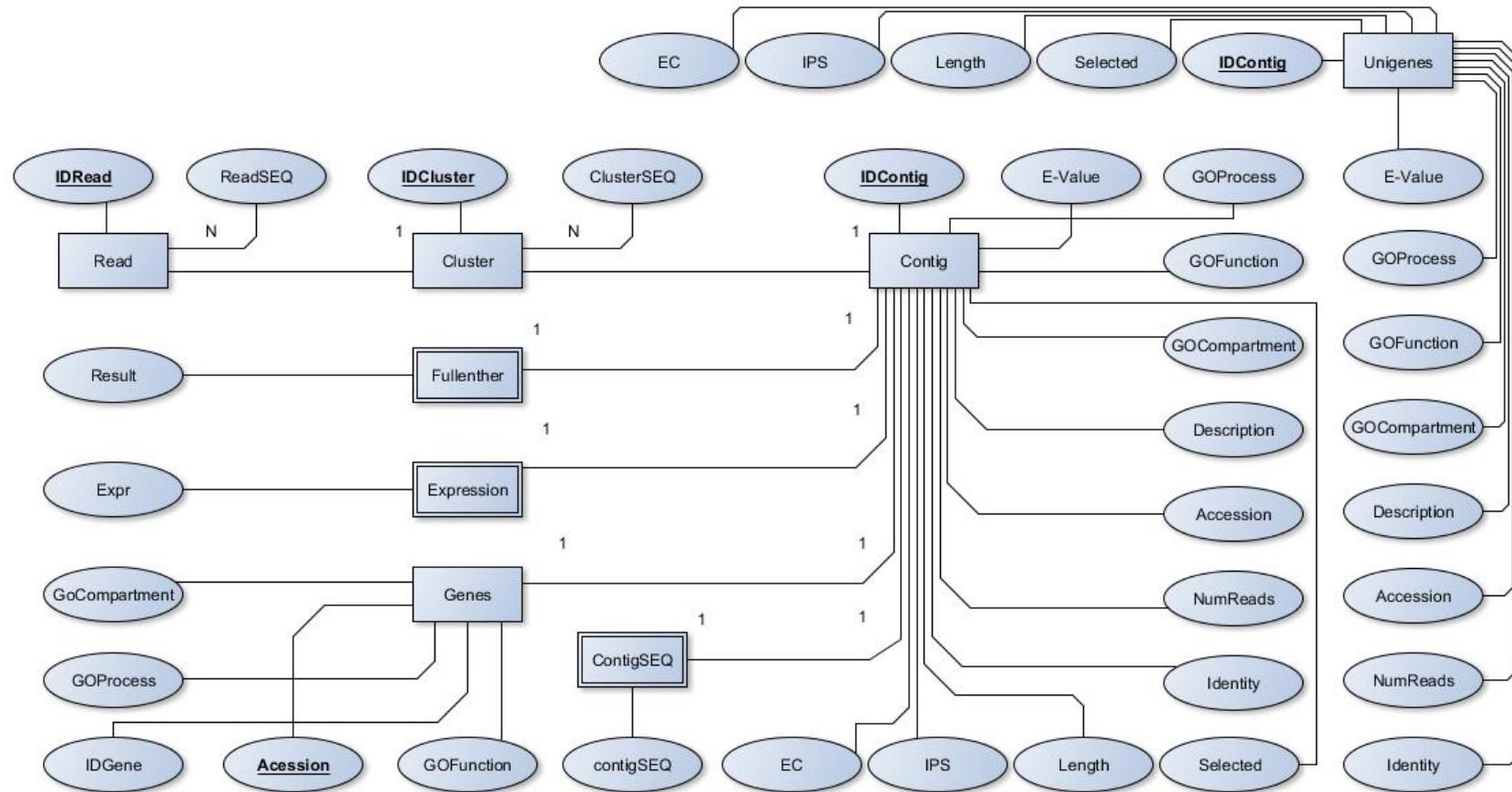


Figura 3.1: Diagrama Entidad-Relación inicial

En la Figura 3.1 puede observarse claramente que la tabla *Contig* es el centro de todas las demás. Adicionalmente, la tabla *Unigenes* posee los mismos campos que la tabla de *Contigs*, pues son instancias de ésta que cumplen ciertas características. Aunque lo lógico hubiera sido tener una tabla o un atributo que identificara qué *Clusters* también deberían estar en los *Unigenes* y no duplicarlos, el hecho es que así es como se encontraba la base de datos. Además, existían 20 tablas más que no se comentarán en la memoria por su escasa relevancia.

3.2. Especificación de requisitos iniciales

El presente proyecto empezó como una actualización de la antigua web. No se precisaba ninguna funcionalidad adicional más que la introducción de mejoras significativas, sobre todo desde el punto de vista ingenieril. Por lo tanto, los requisitos iniciales son los siguientes:

- Mostrar al usuario el contenido de la base de datos de forma paginada y permitir que navegue a través de su información, la cual estará enlazada con diversas bases de datos biológicas.
- Integrar la herramienta BLAST para que sea posible comparar nuevas secuencias por homología con las pre-existentes.
- Integrar la herramienta Jalview, para que sea posible visualizar el alineamiento de los distintos Reads que forman un Contig.

Además, se espera que esta lista crezca a lo largo del desarrollo del proyecto.

Sólo se ha identificado una entidad que interactúe con el sistema, que será el investigador interesado en consultar la información de las bases de datos y los resultados de las herramientas.

3.3. Metodología

Inicialmente, en este proyecto, se esperaba seguir una metodología con prototipado en espiral y así se reflejó en el anteproyecto del mismo. En la Figura 3.2 puede verse un diagrama de esta metodología. Una vez que el cliente definió un boceto de los requisitos iniciales se pudo intuir que los cambios de los mismos

iban a ser frecuentes, algo no deseable en este método. Además, se trata de un modelo difícil de implementar en sistemas pequeños y se precisa experiencia para la identificación de riesgos.

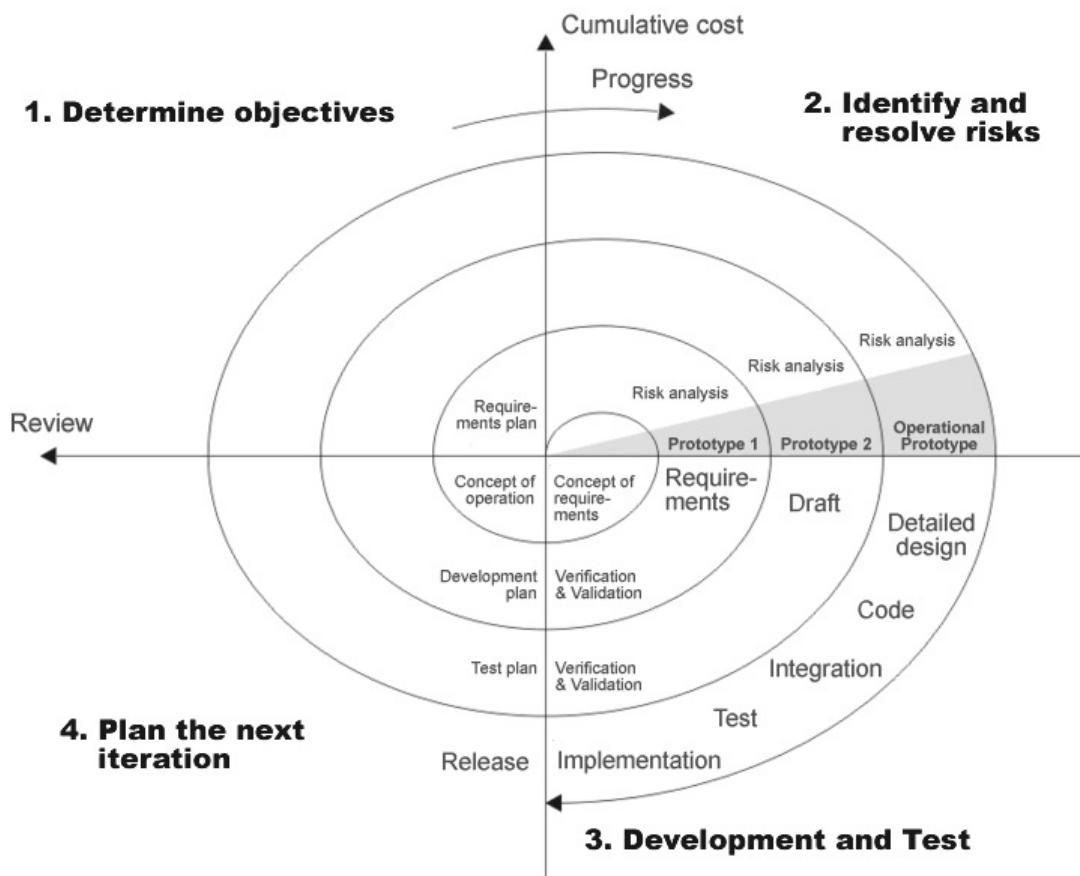


Figura 3.2: Esquema del modelo de espiral con prototipado.

El desarrollo de la web puede definirse como un proyecto de baja envergadura, en el cual el cliente no tiene claro los requisitos al comienzo del mismo. Debido a esto, se ha decidido optar por utilizar una metodología ágil, dado que se ajustaba mejor a sus características que una metodología tradicional. Sus ventajas pueden sintetizarse en los siguientes puntos:

- Maximización de la satisfacción del cliente mediante una rápida y continua entrega de software funcional. En la mayoría de las metodologías tradicionales, solamente se tiene un producto final y entregable al finalizar el ciclo de vida.

- Énfasis en una frecuente comunicación con el cliente y entre el equipo, lo que minimiza malentendidos y evita tener que repetir trabajo.
- Algo esencial en este proyecto, en el que el cliente no tiene claro el producto final, es que el usuario puede ir refinando su idea sobre el mismo a medida que se va desarrollando. Esto es debido a la inherente facilidad para la adaptación al cambio de las metodologías ágiles, incluso en etapas finales del desarrollo. Dicha facilidad, unida a la ya mencionada frecuente comunicación, ayuda a disminuir costes considerablemente.

Por la contra, sus principales desventajas son las siguientes:

- Dificultad de estimar esfuerzo y recursos en los proyectos grandes, algo que no afecta demasiado debido a que este es un proyecto relativamente pequeño.
- Poco énfasis en diseño y documentación, que se intentará solucionar mediante constancia, buenas prácticas y una más exhaustiva fase de diseño.

Una de las muchas diferencias con las metodologías tradicionales, es la forma de estimar el esfuerzo que requiere cada tarea, que se hace mediante los puntos de historia. Éstos no reflejan un valor del esfuerzo en horas, sino una manera de relacionar y dimensionar la complejidad de unas tareas con respecto a otras. Cabe destacar que esta valoración es propia de cada equipo de desarrollo, por lo que no se puede utilizar para comparar la complejidad de una tarea con las de otros equipos.

Los valores que se utilizarán en este proyecto serán los de una estimación exponencial, como la serie de Fibonacci: 1,2,3,5,8,13,21... [3]. Ésto se justifica mediante la aplicación de la Teoría de la Información, que provee de una fuerte justificación matemática de por qué se recomienda una aproximación exponencial para estimar proyectos que siguen una metodología ágil.

Debido a la escueta fase de diseño habitual en las metodologías ágiles, el equipo de desarrollo ha decidido utilizar herramientas más propias de las metodologías tradicionales como pueden ser los casos de uso. Éstos se usarán como una versión más formal, exhaustiva e informativa de las historias de usuario que deberían utilizarse al seguir una metodología ágil.

3.3.1. Scrum

Dentro de los modelos de desarrollo ágiles, probablemente el más conocido y especialmente indicado para desarrollo web con equipos de trabajo pequeños es *Scrum*. Se centra en un desarrollo flexible y holístico, donde el equipo de trabajo se comporta como una unidad para alcanzar el objetivo común.

Scrum adopta una aproximación empírica aceptando que el problema no se puede definir o entender completamente, centrándose en su lugar, en maximizar la capacidad del equipo de realizar rápidas entregas y responder a requerimientos emergentes mediante una sucesión de los denominados *Sprints*. Define un conjunto de prácticas y roles que pueden tomarse como punto de partida para establecer el proceso de desarrollo que se ejecutará durante un proyecto.

En la Figura 3.3 se pueden observar los diferentes roles y fases del ciclo de vida de *Scrum* aplicado a este proyecto en particular. Más adelante se explicarán en profundidad.

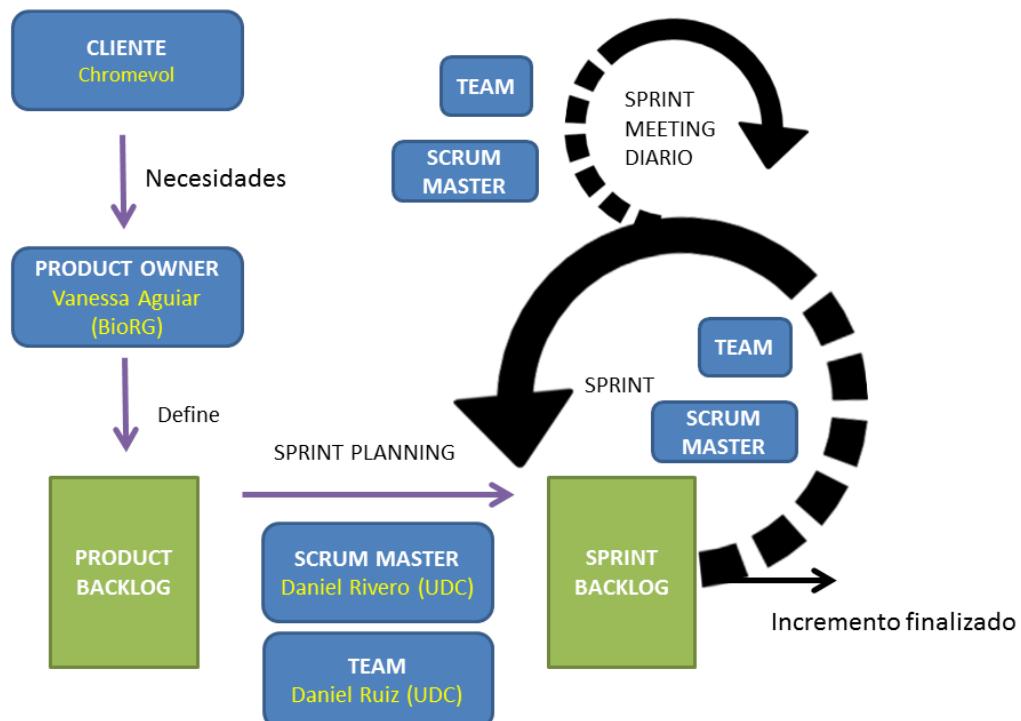


Figura 3.3: Ciclo de vida y roles de *Scrum* particularizados a este proyecto

3.3.1.1. Ciclo de vida

El ciclo de un proyecto *Scrum* se compone de las siguientes fases:

- **Fase de planificación y diseño de alto nivel.** En esta fase se define el sistema que va a ser desarrollado y se crea una lista de requisitos del producto conocidos hasta el momento. Éstos son priorizados y ordenados en función de su coste e importancia estimados. Dicha lista es constantemente revisada y actualizada en cada iteración por el equipo de trabajo para lograr los objetivos en la iteración siguiente. También se incluye la definición del equipo del proyecto, herramientas y otros recursos, valoración y control de riesgos, entrenamiento necesario y una verificación para la aceptación de gerencia.
- **Fase de desarrollo.** En esta fase el sistema se lleva a cabo en *Sprints*, que integran un ciclo iterativo donde se desarrolla la funcionalidad para producir nuevos incrementos. Cada *Sprint* incluye las fases tradicionales del desarrollo software: requisitos, análisis, diseño, implementación, prueba y entrega. El equipo realizará reuniones al principio y al final de cada *Sprint*, además de otra diaria para planificar el día. Es recomendable que la duración de éstos sea constante y definida por el equipo en base a su propia experiencia.
- **Fase de cierre.** Esta fase es completada con la aceptación por parte del cliente, con todos los requisitos cumplidos y estando el sistema listo para el lanzamiento. Esta fase incluye su integración, pruebas del sistema y documentación.

3.3.1.2. Roles

La adopción de esta metodología de desarrollo requiere la asignación de los siguientes roles:

- **Propietario del producto (*Product Owner*):** Es la persona responsable de asegurarse que el equipo realiza lo solicitado por el cliente y de transmitir las prioridades en el proceso de desarrollo. Además, se encarga de la gestión del *Product Backlog* (inventario de características que se desea obtener) y

de la financiación necesaria para el proyecto, decidiendo como debe ser el resultado final de la inversión. En este proyecto este papel lo cumple Vanessa Aguiar Pulido, perteneciente al grupo BioRG (*Bioinformatics Research Group*) de la FIU (*Florida International University*).

- **Responsable del funcionamiento de Scrum (*Scrum Master*):** Es el que debe conseguir el buen funcionamiento de *Scrum* a lo largo del proyecto, eliminando los obstáculos si es necesario. Interactúa con el cliente y el equipo y coordina los encuentros diarios. En este proyecto este papel lo cumple Daniel Rivero Cebrián.
- **Equipo de desarrollo:** Es el encargado de entregar cada incremento al final de los *Sprints*. Se autogestiona y autoorganiza, de forma que está capacitado para tomar decisiones sobre cómo realizar el trabajo. En este proyecto el equipo de desarrollo lo compone el autor del proyecto.
- **Cliente:** Participa en las tareas relacionadas con la especificación de los requisitos del Product Backlog. En este proyecto este papel lo cumple el grupo CHROMEVOL.

3.3.1.3. Documentos

Durante el proceso de *Scrum* se generarán los siguientes documentos:

- **Lista de tareas del producto (*Product Backlog*):** Es el inventario de características que el propietario del producto desea obtener, ordenado por orden de prioridad (requisitos). Se parte de la visión del resultado que se desea obtener y se va evolucionando durante el desarrollo, estando en constante cambio y siendo accesible a todas las personas que intervienen en el desarrollo. El responsable de éste es el *Product Owner*.
 - **Lista de tareas del Sprint (*Sprint Backlog*):** Documento que contiene la lista de los trabajos que realizará el equipo durante el *Sprint* para generar el incremento previsto. Es el equipo quien asume el compromiso de la ejecución.
 - **Gráficos de trabajo pendiente (*Burndown charts*):** Muestra la velocidad a la que se están completando los objetivos/requisitos y permite
-

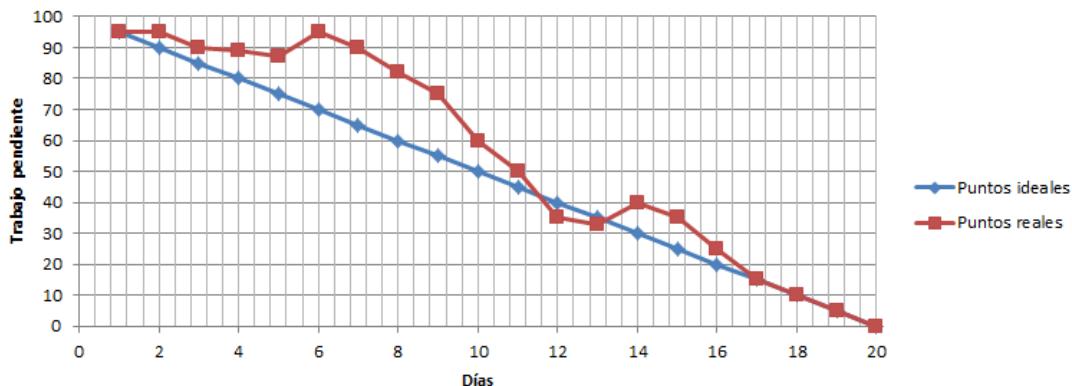


Figura 3.4: *Burndown chart*

extrapolar si el equipo podrá completar el trabajo en el tiempo estimado. Para su obtención es necesario ir dibujando una línea que conecte los puntos de todos los *Sprints* completados. Lo normal es que esta línea sea descendente hasta llegar al eje horizontal, momento en el que el proyecto se ha terminado. Si durante el proceso se añaden requisitos, la recta tendrá pendiente ascendente en determinados segmentos y si se modifican algunos *ítems*, la pendiente variará o incluso será nula en algunos tramos. Un ejemplo puede verse en la Figura 3.4.

3.3.1.4. Reuniones

Se deben llevar a cabo diferentes reuniones durante el desarrollo de un producto mediante *Scrum*. Éstas son:

- **Reunión diaria:** Encuentro de 15 minutos que se realiza al comienzo de una jornada, en el que cada miembro contesta a estas tres preguntas: ¿qué hice desde ayer?, ¿qué voy a hacer hoy? y ¿he tenido algún problema que me haya impedido alcanzar mi objetivo?
- **Reunión de planificación de Sprint:** Cita efectuada al comienzo de cada iteración, en la cual tiene lugar una planificación sobre lo que se llevará a cabo (selección del trabajo, preparación del *Sprint Backlog*, y estimación del tiempo necesario para su completa conclusión)
- **Reunión de revisión del Sprint:** Encuentro efectuado al finalizar cada

ciclo en el cual se revisa el trabajo realizado y no realizado y se presenta el mismo a los interesados.

- **Retrospectiva del *Sprint*:** Cita efectuada al finalizar cada iteración, en la que todos los miembros del equipo dejan sus impresiones sobre el *Sprint* recién superado. El propósito de esta reunión es realizar una mejora continua del proceso.

3.3.1.5. Adaptación a la metodología

Este proyecto se realizará utilizando *Scrum* sin grandes cambios, pero con las siguientes salvedades:

- El equipo de desarrollo está formado por una sola persona, lo cual supone una desventaja, pero no resultará algo crítico.
- El cliente no tiene claros los requisitos completos del producto, con lo que éstos irán cambiando durante el desarrollo.
- No será viable realizar reuniones diarias, debido a la diferente disponibilidad de las personas implicadas.

Los roles son los indicados en la Figura 3.3 de la página 19:

- **Propietario del producto:** Vanessa Aguiar Pulido.
- **Responsable del funcionamiento de *Scrum*:** Daniel Rivero Cebrián.
- **Cliente:** Grupo CHROMEVOL, de la FIU.
- **Equipo:** Daniel Ruiz Pérez.

Con respecto a los *Sprints*, se decidió fijar su duración en tres semanas, reuniéndose el equipo cada semana para revisar el estado del *Sprint*. Las reuniones con el Propietario del producto fueron cada dos semanas y, debido a que éste residía en otro país, se realizaron mediante la herramienta Skype. Además se llevaron a cabo frecuentes comunicaciones mediante correo electrónico.

3.3.2. Definiciones

A continuación se realizará una breve descripción de los conceptos necesarios para el completo entendimiento de esta memoria.

3.3.2.1. UML

UML (*Unified Modeling Language*) es el lenguaje de modelado de sistemas software por excelencia [4] estando respaldado por el OMG (*Object Management Group*); ésta es una organización sin ánimo de lucro, que desde 1989 se dedica a promover el uso de tecnologías orientadas a objetos mediante guías y el establecimiento de diversos estándares de tecnologías orientadas a objetos. El grupo está formado por compañías y organizaciones de *software* como *HP*, *IBM*, *Sun Microsystem* y *Apple Computer*.

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 y funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes reutilizables

UML ofrece una gran diversidad de diagramas, pero para los objetivos que persigue este proyecto será suficiente con los Diagramas de Casos de Uso.

3.3.2.2. Diagrama de Casos de Uso

Un diagrama de casos de uso es una representación de la interacción del usuario con el sistema, mostrando la relación entre el usuario y los diferentes casos de uso en los cuales está involucrado. UML define una notación gráfica para representar casos de uso llamada modelo de casos de uso, pero cabe destacar que es sólo una representación, y precisa de una descripción más detallada para reflejar fielmente la naturaleza de un caso de uso [5].

3.3.2.3. Diagrama de clases

En UML, un diagrama de clases es un tipo de diagrama estático que describe la estructura de un sistema mostrando las clases que lo componen, sus atributos, operaciones y las relaciones entre los objetos [5].

3.3.2.4. Diagrama E-R

Un diagrama o modelo entidad-relación, o E-R (*Entity relationship*), es una herramienta para el modelado de datos que permite representar las entidades relevantes de un sistema de información así como sus interrelaciones y propiedades [6]. Sus componentes principales son los siguientes:

- **Entidad:** Representa un concepto del mundo real con existencia independiente y que es diferenciable con las otras entidades.
- **Atributo:** Cada una de las características que definen o identifican a una entidad. Cada una tiene valores específicos asignados para cada uno de sus atributos, de forma que es posible su identificación única.
- **Relación:** Describe cierta dependencia entre entidades o permite la asociación de las mismas.

3.3.2.5. Diagrama de secuencia

En UML, un diagrama de secuencia es un diagrama de interacción que muestra cómo los procesos, objetos y actores cooperan unos con otros y cuál es su orden [7].

3.4. Elección de la tecnología

Una vez se ha seleccionado *Scrum* como metodología a seguir a lo largo del proyecto, y con un boceto de los requisitos iniciales, el equipo de desarrollo ha procedido a seleccionar la tecnología para el desarrollo. Esta sección consta de un listado de todas las herramientas, lenguajes y tecnologías utilizadas con una breve explicación de las mismas.

3.4.1. Lenguajes y librerías

Se analizarán los lenguajes y las librerías que han sido

3.4.1.1. Python

Python es un lenguaje de programación multiparadigma y multiplataforma. Pertenece a los lenguajes interpretados con tipado dinámico [8]. Para la adminis-

tración de memoria utiliza una combinación de conteo de referencias y recolector de basura, unido a una resolución dinámica de nombres o *late binding*. La filosofía de Python hace especial hincapié en la legibilidad y transparencia del código. Una de sus ventajas son las completas librerías que ofrece, lo que lo hace ideal para infinidad de tareas, entre ellas el análisis de datos, teniendo una gran comunidad de desarrolladores en el ámbito de la bioinformática [9]. Una importante librería de computación biológica es Biopython, que posee algunas de las herramientas que se necesitarán a lo largo del proyecto [10]. Además, el cliente ha especificado que es requisito imprescindible que la web sea realizada en Python.

Más concretamente, se ha utilizado la versión 2.7 de Python debido a que es la versión más madura y la probabilidad de que tenga errores es mucho menor. Además, el soporte de bibliotecas es peor en la nueva versión (3.x). Cuando esta última alcance mayor madurez, y en función de las necesidades del cliente, se podría transitar a la nueva versión.

3.4.1.2. HTML5

HTML (*HyperText Markup Language*) es un estándar que sirve de referencia para la elaboración de páginas web en sus diferentes versiones. Define una estructura básica y un código para la definición de contenido de una página web, como texto e imágenes, entre otros. Es un estándar a cargo de la W3C, organización dedicada a la estandarización de casi todas las tecnologías ligadas a la web, sobre todo en lo referente a su escritura e interpretación [11].

HTML5 es la quinta gran revisión, que especifica dos variantes de sintaxis: un clásico HTML (la variante conocida como HTML5) y otra conocida como sintaxis XHTML5 que deberá ser servida como XML.

En este proyecto, se utiliza esta tecnología para crear la estructura de la vista de la aplicación web.

3.4.1.3. CSS3

CSS (*Cascade Style Sheet*) es un lenguaje estándar usado para definir la presentación de un documento escrito en HTML o XML (o por extensión XHTML) [12]. Su idea del desarrollo es la separación de la estructura de un documento y su presentación. Sus principales ventajas son:

- Control centralizado de la presentación de un sitio web.
- Aumento de la accesibilidad.
- Facilidad y rapidez en el estudio y carga del HTML.

CSS3 es la última versión existente hasta la fecha de realización del proyecto. Incorpora varias mejoras respecto a las versiones anteriores, como puede ser la incorporación de transiciones, degradados, sombras, etc.

En el proyecto se ha utilizado esta tecnología para la creación de la presentación de la aplicación web.

3.4.2. Frameworks

Debido a la relativa complejidad de la web, y a la necesidad de mejorar las pésimas prácticas de programación en la web actual, se ha optado por utilizar un *framework* web. Ésta es una herramienta que define un marco de trabajo que facilita el desarrollo de sitios web dinámicos, definiendo un conjunto de prácticas y conceptos, además de proporcionar herramientas para su aplicación [13]. Las ventajas más notables de la utilización de esta tecnologías son las siguientes:

- Clara organización de archivos y carpetas.
- Abstracción de forma sencilla de la base de datos mediante la API del framework, además de separar la vista de la lógica de negocio.
- Incrementa la seguridad de la aplicación solucionando errores conocidos.
- Una vez familiarizado con el *framework*, incrementa la velocidad de desarrollo del proyecto debido a que minimiza la cantidad de código necesario, evitando, como se suele decir, “reinventar la rueda”.

3.4.2.1. Django

Django es un framework de desarrollo web de código abierto, escrito en Python, que respeta el paradigma conocido como MTV (*Model Template View*) [14]. Éste es parecido al MVC(*Model View Controller*) [15], pero lo que se llamaría controlador se llama vista y lo que se llamaría vista se llama plantilla.

Estos dos patrones de arquitectura software tienen como objetivo separar, en una aplicación, la lógica de negocio de los datos de la interfaz de usuario. Se basan en las ideas de reutilización de código y la separación de conceptos, características que buscan facilitar la tarea de desarrollo de aplicaciones y su posterior mantenimiento.

La meta fundamental de Django es facilitar la creación de sitios web complejos. Pone énfasis en la reutilización, la conectividad y extensibilidad de componentes, el desarrollo rápido y el principio No Te Repitas o DRY (*Don't Repeat Yourself*). Python es usado en todas las partes del framework, incluso en configuraciones, archivos, y en los modelos de datos, lo cual se adapta completamente a los requisitos del cliente. Además, proporciona un potente motor de plantillas, con un lenguaje propio, permitiendo herencia entre las mismas. Entre otras muchas funcionalidades, implementa seguridad por defecto contra ataques como pueden ser *cross site scripting*, *cross site request forgery*, *SQL injection*, *clickjacking*, etc [16].

3.4.2.1.1. Django-tables2

Django-tables2 es una extensión del framework que simplifica la tarea de convertir conjuntos de datos en tablas HTML. Tiene soporte nativo para paginación y ordenado, además de multitud de opciones de personalización [17].

3.4.3. Herramientas

En esta sección se analizarán todas las herramientas que se usarán en el proyecto.

3.4.3.1. Trello

Trello es una herramienta web colaborativa que organiza el proyecto en tableros, dentro de los cuales se adjuntan listas de tareas, por lo que se ajusta perfectamente a las necesidades de un proyecto que siga una metodología *Scrum* y será la herramienta utilizada para la gestión del mismo [18]. Se dispondrá de un tablero para el *Product Backlog*, en el que se plasmarán las necesidades del cliente y un tablero para el *Sprint Backlog* en el que estarán las funcionalidades seleccionadas para realizar en cada *Sprint*. Además, se utilizará un tablero *In*



Figura 3.5: Capturas de la herramienta Trello

progress para el trabajo que está siendo realizado y otro *Done* para el trabajo del *Sprint* ya completado, además de uno adicional *Blocked*, para el trabajo que, por algún motivo, no se puede realizar.

Cada tarea es descompuesta en subtareas, pudiendo marcar en un *checkbox* si está realizada o no. Además se permite asignar a cada tarea un coste estimado siguiendo la serie de Fibonacci, y un color en función de su importancia. Esto puede verse en la Figura 3.5. La subfigura a) muestra el código de colores usado para identificar la importancia de la tarea, la subfigura b) muestra un ejemplo de cómo Trello permite dividir una tarea en varias, indicando si se ha completado o no. Finalmente, la Subfigura c) muestra la vista general de una tarea, con su código de color, importancia estimada (8) y su porcentaje de realización.

Como es costumbre en los proyectos *Scrum*, se irán adjuntando capturas de la situación del tablero en cada momento para facilitar el seguimiento.

3.4.3.1.1. Scrum for Trello

Extensión de la herramienta Trello que incorpora multitud de características,

como los diagramas de trabajo pendiente y la posibilidad de caracterizar cada tarea con un coste estimado [18].

3.4.3.2. Jalview

Herramienta bioinformática para ver y editar el alineamiento múltiple de secuencias. Se utiliza para realizar análisis de árboles filo-genéticos y de componentes principales (PCA), así como para realizar gráficos y explorar estructuras moleculares y anotaciones [19]. En este proyecto se utiliza para visualizar los datos de las secuencias de consenso resultantes de la aplicación de la herramienta Clustal.

3.4.3.3. Clustal Omega

Herramienta bioinformática ampliamente utilizada para realizar alineamientos múltiples de secuencias [20]. En este proyecto se utiliza para obtener los datos de entrada de la herramienta Jalview.

3.4.3.4. BLAST

Herramienta bioinformática que compara una secuencia problema (ADN, ARN o proteínas) con todas las secuencias de una base de datos, tanto local como remota. Encuentra las secuencias que tienen mayor parecido a la secuencia problema [21]. Utiliza algoritmos heurísticos que permiten calcular la significación de los resultados, además de integrarse con Biopython. Se integrará con la web para permitir a los usuarios realizar consultas sobre la base de datos propia.

3.4.3.5. yEd

yEd es una herramienta gratuita para la generación de diagramas profesionales, especialmente en el contexto de la informática. Destaca su calidad para ser gratis, y en este proyecto se usará para toda la parte de diagramas (ER, diagramas de casos de uso, diagramas de secuencia y diagramas de clases) [22].

3.4.4. Sistemas de gestión de bases de datos

Un Sistema de Gestión de Bases de Datos (SGBD) es un conjunto de programas que permiten el almacenamiento, modificación y extracción de la información en una base de datos, además de proporcionar herramientas para añadir, borrar, modificar y analizar los datos. Existen multitud de ellas, pero se procederá a explicar la que se usó en este proyecto.

3.4.4.1. MySQL

MySQL es un sistema de gestión de bases de datos relacional, multihilo y multiusuario que se ofrece bajo una licencia GNU-GPL. Es muy utilizado en desarrollo web, debido a su gran lista de ventajas (velocidad, facilidad de configuración, baja carga computacional,etc) [23]. Es la base de datos utilizada en la versión antigua de la web de CHROMEVALOA y será el sistema que se utilizará en este proyecto por todas sus ventajas y para evitar problemas de compatibilidad.

3.5. Desarrollo iterativo

Ésta es la sección principal del desarrollo de la herramienta, en la que se detallarán los diversos *Sprints* realizados.

3.5.1. Sprint 1

Con los requisitos iniciales (Sección 3.2) y el diseño de la base de datos completado (Sección 3.1.2), el equipo de desarrollo se dispuso a comenzar el primer *Sprint*. El objetivo del mismo es mostrar las bases de datos integrando las herramientas BLAST y Jalview.

3.5.1.1. Captura de requisitos

Los requisitos iniciales ya estaban establecidos antes de escoger la metodología, por lo que en este *Sprint* no será necesario esta fase porque las funcionalidades deseadas no han cambiado. Estos requisitos pueden observarse en la Sección 3.2

3.5.1.2. Análisis

El único actor contemplado en este *Sprint* es el investigador nombrado en la Sección 3.2, que será el principal usuario de la herramienta. En todos los diagramas se le denominará simplemente “Usuario”.

Los casos de uso extraídos de los requisitos pueden observarse en la Figura 3.6.

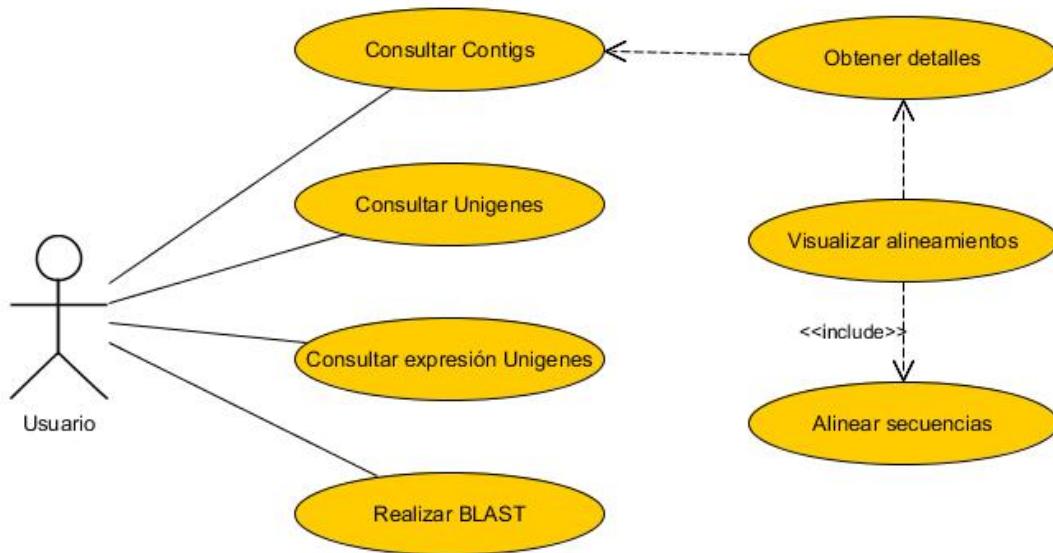


Figura 3.6: Casos de uso iniciales

- Caso de uso: Consultar *Contigs*

Cuando el usuario quiere consultar los *Contigs* almacenados en la base de datos, lo hace en la pestaña “CONTIGS”.

- Precondiciones:
 - Ninguna.
 - El escenario del caso de uso se puede ver en el Cuadro 3.1.
 - Postcondiciones:
 - Ninguna.
 - Posibles errores:

Paso	Actor	Acción
1	Usuario	Selecciona la pestaña “CONTIGS”
2	Aplicación	Muestra los campos relevantes de los <i>Contigs</i> almacenados

Cuadro 3.1: Caso de uso “Consultar *Contigs*”

Vista	Descripción
Tabla de <i>Contigs</i>	Contiene los campos más relevantes de los <i>Contigs</i> almacenados

Cuadro 3.2: Vistas usadas por el caso de uso “Consultar *Contigs*”

- Error de conexión con la base de datos.
- Las vistas usadas se muestran en el Cuadro 3.2.

▪ **Caso de uso: Consultar *Unigenes***

Cuando el usuario quiere consultar los *Unigenes* almacenados en la base de datos, lo hace en la pestaña “UNIGENES”.

- Precondiciones:
 - Ninguna.
- El escenario del caso de uso se puede ver en el Cuadro 3.3.
- Postcondiciones:
 - Ninguna.
- Posibles errores:

Paso	Actor	Acción
1	Usuario	Selecciona la pestaña “UNIGENES”
2	Aplicación	Muestra los campos relevantes de los <i>Unigenes</i> almacenados

Cuadro 3.3: Caso de uso “Consultar *Unigenes*”

Vista	Descripción
Tabla de <i>Unigenes</i>	Contiene los campos más relevantes de los <i>Unigenes</i> almacenados

Cuadro 3.4: Vistas usadas por el caso de uso “Consultar *Unigenes*”

Paso	Actor	Acción
1	Usuario	Selecciona la pestaña “EXPRESSION”
2	Aplicación	Muestra la expresión de los <i>Unigenes</i> almacenados

Cuadro 3.5: Caso de uso “Consultar expresión *Unigenes*”

- Error de conexión con la base de datos.
- Las vistas usadas se muestran en el Cuadro 3.4.

■ Caso de uso: Consultar expresión *Unigenes*

Cuando el usuario quiere consultar la expresión de los *Unigenes* almacenados en la base de datos, lo hace en la pestaña “EXPRESSION”.

- Precondiciones:
 - Ninguna.
- El escenario del caso de uso se puede ver en el Cuadro 3.5.
- Postcondiciones:
 - Ninguna.
- Posibles errores:
 - Error de conexión con la base de datos.
- Las vistas usadas se muestran en el Cuadro 3.6.

■ Caso de uso: Obtener detalles

Cuando el usuario quiere consultar la secuencia de un *Contig* o un *Unigen*.

- Precondiciones:

Vista	Descripción
Tabla de expresión <i>Unigenes</i>	Contiene los campos relevantes de la expresión de los <i>Unigenes</i> almacenados

Cuadro 3.6: Vistas usadas por el caso de uso “Consultar expresión *Unigenes*”

Paso	Actor	Acción
1	Usuario	Selecciona el ID de uno de los <i>Contig</i> o <i>Unigenes</i> en las tablas
2	Aplicación	Muestra el ID y la secuencia correspondiente, dando la posibilidad de alinearla y visualizarla

Cuadro 3.7: Caso de uso “Obtener detalles”

- Ninguna.
- El escenario del caso de uso se puede ver en el Cuadro 3.7.
- Postcondiciones:
 - Ninguna.
- Posibles errores:
 - Error de conexión con la base de datos.
 - *Contig* no existente.
- Las vistas usadas se muestran en el Cuadro 3.8.

■ **Caso de uso: Alinear secuencias**

Cuando el usuario quiere realizar mediante la herramienta Clustal Omega, el alineamiento de la secuencia de un *Contig* o *Unigen* con todos los *Reads* de los *Clusters* que lo forman.

- Precondiciones:

Vista	Descripción
Vista de detalles	Muestra el ID y la secuencia del <i>Contig</i> . Además permite realizar y visualizar el alineamiento de esa secuencia

Cuadro 3.8: Vistas usadas por el caso de uso “Obtener detalles”

Paso	Actor	Acción
1	Usuario	Selecciona la opción de “Alinear secuencias” en la vista de detalles
2	Aplicación	Ejecuta la herramienta Clustal Omega y permite descargar el fichero de alineamiento obtenido

Cuadro 3.9: Caso de uso “Alinear secuencias”

Vista	Descripción
Vista de detalles	Muestra el ID y la secuencia del <i>Contig</i> . Además permite realizar y visualizar el alineamiento de esa secuencia

Cuadro 3.10: Vistas usadas por el caso de uso “Alinear secuencias”

- Se tiene un *Contig* de partida.
- El escenario del caso de uso se puede ver en el Cuadro 3.9.
- Postcondiciones:
 - Se crea un fichero de alineamiento.
- Posibles errores:
 - Error de conexión con la base de datos.
 - Error de Clustal Omega.
- Las vistas usadas se muestran en el Cuadro 3.10.

■ Caso de uso: Visualizar alineamientos

Cuando el usuario de la base de datos necesita comparar todos los *Reads* de los *Clusters* que forman el *Contig*, lo hace mediante la herramienta Jalview.

- Precondiciones:
 - Se dispone de un fichero de alineamiento de partida.
- El escenario del caso de uso se puede ver en el Cuadro 3.11.
- Postcondiciones:
 - Ninguna.

Paso	Actor	Acción
1	Usuario	Selecciona la opción de “Comparar con Jalview” en los detalles de un <i>Contig</i>
2	Aplicación	Se descarga y ejecuta Jalview

Cuadro 3.11: Caso de uso “Visualizar alineamientos”

Vista	Descripción
Detalles del <i>Contig</i>	Muestra la secuencia del <i>Contig</i> con el botón de llamar a Jalview

Cuadro 3.12: Vistas usadas por el caso de uso “Visualizar alineamientos”

- Posibles errores:
 - Error en la descarga.
- Las vistas usadas se muestran en el Cuadro 3.12.

■ Caso de uso: Realizar BLAST

Cuando el usuario de la base de datos necesita comparar una secuencia con todas las disponibles en la base de datos, lo hace mediante la herramienta BLAST.

- Precondiciones:
 - El usuario posee al menos una secuencia que comparar.
- El escenario del caso de uso se puede ver en el Cuadro 3.13.

Paso	Actor	Acción
1	Usuario	Selecciona la pestaña “BLAST”
2	Aplicación	Muestra el formulario de configuración de la llamada a BLAST
3	Usuario	Cubre el formulario y pulsa Subir
4	Aplicación	Procesa y muestra los resultados de la consulta

Cuadro 3.13: Caso de uso “BLAST”

Vista	Descripción
Formulario de configuración	Permite añadir todos los datos
Resultados obtenidos	Presenta los resultados de forma amigable

Cuadro 3.14: Vistas usadas por el caso de uso “BLAST”

- Postcondiciones:
 - Ninguna.
- Posibles errores:
 - La secuencia no es válida.
 - El fichero no es válido.
 - Error de conexión con la base de datos.
- Las vistas usadas se muestran en el Cuadro 3.14.

3.5.1.3. Diseño

La clase de los *Contigs* se utilizará para la vista “Browse”, en la que se mostrará una tabla con los datos relevantes de los mismos (Consultando el campo *result* de la clase *Fulllengther*). Cuando el usuario selecciona el ID de uno, se le muestran los detalles de ese *Contig*, más concretamente su secuencia. Es en esta vista en la que el usuario podrá realizar el alineamiento de la secuencia. Al seleccionar la acción correspondiente, se realizarán consultas sobre las clases para acceder a todos los *Reads* de los *Clusters* que forman el *Contig* y poder llamar a Clustal Omega. La salida de esta herramienta es un fichero descargable. Después de esto y, en caso de querer visualizar su alineamiento, se llamará a la herramienta Jalview, que se descargará y ejecutará en local. Esta secuencia puede resultar algo complicada, por lo que se ha desglosado en la Figura 3.7.

3. Desarrollo de la herramienta

39

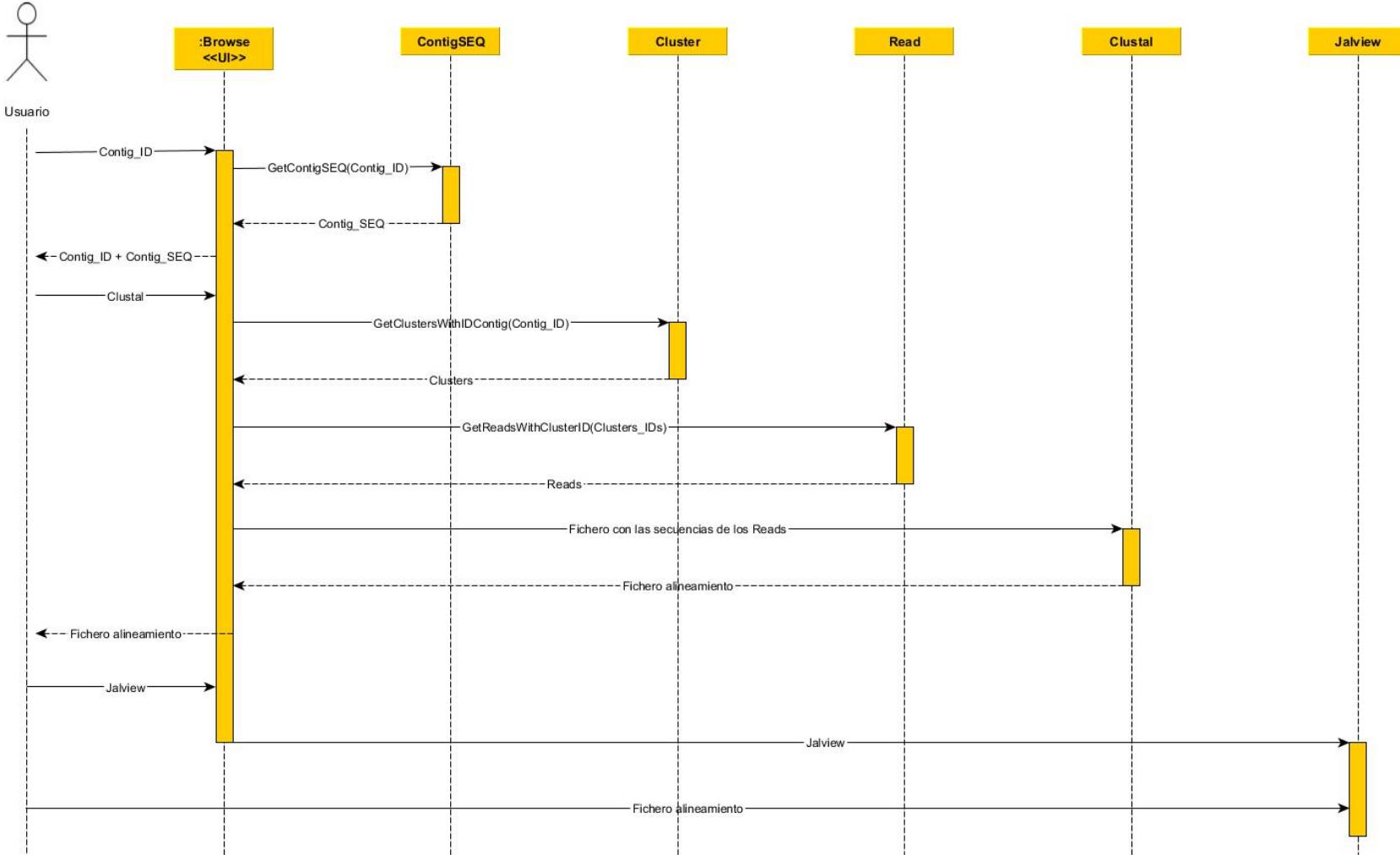


Figura 3.7: Diagrama de secuencia con las operaciones necesarias para obtener el fichero de entrada de Jalview

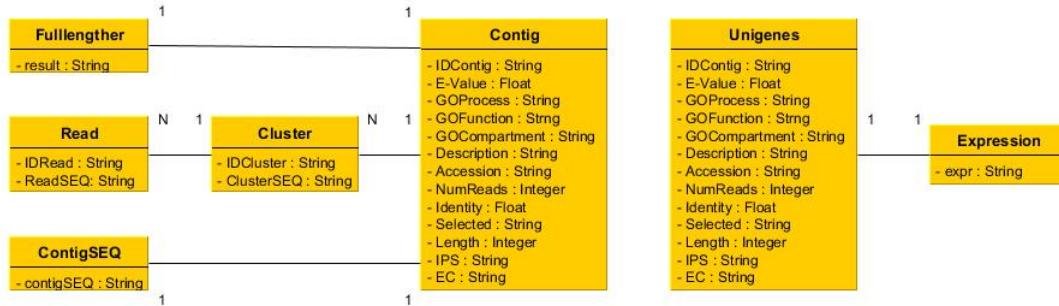


Figura 3.8: Diagrama de clases del primer *Sprint*

En la vista “BLAST” habrá un formulario con los parámetros de la llamada. Esta consulta no se realiza sobre las clases si no sobre librerías que contiene los datos de secuenciación en un formato adecuado, las cuales están almacenadas en el servidor.

Mientras, en la vista “Expression” se mostrará la expresión de los *Unigenes*, con sus tres campos más relevantes, y en el apartado “Unigenes”, aparecerán todos los datos de los mismos.

Como no se precisa de ninguna operación más que simplemente obtener, mostrar y ordenar, el diagrama de clases que se usará en esta iteración, construido a partir del modelo Entidad-Relación de la Figura 3.1, es el mostrado en la Figura 3.8. En él se pueden observar las clases y las relaciones entre las mismas. Una clase *Cluster* se relaciona con muchas entidades de la clase *Read*, y la clase *Contig* con muchas de *Cluster*. Todas ellas poseen sus atributos de forma interna excepto la clase *Contig*, cuya secuencia se encuentra en la clase *ContigSEQ*, e información sobre su longitud en la clase *Fulllengther*. La clase de los *Unigenes* contiene exactamente los mismos atributos que *Contigs*, debido a que está compuesta por instancias especiales de ésta, además, posee una relación con *Expression* que indica su expresión. No se precisa que ninguna de ellas implemente ningún método específico por la relativa simplicidad de las operaciones a realizar.

3.5.1.4. Implementación

Al ser el presente el primer *Sprint*, surgieron problemas con la tecnología. La fase de configuración del framework Django resultó más tediosa de lo esperado.

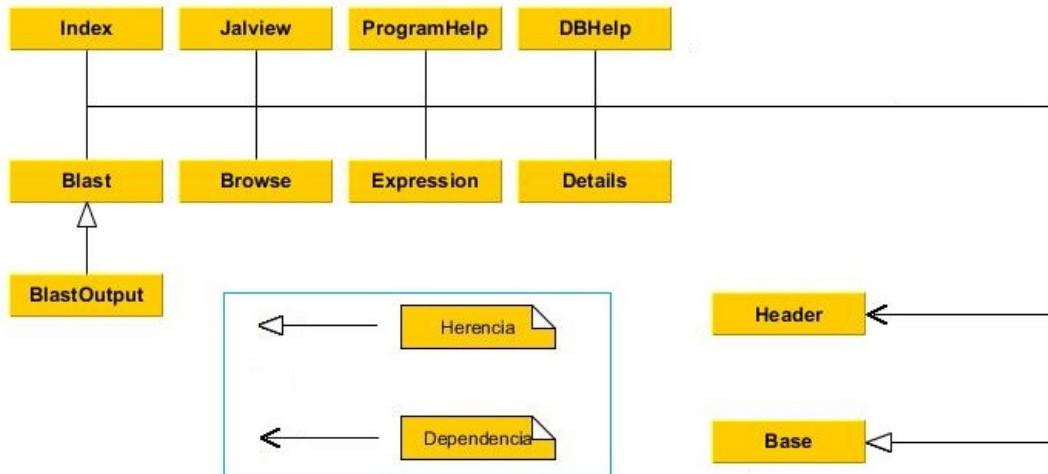


Figura 3.9: Diseño de plantillas HTML de Django

El equipo de desarrollo decidió dedicar más tiempo a dotar de una mejor parte gráfica a la web, lo que ocupó gran parte del tiempo del *Sprint*. Ésta decisión se tomó con el objetivo de aportar mayores beneficios en un futuro, anteponiendo las necesidades del cliente. Además, crear el sistema de herencia de plantillas HTML de Django de forma correcta y una estructura de CSS jerarquizada atrasaron la el *Sprint*.

Este diseño de la jerarquía entre las plantillas de Django es el mostrado en la Figura 3.9. Hay una plantilla Base, la cual extienden todas las demás. En ésta se cargan todos los ficheros estáticos como pueden ser los CSS y se define una estructura a seguir por el resto de las plantillas, que irán completando las características que esta plantilla base define pero no implementa. Algo parecido sucede con la plantilla cabecera, la cual usan todas las demás. Ésta es la encargada de construir el menú de navegación entre plantillas, recibiendo como parámetro la pestaña actual para realizar las acciones en consecuencia. Por último, la plantilla BlastOutput hereda todo el comportamiento de Blast, y rellena el bloque que ésta había dejado sin implementar.

El propietario del producto manifestó el deseo del cliente de rehacer la base de datos, pero que todavía no sabía cómo la quería. Ante esta situación se optó por conectarse de forma remota a la base de datos de la antigua web de CHROMEVALOA como solución provisional. El hecho de que en mitad del *Sprint* el

servidor dejara de funcionar durante una semana paró por completo el desarrollo debido a que Django no permite arrancar el proyecto sin conectarse a la base de datos. Para evitar estas situaciones, una vez volvió a estar operativo el servidor, se realizó un clonado en local del mismo con el objetivo evitar más retrasos.

Así, al finalizar este *Sprint* se desarrolló toda la estructura de la web, con la interfaz terminada para las características actuales. Además, se completaron los requisitos de mostrar las tablas, integrarlas con las bases de datos biológicas establecidas y permitir navegación paginada por las mismas. Todo esto se hizo con ayuda de la herramienta Django-tables2 que, a partir de un modelo de Django genera el HTML de tablas totalmente configurables mediante código, pudiendo editar todas sus características. Además se añadió un botón que permitiese descargar las tablas en una hoja de cálculo.

En la Figura 3.10 se puede observar el seguimiento de *Scrum* realizado en la herramienta Trello. La imagen a) muestra la lista de tareas del producto (*Product backlog*), que representa la totalidad de características que el propietario del producto desea obtener. La imagen b) muestra la lista de tareas del *Sprint* (*Sprint backlog*), que contiene la lista de trabajos que el equipo de desarrollo realizará en esta iteración. Cabe destacar que al iniciar el *Sprint* se esperaba realizar la totalidad de las tareas del *Product backlog*. La imagen c) muestra el estado de las listas al finalizar el *Sprint*. No se ha completado la totalidad de funcionalidades estimadas para esta iteración, por lo que se pospuso para la siguiente la integración con las herramientas BLAST y Jalview. Así, el gráfico de trabajo pendiente del primer *Sprint* es el mostrado en la Figura 3.11.

3.5.1.5. Pruebas

Debido a que este *Sprint* se ha realizado con ayuda de Django y Django-tables2, no resultaría apropiado realizar pruebas exhaustivas debido a que se trata de un framework fiable que ya ha sido probado con más detenimiento del que se podría lograr en este trabajo. Sin embargo, sí se ha optado por comprobar el correcto funcionamiento del mismo para evitar errores por parte del programador. Ésto se ha realizado mediante un análisis de los valores límite, método que se basa en la evidencia experimental de que los errores suelen aparecer con mayor probabilidad en los extremos. Puede verse en el Cuadro 3.15.

3. Desarrollo de la herramienta

43

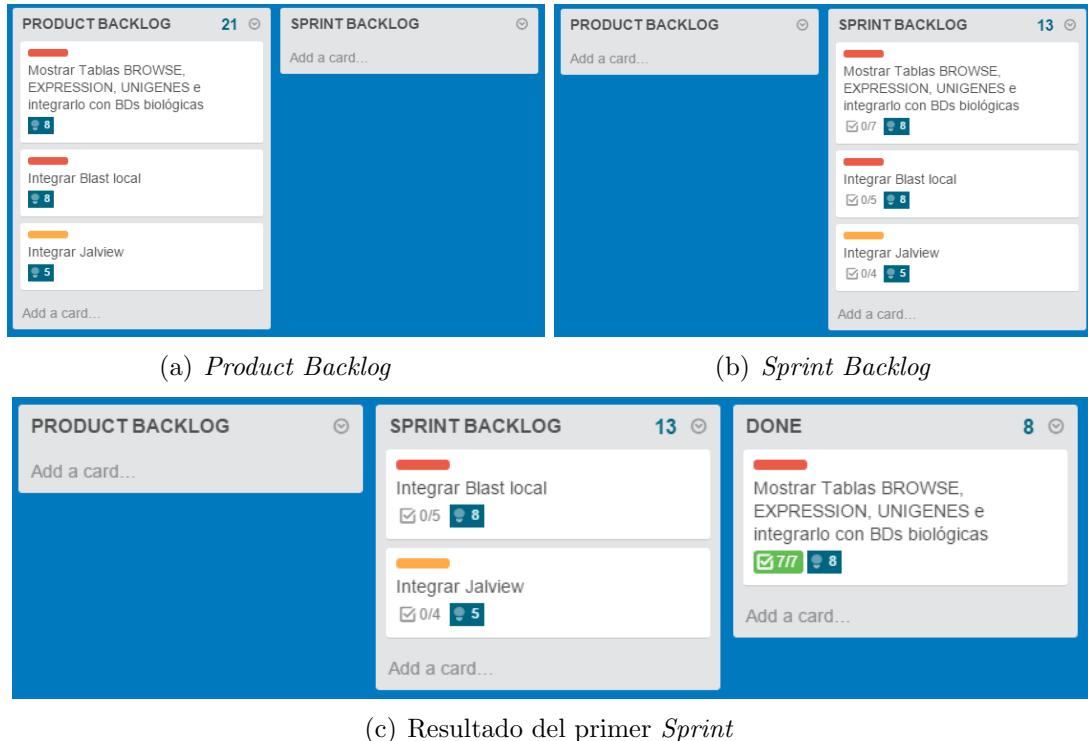


Figura 3.10: Capturas de la herramienta Trello del primer *Sprint*

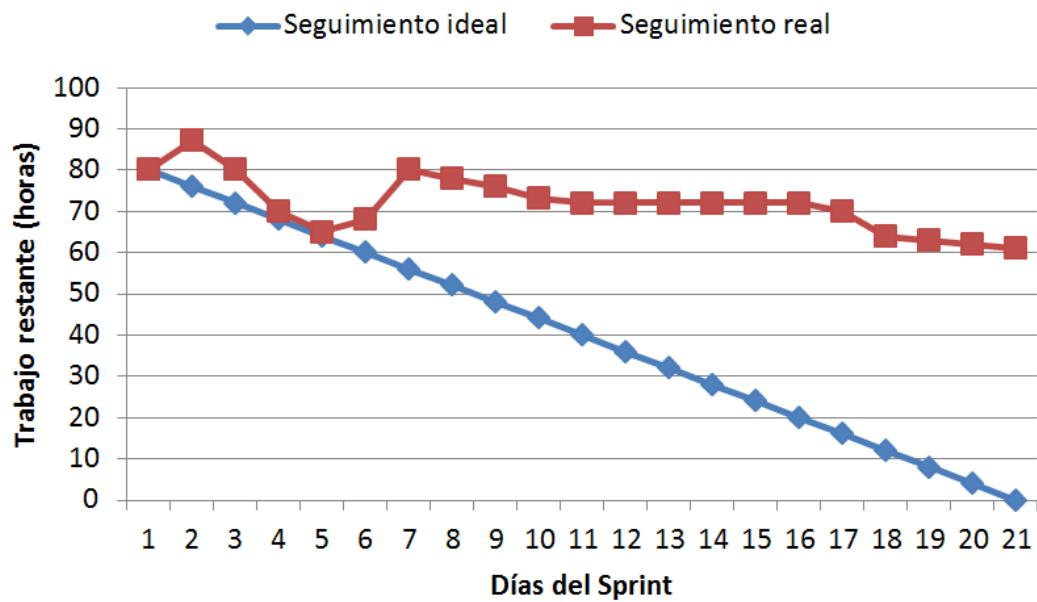


Figura 3.11: Gráfico de trabajo pendiente del primer *Sprint*

Acción realizada	Resultado esperado	Resultado obtenido
Ordenar mediante la primera columna	Resultados ordenados por la primera columna	Resultados ordenados por la primera columna
Ordenar mediante la última columna	Resultados ordenados por la última columna	Resultados ordenados por la última columna
Ordenar mediante la columna del medio	Resultados ordenados por la columna del medio	Resultados ordenados por la columna del medio
Ir de la primera página a la segunda	Mostrar la segunda página	Mostrar la segunda página
Ir de la segunda página a la primera	Mostrar la primera página	Mostrar la primera página
Ir de la penúltima página a la última	Mostrar la última página	Mostrar la última página
Ir de la última página a la penúltima	Mostrar la penúltima página	Mostrar la penúltima página
Ir de una página al azar a la siguiente	Mostrar la siguiente página	Mostrar la siguiente página
Seleccionar mostrar detalles del primer <i>Contig</i>	Mostrar detalles del primer <i>Contig</i>	Mostrar detalles del primer <i>Contig</i>
Seleccionar mostrar detalles del último <i>Contig</i>	Mostrar detalles del último <i>Contig</i>	Mostrar detalles del último <i>Contig</i>
Seleccionar mostrar detalles de un <i>Contig</i> al azar	Mostrar detalles de ese <i>Contig</i>	Mostrar detalles de ese <i>Contig</i>

Cuadro 3.15: Pruebas de valores frontera del primer *Sprint*

3.5.1.6. Cambios en los requisitos

El equipo de desarrollo estaba a punto de finalizar el primer *Sprint*, cuando el Propietario del Producto contacta con el equipo de desarrollo para comunicarle cambios en los requisitos. Al estar al final del *Sprint*, se tomaron nota de ellos y se pospuso el análisis de los mismos para el comienzo de la siguiente iteración.

3.5.2. Sprint 2

En este segundo *Sprint* se intentará terminar el trabajo pendiente del anterior, además de implementar una operación de búsqueda en las dos tablas principales.

3.5.2.1. Cambios en los requisitos

Al ver el resultado del *Sprint* previo, al cliente le pareció que la navegación por las tablas podía ser más eficiente, por lo que decidió que quería poder realizar búsquedas por todos los campos en todas las tablas. Adicionalmente, al seleccionar uno de los campos en la tabla *Unigenes*, quería realizar una búsqueda de todos los *Contigs* con esa descripción. Además, para los propósitos de la web, no necesitaba la pestaña “UNIGENES” que ya estaba implementada, por lo que simplemente se eliminará.

Las tareas a realizar en este *Sprint* son las atrasadas del anterior, además de implementar una búsqueda por los campos de las tablas.

3.5.2.2. Análisis

En esta sección se analizarán los casos de uso a implementar en el presente *Sprint*.

El diagrama de casos de uso de búsquedas puede verse en la Figura 3.12. En ella se puede observar al único actor que interactúa con la herramienta. Los dos casos de uso aquí presentes son respectivas búsquedas por las dos tablas principales de la base de datos.

■ Caso de uso: Búsqueda de *Contig* por campo.

Cuando el usuario del sistema necesita buscar un determinado *Contig* en la base de datos.

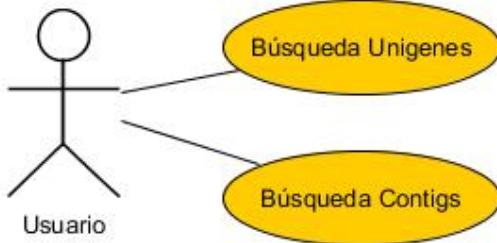


Figura 3.12: Diagrama de casos de uso de las búsquedas en las tablas

- Precondiciones:
 - Ninguna.
- El escenario del caso de uso se puede ver en el Cuadro 3.16.

Paso	Actor	Acción
1	Usuario	Selecciona el campo sobre el que desea buscar e introduce el texto de búsqueda
2	Aplicación	Muestra los resultados coincidentes con la búsqueda

Cuadro 3.16: Caso de uso “Búsqueda de *Contigs* por campo”

- Postcondiciones:
 - Ninguna.
- Posibles errores:
 - Error en la conexión a la base de datos.
- Las vistas usadas se muestran en el Cuadro 3.17.

Vista	Descripción
Formulario de búsqueda	Muestra los campos necesarios para realizar una búsqueda
Tabla <i>Contigs</i>	Muestra los resultados de la búsqueda

Cuadro 3.17: Vistas usadas por el caso de uso “Búsqueda de *Contigs* por campo”

■ Caso de uso: Búsqueda de *Unigenes* por campo.

Cuando el usuario del sistema necesita buscar un determinado *Unigen* en la base de datos.

- Precondiciones:
 - Ninguna.
- El escenario del caso de uso se puede ver en el Cuadro 3.18.

Paso	Actor	Acción
1	Usuario	Selecciona el campo sobre el que desea buscar e introduce el texto de búsqueda
2	Aplicación	Muestra los resultados coincidentes con la búsqueda

Cuadro 3.18: Caso de uso “Búsqueda de *Unigenes* por campo”

- Postcondiciones:
 - Ninguna.
- Posibles errores:
 - Error en la conexión a la base de datos.
- Las vistas usadas se muestran en el Cuadro 3.19.

Vista	Descripción
Formulario de búsqueda	Muestra los campos necesarios para realizar una búsqueda
Tabla <i>Unigenes</i>	Muestra los resultados de la búsqueda

Cuadro 3.19: Vistas usadas por el caso de uso “Búsqueda de *Unigenes* por campo”

Además de las búsquedas, se requerirá de la integración de la herramienta BLAST y Jalview, ya analizadas en el *Sprint* anterior, en las secciones 3.5.1.2 y 3.5.1.3.

3.5.2.3. Diseño

El diagrama de clases, por el momento, es el mismo que el realizado en el primer *Sprint*, por lo que puede consultarse en la Figura 3.8 de la página 40. Esto es debido a que tratan sobre las mismas clases principales, y lo único que resta por analizar en este *Sprint* son las dos búsquedas.

3.5.2.4. Implementación

Se añadieron los elementos HTML necesarios para que el usuario pudiera introducir los parámetros de la búsqueda, los cuales son enviados mediante el sistema de URLs de Django al método correspondiente en la vista (recordar que en Django al controlador se le llama vista). Este método parsea los parámetros y realiza la búsqueda, pasándole el resultado a Django-tables2 para que formatee las tablas. Esta secuencia puede verse en la Figura 3.13. Como hay que realizar búsquedas en dos tablas distintas se implementaron dos métodos diferentes. Con respecto a la pestaña “UNIGENES” que el cliente ya no deseaba, simplemente se eliminó.

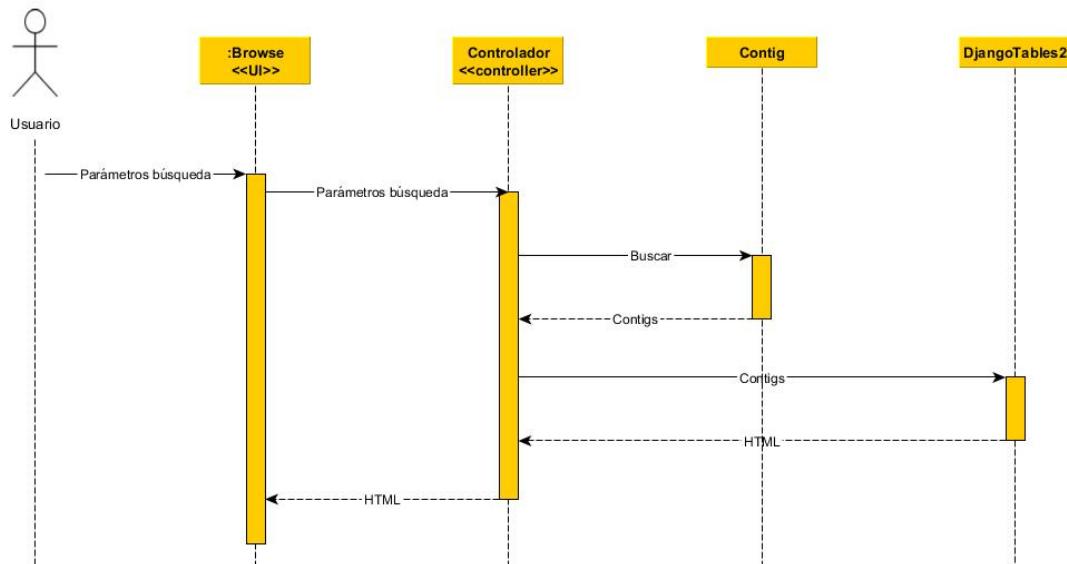


Figura 3.13: Diagrama de secuencia de la búsqueda de *Contigs*

En cuanto a BLAST, trabaja sobre una base de datos local la cual se importó desde el servidor de CHROMEVALOA. Se creó el formulario HTML que, una vez rellenado llama al método correspondiente en la vista (nuevamente, la vista es lo que correspondería al controlador). Éste parsea los datos y hace un exhaustivo control de errores. En el caso de que no se haya subido un fichero si no una secuencia, crea un documento que pasará al BLAST de línea de comandos correspondiente en función del modo elegido. Para llevar un registro de las llamadas, crea una carpeta con la IP del cliente en la que guarda la secuencia

problema y un XML con el resultado obtenido (se podrían haber añadido más datos como fecha o tiempo computacional pero no se le vio utilidad por ahora). Ese fichero XML con el resultado se parsea mediante una clase del NCBI y se envía al motor de plantillas de Django que se encargará de formatear y mostrar los datos relevantes.

Sobre Jalview, se implementaron los pasos especificados en la fase de Diseño del primer *Sprint*. Cuando se pulsa la opción de obtener el fichero de alineamiento en la pantalla de detalles de un *Contig*, se obtienen las secuencias de los *Reads* de los *Clusters* que lo forman. Éstas se guardan en un fichero en un formato específico que se pasa a la herramienta Clustal Omega como entrada. La salida de ésta es un fichero de alineamiento que se descarga. En el caso de querer además visualizarlo mediante la herramienta Jalview, se añadió un botón que al pulsarlo se descarga la herramienta de visualización en local.

La Figura 3.14 muestra el seguimiento de *Scrum* realizado en Trello. La imagen a) muestra la lista de tareas del producto. Destacar que contiene tanto las nuevas funcionalidades añadidas por el cliente, como las que no se pudieron realizar en el *Sprint* anterior. La imagen b) muestra la lista de tareas que se espera realizar en la iteración actual, que al igual que en el *Sprint* anterior, son todas las del *Product backlog*. La imagen c) muestra el estado de las listas al finalizar el *Sprint*. En esta ocasión sí que se ha completado la totalidad de las funcionalidades previstas, de modo que el gráfico de trabajo pendiente del primer *Sprint* es el mostrado en la Figura 3.15.

3.5.2.5. Pruebas

Las búsquedas se realizan con la ayuda de Django. Éste crea una API que abstrae de la base de datos, la cual permite una serie de operaciones, entre ellas búsquedas. Es por ésto que no es necesario comprobar el algoritmo específico de búsqueda (Django ya lo ha hecho) sino la implementación concreta, parseado de parámetros, formateado de salida y errores en la entrada. La forma más eficiente y real de realizar estas comprobaciones es sobre el resultado final, en la que se comprueba todo esto a la vez, es decir, sobre la propia interfaz web. Herramientas como Selenium permiten automatizar este proceso de prueba en la interfaz de usuario mediante la especificación de acciones, pero al ser una funcionalidad relativamente sencilla se decidió probarlo de forma manual.



Figura 3.14: Capturas de la herramienta Trello del segundo *Sprint*

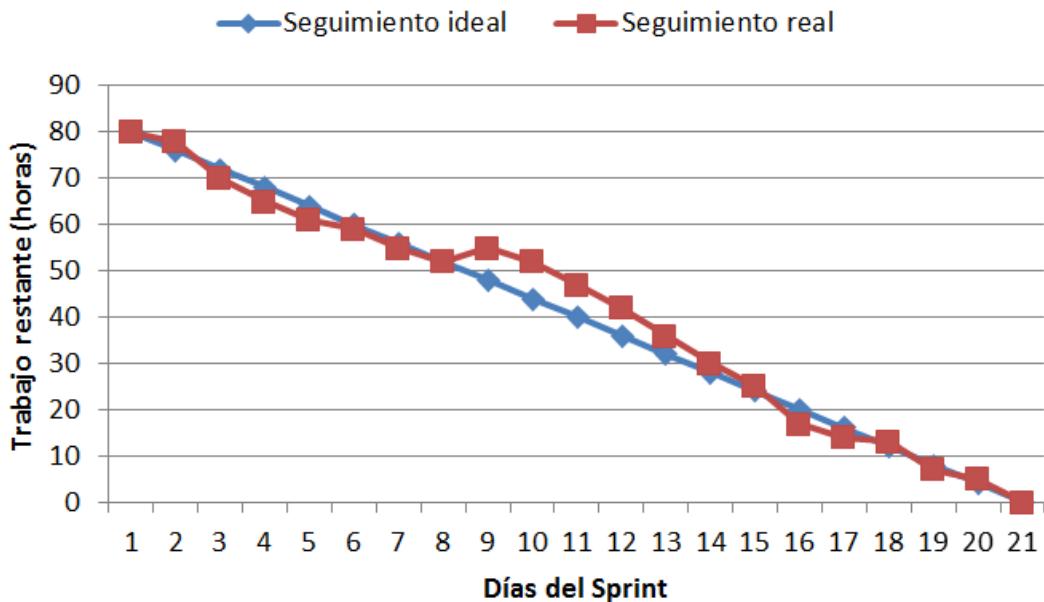


Figura 3.15: Diagrama de trabajo pendiente del segundo *Sprint*

Se han realizado, al igual que en la iteración anterior, pruebas de valores límite. Se han comprobado salidas deseadas frente a salidas obtenidas con todos los diferentes campos de búsqueda. No se ha plasmado en la memoria las salidas específicas debido a que éstas son una serie de instancias demasiado grande.

Así, para todos los campos se han realizado las pruebas mostradas en el Cuadro 3.20 sobre las dos pestañas (la primera sobre instancias de *Contigs* y la segunda sobre *Unigenes*).

Con respecto a la herramienta BLAST, se debe probar su correcta integración, ya que el funcionamiento de la misma lo probaron los autores. Ésta es una de las pocas cosas que funcionaban sin ningún fallo conocido en la web antigua, por lo que se realizaron consultas en ambas webs y se compararon los resultados. Los resultados de las consultas a las dos webs fue el mismo. Así, se asume que la integración está correctamente realizada. Éstas pruebas se realizaron para secuencias de diferentes tipos, tamaños y procedencias.

Además, con el objetivo de una completa comprobación de errores, se realizaron las pruebas mostradas en el Cuadro 3.21, comprobando los parámetros que el usuario puede modificar.

Con respecto a la generación del fichero de alineamiento, la herramienta Clus-

Acción realizada	Resultado esperado	Resultado obtenido
Búsqueda vacía	Mensaje de entrada vacía	Mensaje de entrada vacía
Búsqueda con caracteres no válidos	Salida vacía	Salida vacía
Búsqueda válida inexistente	Salida vacía	Salida vacía
Búsqueda válida existente	Instancias correctas	Instancias correctas

Cuadro 3.20: Pruebas de valores frontera del segundo *Sprint*

Acción realizada	Resultado esperado	Resultado obtenido
Consulta de nucleótidos con caracteres inválidos	Mensaje de error en la consulta	Mensaje de error en la consulta
Consulta de proteínas con caracteres inválidos	Mensaje de error en la consulta	Mensaje de error en la consulta
Consulta con un e-valor inválido	Mensaje de error en el e-valor	Mensaje de error en el e-valor
Consulta con un e-valor en blanco	Mensaje de error en el e-valor	Mensaje de error en el e-valor
Consulta con un número máximo inválido	Mensaje de número máximo inválido	Mensaje de número máximo inválido
Consulta con un número máximo en blanco	Mensaje de número máximo inválido	Mensaje de número máximo inválido

Cuadro 3.21: Pruebas realizadas sobre BLAST

tal Omega no tiene errores conocidos que afecten a este proyecto. De este modo, solamente resulta necesario probar la generación del fichero que recibe como parámetro. Al ser éste simplemente los resultados de una consulta a la base de datos, se generaron ficheros con diferentes entradas, comparando el resultado esperado con una consulta SQL. El resultado fue el esperado y se obtenían satisfactoriamente todos los Reads de los *Clusters* que forman el *Contig*.

Para probar el correcto formato del fichero de salida de Clustal, se introdujo en la herramienta Jalview. De este modo se probó que esta última herramienta funcionaba y aceptaba el fichero generado previamente. De igual modo que Clustal, Jalview no necesita ser probado debido a que no tiene errores conocidos que afecten al proyecto.

3.5.3. Sprint 3

El presente es el último *Sprint* de la parte de la herramienta web. En él se abordará la gestión de un menú de administrador.

3.5.3.1. Captura de requisitos

El cliente necesitaba más funcionalidades de las que inicialmente pensaba. Es por esto que a la lista de requisitos se añadió un menú de administrador con las siguientes funcionalidades:

- Gestión de usuarios:
 - Búsqueda por nombre y correo.
 - Altas de usuario.
 - Bajas de usuario.
 - Modificaciones de usuario.

 - Gestión de grupos:
 - Búsqueda por nombre.
 - Altas de grupo.
 - Bajas de grupo.
 - Modificaciones de grupo.
-

■ Gestión de *Reads*:

- Búsqueda por los campos ID y SEQ.
- Altas de *Read*.
- Bajas de *Read*.
- Modificaciones de *Read*.

■ Gestión de *Clusters*:

- Búsqueda por los campos ID y SEQ.
- Altas de *Cluster*.
- Bajas de *Cluster*.
- Modificaciones de *Cluster*.

■ Gestión de *Contigs*:

- Búsqueda por los campos ID, SEQ y descripción.
- Altas de *Contig*.
- Bajas de *Contig*.
- Modificaciones de *Contig*.

Destacar que el cliente quería una completa gestión de usuarios con sus permisos y grupos. Ésto es debido a que quiere poder restringir ciertas acciones a ciertos usuarios, ya que no todos los integrantes del grupo CHROMEVOL tienen el mismo estatus. Además, de esta forma se podría también dar permisos limitados a usuarios externos del grupo.

A continuación, manifestó su deseo de cambiar la estructura de la base de datos. Quería una estructura parecida a un *Data Warehouse* (DW), para almacenar grandes cantidades de datos y poder acceder a ellos de forma eficiente para consultarlos y analizarlos. Ésto dotará al sistema de una mayor escalabilidad, algo realmente deseable. También le resultaba muy tedioso al añadir un *Contig* tener que modificar varias tablas (*Contig*, *ContigSEQ* y *Expression*). Debido a esto, deseaba fusionar la tabla *Contig* con las que incluían datos sobre ella, lo que además proporcionaría un acceso más eficiente a las filas completas al no tener que recurrir a operaciones de *join*. Dicha eficiencia es algo que destacó especialmente

debido a la lentitud de las consultas de la antigua web, que llegaban a tardar hasta 5 segundos. Además, se negó a cambiar la tabla *Unigenes* por otra tabla con un identificador de *Contig* y un atributo, a pesar de tener valores duplicados. Expuso que quería tener esa información aparte, debido a que los *Unigenes* serían sustituidos en un futuro no muy lejano. Como ya se mencionó anteriormente (ver Sección 3.1.2), esta tabla se obtuvo a partir de la BD UniGene del NCBI. A día de hoy, se sabe que esta aproximación tiene numerosas desventajas [24], además de la controversia existente en cuanto a cómo escoger la secuencia representativa. Por todo ello, esta tecnología quedará obsoleta muy pronto.

3.5.3.2. Análisis

En esta sección se abordará el análisis de las funcionalidades a realizar. En todas ellas el agente que interactúa con el sistema es un investigador del grupo CHROMEVOL con los permisos necesarios. Éste será un usuario que podrá editar la base de datos, siendo referido como el “administrador”.

Los casos de uso contemplados en este *Sprint* son los siguientes:

- **Casos de uso: Gestión de usuarios.**

El diagrama de casos de uso de Gestión de usuarios puede verse en la Figura 3.16. En ésta se puede observar que el actor es ahora el administrador. Para poder acceder a estas funcionalidades es necesario que se identifique, momento en el cual se le ofrecerán las posibilidades especificadas en los requisitos. Puede crear un usuario, buscarlo, modificarlo o eliminarlo. Para estas dos últimas funcionalidades es necesario que previamente busque al usuario que quiera editar, de cualquiera de las formas presentadas.

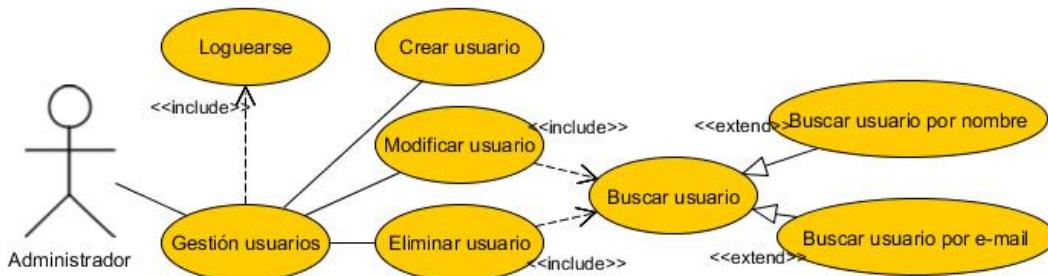


Figura 3.16: Diagrama de casos de uso “Gestión de usuarios”

- **Caso de uso: Crear usuario.**

Cuando un administrador quiere añadir un nuevo usuario a la base de datos.

- Precondiciones:
 - ◊ El administrador está autenticado en el sistema.
 - ◊ El usuario no existe en la base de datos.
- El escenario del caso de uso se puede ver en el Cuadro 3.22.

Paso	Actor	Acción
1	Administrador	Selecciona la opción de “añadir usuario”
2	Aplicación	Muestra el formulario de creación de un nuevo usuario
3	Administrador	Cubre el formulario y pulsa la opción de confirmación
4	Aplicación	Muestra un resumen de los datos del usuario y un formulario para llenar datos opcionales (permisos, grupo, e-mail...)
5	Administrador	Pulsa la opción de confirmación
6	Aplicación	Muestra una pantalla con todos los usuarios almacenados en la base de datos habiendo dado de alta el seleccionado

Cuadro 3.22: Caso de uso “Crear usuario”

- Postcondiciones:
 - ◊ El usuario se crea en la base de datos.
- Posibles errores:
 - ◊ El usuario ya existe en la base de datos.
 - ◊ Error en la conexión con la base de datos.
- Las vistas usadas se muestran en la el Cuadro 3.23.

- **Caso de uso: Modificar usuario.**

Cuando un administrador quiere modificar los datos relacionados con un usuario.

- Precondiciones:
 - ◊ El administrador está autenticado en el sistema.

Vista	Descripción
Formulario básico de usuario	Contiene los campos de nombre y contraseña
Formulario de usuario completo	Contiene un formulario con todos los datos del usuario seleccionado
Vista de usuarios	Contiene una vista de los usuarios dados de alta

Cuadro 3.23: Vistas usadas en el caso de uso “Crear usuario”

- ◊ El usuario existe en la base de datos.
- El escenario del caso de uso se puede ver en el Cuadro 3.24.

Paso	Actor	Acción
1	Administrador	Selecciona la opción de “modificar usuario”
2	Aplicación	Muestra un listado de los usuarios en la base de datos
3	Administrador	Selecciona el usuario que quiere modificar
4	Aplicación	Muestra un resumen de los datos del usuario
5	Administrador	Modifica los datos pertinentes y pulsa la opción de confirmación
6	Aplicación	Muestra un listado de todos los usuarios en la base de datos habiendo modificado el seleccionado

Cuadro 3.24: Caso de uso “Modificar usuario”

- Postcondiciones:
 - ◊ Los datos del usuario se modifican en la base de datos.
- Posibles errores:
 - ◊ Error en la conexión a la base de datos.
 - ◊ El usuario no existe en la base de datos.
- Las vistas usadas se muestran en el Cuadro 3.25.
- **Caso de uso: Eliminar usuario.**

Cuando un administrador quiere eliminar a un usuario de la base de datos.

 - Precondiciones:

Vista	Descripción
Vista de usuarios	Contiene una vista de los usuarios dados de alta
Formulario de usuario completo	Contiene un formulario con todos los datos del usuario seleccionado

Cuadro 3.25: Vistas usadas en el caso de uso “Modificar usuario”

- ◊ El administrador está autenticado en el sistema.
- ◊ El usuario existe en la base de datos.
- El escenario del caso de uso se puede ver en el Cuadro 3.26.

Paso	Actor	Acción
1	Administrador	Selecciona la opción de gestión de usuarios
2	Aplicación	Muestra un listado de los usuarios de la base de datos
3	Administrador	Selecciona el usuario que quiere eliminar
4	Aplicación	Muestra un resumen de los datos del usuario
5	Administrador	Selecciona la opción de “eliminar usuario”
6	Aplicación	Muestra una pantalla de confirmación
7	Administrador	Selecciona la opción de “confirmar”
8	Aplicación	Muestra un listado de los usuarios de la base de datos habiendo borrado el seleccionado

Cuadro 3.26: Caso de uso “Eliminar usuario”

- Postcondiciones:
 - ◊ El usuario es eliminado de la base de datos.
- Posibles errores:
 - ◊ Error en la conexión a la base de datos.
 - ◊ El usuario no existe en la base de datos.
- Las vistas usadas se muestran en el Cuadro 3.27.
- **Caso de uso: Buscar usuario por nombre.**

Cuando un administrador quiere buscar a un usuario por su nombre.

 - Precondiciones:
 - ◊ Ninguna.

Vista	Descripción
Vista de usuarios	Contiene una vista de los usuarios dados de alta
Formulario de usuario completo	Contiene un formulario con todos los datos del usuario seleccionado
Pantalla de confirmación de eliminar usuario	Pregunta al administrador si quiere eliminar el usuario seleccionado

Cuadro 3.27: Vistas usadas en el caso de uso “Eliminar usuario”

- El escenario del caso de uso se puede ver en la Tabla 3.28.

Paso	Actor	Acción
1	Administrador	Selecciona la opción de gestión de usuarios
2	Aplicación	Muestra un listado de los usuarios de la base de datos
3	Administrador	Introduce el nombre del usuario que desea buscar en el espacio de búsqueda
4	Aplicación	Busca y muestra los usuarios con ese nombre

Cuadro 3.28: Caso de uso “Buscar usuario por nombre”

- Postcondiciones:
 - ◊ Ninguna.
- Posibles errores:
 - ◊ Error en la conexión a la base de datos.
- Las vistas usadas se muestran en el Cuadro 3.29.

Vista	Descripción
Vista de usuarios	Contiene una vista de los usuarios dados de alta

Cuadro 3.29: Vistas usadas en el caso de uso “Buscar usuario por nombre”

- **Caso de uso: Buscar usuario por e-mail.**

Cuando un administrador quiere buscar a un usuario por su correo electrónico.

- Precondiciones:

- ◊ Ninguna.
- El escenario del caso de uso se puede ver en el Cuadro 3.30.

Paso	Actor	Acción
1	Administrador	Selecciona la opción de gestión de usuarios
2	Aplicación	Muestra un listado de los usuarios de la base de datos
3	Administrador	Introduce el correo del usuario que desea buscar en el espacio de búsqueda
4	Aplicación	Busca y muestra los usuarios con ese e-mail

Cuadro 3.30: Caso de uso “Buscar usuario por e-mail”

- Postcondiciones:
 - ◊ Ninguna.
- Posibles errores:
 - ◊ Error en la conexión a la base de datos.
- Las vistas usadas se muestran en la Tabla 3.31.

Vista	Descripción
Vista de usuarios	Contiene una vista de los usuarios dados de alta

Cuadro 3.31: Vistas usadas en el caso de uso “Buscar usuario por e-mail”

■ Casos de uso: Gestión de grupos.

El diagrama de casos de uso de Gestión de grupos puede verse en la Figura 3.17. En ésta se puede observar que el actor es cualquier administrador con permisos. Para poder acceder a estas funcionalidades es necesario que se identifique, momento en el cual se le ofrecerán las posibilidades especificadas en los requisitos. Puede crear un grupo, buscarno, modificarlo o eliminarlo. Para estas dos últimas funcionalidades es necesario que previamente busque al grupo que quiera editar.

• Caso de uso: Crear grupo.

Cuando un administrador quiere añadir un nuevo grupo a la base de datos

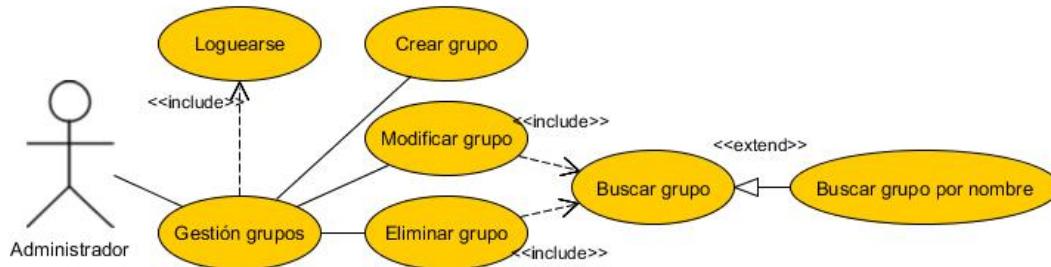


Figura 3.17: Diagrama de casos de uso “Gestión de grupos”

- Precondiciones:
 - ◊ El administrador está autenticado en el sistema.
 - ◊ El grupo no existe en la base de datos.
- El escenario del caso de uso se puede ver en el Cuadro 3.32.

Paso	Actor	Acción
1	Administrador	Selecciona la opción de “añadir grupo”
2	Aplicación	Muestra el formulario de creación de un nuevo grupo
3	Administrador	Cubre el nombre del grupo, selecciona los permisos deseados y pulsa la opción de confirmación
6	Aplicación	Muestra una pantalla con todos los grupos almacenados en la base de datos habiendo dado de alta el seleccionado

Cuadro 3.32: Caso de uso “Crear grupo”

- Postcondiciones:
 - ◊ El grupo se crea en la base de datos.
- Posibles errores:
 - ◊ El grupo ya existe en la base de datos.
 - ◊ Error en la conexión con la base de datos.
- Las vistas usadas se muestran en el Cuadro 3.33.
- **Caso de uso: Modificar grupo.**
Cuando un administrador quiere modificar los datos relacionados con un grupo.

Vista	Descripción
Formulario de grupo	Contiene los campos de nombre y permisos
Vista de grupos	Contiene una vista de los grupos dados de alta

Cuadro 3.33: Vistas usadas en el caso de uso “Crear grupo”

- Precondiciones:
 - ◊ El administrador está autenticado en el sistema.
 - ◊ El grupo existe en la base de datos.
- El escenario del caso de uso se puede ver en el Cuadro 3.34.

Paso	Actor	Acción
1	Administrador	Selecciona la opción de “modificar grupo”
2	Aplicación	Muestra un listado de los grupo en la base de datos
3	Administrador	Selecciona el grupo que quiere modificar
4	Aplicación	Muestra un resumen de los datos del grupo
5	Administrador	Modifica los datos pertinentes y pulsa la opción de confirmación
6	Aplicación	Muestra un listado de todos los grupo en la base de datos habiendo modificado el seleccionado

Cuadro 3.34: Caso de uso “Modificar grupo”

- Postcondiciones:
 - ◊ Los datos del grupo se modifican en la base de datos.
- Posibles errores:
 - ◊ Error en la conexión a la base de datos.
 - ◊ El grupo no existe en la base de datos.
- Las vistas usadas se muestran en el Cuadro 3.35.
- **Caso de uso: Eliminar grupo.**
 Cuando un administrador quiere eliminar un grupo de la base de datos.
 - Precondiciones:
 - ◊ El administrador está autenticado en el sistema.

Vista	Descripción
Vista de grupos	Contiene una vista de los grupos dados de alta
Formulario de grupo	Contiene un formulario con todos los datos del grupo seleccionado

Cuadro 3.35: Vistas usadas en el caso de uso “Modificar grupo”

- ◊ El grupo existe en la base de datos.
- El escenario del caso de uso se puede ver en el Cuadro 3.36.

Paso	Actor	Acción
1	Administrador	Selecciona la opción de gestión de grupos
2	Aplicación	Muestra un listado de los grupos de la base de datos
3	Administrador	Selecciona el grupo que quiere eliminar
4	Aplicación	Muestra un resumen de los datos del grupo
5	Administrador	Selecciona la opción de “eliminar grupo”
6	Aplicación	Muestra una pantalla de confirmación
7	Administrador	Selecciona la opción de “confirmar”
8	Aplicación	Muestra un listado de los grupo de la base de datos habiendo borrado el seleccionado

Cuadro 3.36: Caso de uso “Eliminar grupo”

- Postcondiciones:
 - ◊ El grupo es eliminado de la base de datos.
- Posibles errores:
 - ◊ Error en la conexión a la base de datos.
 - ◊ El grupo no existe en la base de datos.
- Las vistas usadas se muestran en el Cuadro 3.37.
- **Caso de uso: Buscar grupo por nombre.**

Cuando un administrador quiere buscar a un grupo por su nombre.

 - Precondiciones:
 - ◊ Ninguna.

Vista	Descripción
Vista de grupo	Contiene una vista de los grupo dados de alta
Formulario de grupo	Contiene un formulario con todos los datos del grupo seleccionado
Pantalla de confirmación de eliminar grupo	Pregunta al administrador si quiere eliminar el grupo seleccionado

Cuadro 3.37: Vistas usadas en el caso de uso “Eliminar grupo”

- El escenario del caso de uso se puede ver en la Tabla 3.38.

Paso	Actor	Acción
1	Administrador	Selecciona la opción de gestión de grupos
2	Aplicación	Muestra un listado de los grupos de la base de datos
3	Administrador	Introduce el nombre del grupo que desea buscar en el espacio de búsqueda
4	Aplicación	Busca y muestra los grupos con ese nombre

Cuadro 3.38: Caso de uso “Buscar grupo por nombre”

- Postcondiciones:
 - ◊ Ninguna.
- Posibles errores:
 - ◊ Error en la conexión a la base de datos.
- Las vistas usadas se muestran en el Cuadro 3.39.

Vista	Descripción
Vista de grupos	Contiene una vista de los grupo dados de alta

Cuadro 3.39: Vistas usadas en el caso de uso “Buscar grupo por nombre”

▪ Casos de uso: Gestión de *Reads*

El diagrama de casos de uso de Gestión de *Reads* puede verse en la Figura 3.18. En ésta se puede observar que el único actor contemplado es el administrador. Para poder gestionar los *Reads* es necesario que se identifique, momento en el cual se le ofrecerán las posibilidades especificadas en los requisitos. Puede crear, buscar, modificar o eliminar *Reads*. Para estas dos últimas funcionalidades es necesario que previamente busque la entidad que quiera editar, de cualquiera de las formas presentadas.

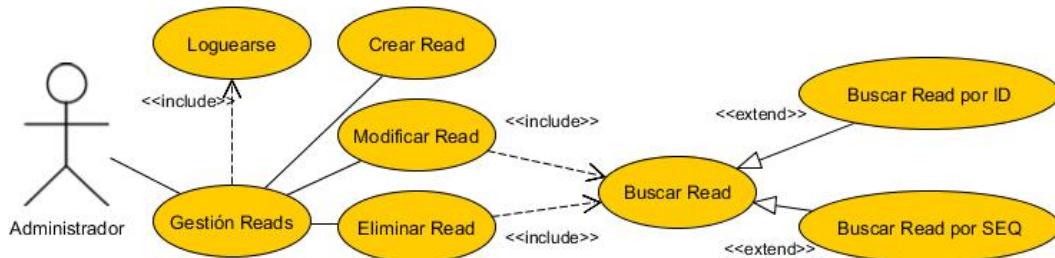


Figura 3.18: Diagrama de casos de uso “Gestión de *Reads*”

- **Caso de uso: Crear *Read*.**

Cuando el administrador quiere añadir un nuevo *Read* a la base de datos.

- Precondiciones:
 - ◊ El administrador está autenticado en el sistema.
 - ◊ El usuario no existe en la base de datos.
- El escenario del caso de uso se puede ver en el Cuadro 3.40.

Paso	Actor	Acción
1	Administrador	Selecciona la opción de “añadir <i>Read</i> ”
2	Aplicación	Muestra el formulario de creación de un nuevo <i>Read</i>
3	Administrador	Cubre el formulario y pulsa la opción de confirmación
4	Aplicación	Muestra una pantalla con todos los <i>Reads</i> almacenados en la base de datos habiendo dado de alta el seleccionado

Cuadro 3.40: Caso de uso “Crear *Read*”

- Postcondiciones:
 - ◊ El *Read* se crea en la base de datos.
- Posibles errores:
 - ◊ El *Read* ya existe en la base de datos.
 - ◊ Error en la conexión con la base de datos.
- Las vistas usadas se muestran en el Cuadro 3.41.

Vista	Descripción
Vista de <i>Reads</i>	Contiene una lista de los <i>Reads</i> de la base de datos
Formulario de <i>Read</i>	Contiene un formulario con todos los datos del <i>Read</i>

Cuadro 3.41: Vistas usadas en el caso de uso “Crear *Read*”

- **Caso de uso: Modificar *Read*.**

Cuando un administrador quiere modificar los datos relacionados con un *Read*.

- Precondiciones:
 - ◊ El administrador está autenticado en el sistema.
 - ◊ El *Read* existe en la base de datos.
- El escenario del caso de uso se puede ver en el Cuadro 3.42.

Paso	Actor	Acción
1	Administrador	Selecciona la opción de “modificar <i>Read</i> ”
2	Aplicación	Muestra un listado de los <i>Reads</i> en la base de datos
3	Administrador	Selecciona el <i>Read</i> que quiere modificar
4	Aplicación	Muestra un resumen de los datos del <i>Read</i>
5	Administrador	Modifica los datos pertinentes y pulsa la opción de confirmación
6	Aplicación	Muestra un listado de todos los <i>Reads</i> en la base de datos habiendo modificado el seleccionado

Cuadro 3.42: Caso de uso “Modificar *Read*”

- Postcondiciones:
 - ◊ Los datos del *Read* se modifican en la base de datos.
- Posibles errores:
 - ◊ Error en la conexión a la base de datos.
 - ◊ El *Read* no existe en la base de datos.
- Las vistas usadas se muestran en el Cuadro 3.43.

Vista	Descripción
Vista de <i>Reads</i>	Contiene una vista de los <i>Reads</i> dados de alta
Formulario de <i>Read</i> completo	Contiene un formulario con todos los datos del <i>Read</i> seleccionado

Cuadro 3.43: Vistas usadas en el caso de uso “Modificar *Read*”

• **Caso de uso: Eliminar *Read*.**

Cuando un administrador quiere eliminar a un *Read* de la base de datos.

- Precondiciones:
 - ◊ El administrador está autenticado en el sistema.
 - ◊ El *Read* existe en la base de datos.
- El escenario del caso de uso se puede ver en el Cuadro 3.44.
- Postcondiciones:
 - ◊ El *Read* es eliminado de la base de datos.
- Posibles errores:
 - ◊ Error en la conexión a la base de datos.
 - ◊ El *Read* no existe en la base de datos.
- Las vistas usadas se muestran en el Cuadro 3.45.

• **Caso de uso: Buscar *Read* por ID.**

Cuando un administrador quiere buscar a un *Read* por su ID.

- Precondiciones:
 - ◊ Ninguna.
- El escenario del caso de uso se puede ver en el Cuadro 3.46.

Paso	Actor	Acción
1	Administrador	Selecciona la opción de gestión de <i>Reads</i>
2	Aplicación	Muestra un listado de los <i>Reads</i> de la base de datos
3	Administrador	Selecciona el <i>Read</i> que quiere eliminar
4	Aplicación	Muestra un resumen de los datos del <i>Read</i>
5	Administrador	Selecciona la opción de “eliminar <i>Read</i> ”
6	Aplicación	Muestra una pantalla de confirmación
7	Administrador	Selecciona la opción de “confirmar”
8	Aplicación	Muestra un listado de los <i>Reads</i> de la base de datos habiendo borrado el seleccionado

Cuadro 3.44: Caso de uso “Eliminar *Read*”

Vista	Descripción
Vista de <i>Reads</i>	Contiene una vista de los <i>Reads</i> dados de alta
Formulario de <i>Read</i> completo	Contiene un formulario con todos los datos del <i>Read</i> seleccionado
Pantalla de confirmación de eliminar <i>Read</i>	Pregunta al administrador si quiere eliminar el <i>Read</i> seleccionado

Cuadro 3.45: Vistas usadas en el caso de uso “Eliminar *Read*”

- Postcondiciones:
 - ◊ Ninguna.
- Posibles errores:
 - ◊ Error en la conexión a la base de datos.
- Las vistas usadas se muestran en el Cuadro 3.47.
- **Caso de uso: Buscar *Read* por SEQ.**

Cuando un administrador quiere buscar a un *Read* por su SEQ.

 - Precondiciones:
 - ◊ Ninguna.
 - El escenario del caso de uso se puede ver en el Cuadro 3.48.

Paso	Actor	Acción
1	Administrador	Selecciona la opción de gestión de <i>Reads</i>
2	Aplicación	Muestra un listado de los <i>Reads</i> de la base de datos
3	Administrador	Introduce el ID del <i>Read</i> que desea buscar en el espacio de búsqueda
4	Aplicación	Busca y muestra los <i>Reads</i> con ese ID

Cuadro 3.46: Caso de uso “Buscar *Read* por ID”

Vista	Descripción
Vista de <i>Reads</i>	Contiene una vista de los <i>Reads</i> dados de alta

Cuadro 3.47: Vistas usadas en el caso de uso “Buscar *Read* por ID”

Paso	Actor	Acción
1	Administrador	Selecciona la opción de gestión de <i>Reads</i>
2	Aplicación	Muestra un listado de los <i>Reads</i> de la base de datos
3	Administrador	Introduce el campo SEQ del <i>Read</i> que desea buscar en el espacio de búsqueda
4	Aplicación	Busca y muestra los <i>Reads</i> con ese SEQ

Cuadro 3.48: Caso de uso “Buscar *Read* por SEQ”

- Postcondiciones:
 - ◊ Ninguna.
- Posibles errores:
 - ◊ Error en la conexión a la base de datos.
- Las vistas usadas se muestran en el Cuadro 3.49.

Vista	Descripción
Vista de <i>Reads</i>	Contiene una vista de los <i>Reads</i> dados de alta

Cuadro 3.49: Vistas usadas en el caso de uso “Buscar *Read* por SEQ”

- Casos de uso: Gestión de *Contigs*.

El diagrama de casos de uso de Gestión de *Contigs* puede verse en la Figura 3.19. El único actor contemplado es el administrador. Es necesario que previamente se identifique, momento en el cual se le ofrecerán las posibilidades especificadas en los requisitos. Puede crear, buscar, modificar o eliminar *Contigs*. Para estas dos últimas funcionalidades es necesario que previamente busque la entidad que quiera editar, de cualquiera de las formas presentadas.

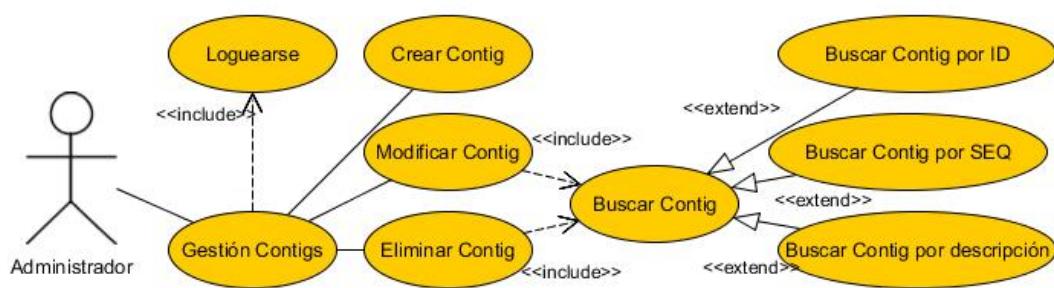


Figura 3.19: Diagrama de casos de uso “Gestión de *Contigs*”

- **Caso de uso: Crear *Contig*** Cuando el administrador quiere añadir un nuevo *Contig* a la base de datos.
 - Precondiciones:
 - ◊ El administrador está autenticado en el sistema.
 - ◊ El *Contig* no existe en la base de datos.
 - El escenario del caso de uso se puede ver en el Cuadro 3.50.

Paso	Actor	Acción
1	Administrador	Selecciona la opción de “añadir <i>Contig</i> ”
2	Aplicación	Muestra el formulario de creación de un nuevo <i>Contig</i>
3	Administrador	Cubre el formulario y pulsa la opción de confirmación
4	Aplicación	Muestra una pantalla con todos los <i>Contigs</i> almacenados en la base de datos habiendo dado de alta el seleccionado

Cuadro 3.50: Caso de uso “Crear *Contig*”

- Postcondiciones:
 - ◊ El *Contig* se crea en la base de datos.
- Posibles errores:
 - ◊ El *Contig* ya existe en la base de datos.
 - ◊ Error en la conexión con la base de datos.
- Las vistas usadas se muestran en el Cuadro 3.51.

Vista	Descripción
Vista de <i>Contigs</i>	Contiene una lista de los <i>Contigs</i> de la base de datos
Formulario de <i>Contig</i>	Contiene un formulario con todos los datos del <i>Contig</i>

Cuadro 3.51: Vistas usadas en el caso de uso “Crear *Contig*”

• **Caso de uso: Modificar *Contig*.**

Cuando un administrador quiere modificar los datos relacionados con un *Contig*.

- Precondiciones:
 - ◊ El administrador está autenticado en el sistema.
 - ◊ El *Contig* existe en la base de datos.
- El escenario del caso de uso se puede ver en el Cuadro 3.52.

Paso	Actor	Acción
1	Administrador	Selecciona la opción de “modificar <i>Contig</i> ”
2	Aplicación	Muestra un listado de los <i>Contigs</i> en la base de datos
3	Administrador	Selecciona el <i>Contig</i> que quiere modificar
4	Aplicación	Muestra un resumen de los datos del <i>Contig</i>
5	Administrador	Modifica los datos pertinentes y pulsa la opción de confirmación
6	Aplicación	Muestra un listado de todos los <i>Contigs</i> en la base de datos habiendo modificado el seleccionado

Cuadro 3.52: Caso de uso “Modificar *Contig*”

- Postcondiciones:
 - ◊ Los datos del *Contig* se modifican en la base de datos.
- Posibles errores:
 - ◊ Error en la conexión a la base de datos.
 - ◊ El *Contig* no existe en la base de datos.
- Las vistas usadas se muestran en el Cuadro 3.53.

Vista	Descripción
Vista de <i>Contigs</i>	Contiene una vista de los <i>Contigs</i> dados de alta
Formulario de <i>Contig</i> completo	Contiene un formulario con todos los datos del <i>Contig</i> seleccionado

Cuadro 3.53: Vistas usadas en el caso de uso “Modificar *Contig*”

- **Caso de uso: Eliminar *Contig*.**

Cuando un administrador quiere eliminar a un *Contig* de la base de datos.

- Precondiciones:
 - ◊ El administrador está autenticado en el sistema.
 - ◊ El *Contig* existe en la base de datos.
- El escenario del caso de uso se puede ver en el Cuadro 3.54.
- Postcondiciones:
 - ◊ El *Contig* es eliminado de la base de datos.
- Posibles errores:
 - ◊ Error en la conexión a la base de datos.
 - ◊ El *Contig* no existe en la base de datos.
- Las vistas usadas se muestran en el Cuadro 3.55.

- **Caso de uso: Buscar *Contig* por ID.**

Cuando un administrador quiere buscar a un *Contig* por su ID.

- Precondiciones:
 - ◊ Ninguna.
- El escenario del caso de uso se puede ver en el Cuadro 3.56.

Paso	Actor	Acción
1	Administrador	Selecciona la opción de gestión de <i>Contigs</i>
2	Aplicación	Muestra un listado de los <i>Contigs</i> de la base de datos
3	Administrador	Selecciona el <i>Contig</i> que quiere eliminar
4	Aplicación	Muestra un resumen de los datos del <i>Contig</i>
5	Administrador	Selecciona la opción de “eliminar <i>Contig</i> ”
6	Aplicación	Muestra una pantalla de confirmación
7	Administrador	Selecciona la opción de “confirmar”
8	Aplicación	Muestra un listado de los <i>Contigs</i> de la base de datos habiendo borrado el seleccionado

Cuadro 3.54: Caso de uso “Eliminar *Contig*”

Vista	Descripción
Vista de <i>Contigs</i>	Contiene una vista de los <i>Contigs</i> dados de alta
Formulario de <i>Contig</i> completo	Contiene un formulario con todos los datos del <i>Contig</i> seleccionado
Pantalla de confirmación de eliminar <i>Contig</i>	Pregunta al administrador si quiere eliminar el <i>Contig</i> seleccionado

Cuadro 3.55: Vistas usadas en el caso de uso “Eliminar *Contig*”

- Postcondiciones:
 - ◊ Ninguna.
- Posibles errores:
 - ◊ Error en la conexión a la base de datos.
 - Las vistas usadas se muestran en el Cuadro 3.57.
- **Caso de uso: Buscar *Contig* por descripción.**

Cuando un administrador quiere buscar a un *Contig* por su descripción.

 - Precondiciones:
 - ◊ Ninguna.
 - El escenario del caso de uso se puede ver en el Cuadro 3.58.

Paso	Actor	Acción
1	Administrador	Selecciona la opción de gestión de <i>Contigs</i>
2	Aplicación	Muestra un listado de los <i>Contigs</i> de la base de datos
3	Administrador	Introduce el ID del <i>Contig</i> que desea buscar en el espacio de búsqueda
4	Aplicación	Busca y muestra los <i>Contigs</i> con ese SEQ

Cuadro 3.56: Caso de uso “Buscar *Contig* por ID”

Vista	Descripción
Vista de <i>Contigs</i>	Contiene una vista de los <i>Contigs</i> dados de alta

Cuadro 3.57: Vistas usadas en el caso de uso “Buscar *Contig* por ID”

Paso	Actor	Acción
1	Administrador	Selecciona la opción de gestión de <i>Contigs</i>
2	Aplicación	Muestra un listado de los <i>Contigs</i> de la base de datos
3	Administrador	Introduce el campo descripción del <i>Contig</i> que desea buscar en el espacio de búsqueda
4	Aplicación	Busca y muestra los <i>Contigs</i> con esa descripción

Cuadro 3.58: Caso de uso “Buscar *Contig* por descripción”

- Postcondiciones:
 - ◊ Ninguna.
- Posibles errores:
 - ◊ Error en la conexión a la base de datos.
- Las vistas usadas se muestran en el Cuadro 3.59.

Vista	Descripción
Vista de <i>Contigs</i>	Contiene una vista de los <i>Contigs</i> dados de alta

Cuadro 3.59: Vistas usadas en el caso de uso “Buscar *Contig* por descripción”

- Caso de uso: Buscar *Contig* por SEQ.

Cuando un administrador quiere buscar a un *Contig* por su SEQ.

- Precondiciones:
 - ◊ Ninguna.
- El escenario del caso de uso se puede ver en el Cuadro 3.60.

Paso	Actor	Acción
1	Administrador	Selecciona la opción de gestión de <i>Contigs</i>
2	Aplicación	Muestra un listado de los <i>Contigs</i> de la base de datos
3	Administrador	Introduce el campo SEQ del <i>Contig</i> que desea buscar en el espacio de búsqueda
4	Aplicación	Busca y muestra los <i>Contigs</i> con ese SEQ

Cuadro 3.60: Caso de uso “Buscar *Contig* por SEQ”

- Postcondiciones:
 - ◊ Ninguna.
- Posibles errores:
 - ◊ Error en la conexión a la base de datos.
- Las vistas usadas se muestran en el Cuadro 3.61.

Vista	Descripción
Vista de <i>Contigs</i>	Contiene una vista de los <i>Contigs</i> dados de alta

Cuadro 3.61: Vistas usadas en el caso de uso “Buscar *Contig* por SEQ”

■ Casos de uso: Gestión de *Clusters*.

El diagrama de casos de uso de Gestión de *Clusters* puede verse en la Figura 3.20. En ésta podemos observar que el único actor contemplado es el administrador. Para poder gestionar los *Clusters* es necesario que se identifique, momento en el cual se le ofrecerán las posibilidades especificadas en los requisitos. Puede crear, buscar, modificar o eliminar *Clusters*. Para estas dos últimas funcionalidades es necesario que previamente busque la entidad que quiera editar, de cualquiera de las formas presentadas.

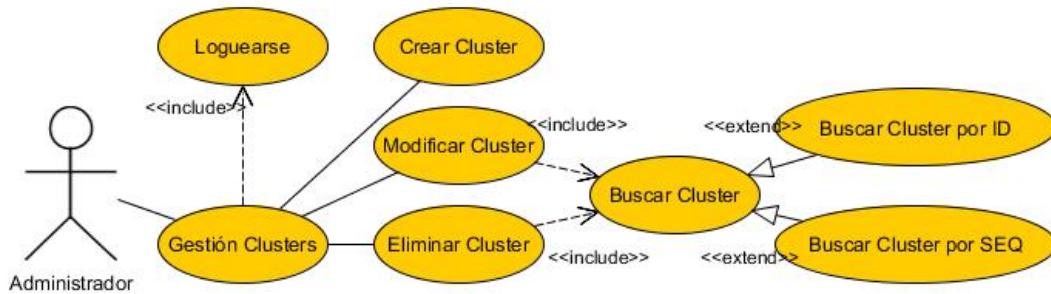


Figura 3.20: Diagrama de casos de uso “Gestión de *Clusters*”

- **Caso de uso: Crear *Cluster*.**

Cuando el administrador quiere añadir un nuevo *Cluster* a la base de datos.

- Precondiciones:
 - ◊ El administrador está autenticado en el sistema.
 - ◊ El *Clusters* no existe en la base de datos.
- El escenario del caso de uso se puede ver en el Cuadro 3.62.

Paso	Actor	Acción
1	Administrador	Selecciona la opción de “añadir <i>Cluster</i> ”
2	Aplicación	Muestra el formulario de creación de un nuevo <i>Cluster</i>
3	Administrador	Cubre el formulario y pulsa la opción de confirmación
4	Aplicación	Muestra una pantalla con todos los <i>Clusters</i> almacenados en la base de datos habiendo dado de alta el seleccionado

Cuadro 3.62: Caso de uso “Crear *Cluster*”

- Postcondiciones:
 - ◊ El *Cluster* se crea en la base de datos.
- Posibles errores:
 - ◊ El *Cluster* ya existe en la base de datos.
 - ◊ Error en la conexión con la base de datos.

- Las vistas usadas se muestran en el Cuadro 3.63.

Vista	Descripción
Vista de <i>Clusters</i>	Contiene una lista de los <i>Clusters</i> de la base de datos
Formulario de <i>Cluster</i>	Contiene un formulario con todos los datos del <i>Cluster</i>

Cuadro 3.63: Vistas usadas en el caso de uso “Crear *Cluster*”

- **Caso de uso: Modificar *Cluster*.**

Cuando un administrador quiere modificar los datos relacionados con un *Cluster*.

- Precondiciones:
 - ◊ El administrador está autenticado en el sistema.
 - ◊ El *Cluster* existe en la base de datos.
- El escenario del caso de uso se puede ver en el Cuadro 3.64.

Paso	Actor	Acción
1	Administrador	Selecciona la opción de “modificar <i>Cluster</i> ”
2	Aplicación	Muestra un listado de los <i>Clusters</i> en la base de datos
3	Administrador	Selecciona el <i>Cluster</i> que quiere modificar
4	Aplicación	Muestra un resumen de los datos del <i>Cluster</i>
5	Administrador	Modifica los datos pertinentes y pulsa la opción de confirmación
6	Aplicación	Muestra un listado de todos los <i>Clusters</i> en la base de datos habiendo modificado el seleccionado

Cuadro 3.64: Caso de uso “Modificar *Cluster*”

- Postcondiciones:
 - ◊ Los datos del *Cluster* se modifican en la base de datos.
- Posibles errores:
 - ◊ Error en la conexión a la base de datos.
 - ◊ El *Cluster* no existe en la base de datos.
- Las vistas usadas se muestran en el Cuadro 3.65.

Vista	Descripción
Vista de Clusters	Contiene una vista de los <i>Clusters</i> dados de alta
Formulario de Contig completo	Contiene un formulario con todos los datos del <i>Cluster</i> seleccionado

Cuadro 3.65: Vistas usadas en el caso de uso “Modificar *Cluster*”

- **Caso de uso: Eliminar *Cluster*.**

Cuando un administrador quiere eliminar a un *Cluster* de la base de datos.

- Precondiciones:
 - ◊ El administrador está autenticado en el sistema.
 - ◊ El *Cluster* existe en la base de datos.
- El escenario del caso de uso se puede ver en el Cuadro 3.66.

Paso	Actor	Acción
1	Administrador	Selecciona la opción de gestión de <i>Clusters</i>
2	Aplicación	Muestra un listado de los <i>Clusters</i> de la base de datos
3	Administrador	Selecciona el <i>Cluster</i> que quiere eliminar
4	Aplicación	Muestra un resumen de los datos del <i>Cluster</i>
5	Administrador	Selecciona la opción de “eliminar <i>Cluster</i> ”
6	Aplicación	Muestra una pantalla de confirmación
7	Administrador	Selecciona la opción de “confirmar”
8	Aplicación	Muestra un listado de los <i>Clusters</i> de la base de datos habiendo borrado el seleccionado

Cuadro 3.66: Caso de uso “Eliminar *Cluster*”

- Postcondiciones:
 - ◊ El *Cluster* es eliminado de la base de datos.
- Posibles errores:
 - ◊ Error en la conexión a la base de datos.
 - ◊ El *Cluster* no existe en la base de datos.
- Las vistas usadas se muestran en el Cuadro 3.67.

Vista	Descripción
Vista de Clusters	Contiene una vista de los <i>Clusters</i> dados de alta
Formulario de Cluster completo	Contiene un formulario con todos los datos del <i>Cluster</i> seleccionado
Pantalla de confirmación de eliminar Cluster	Pregunta al administrador si quiere eliminar el <i>Cluster</i> seleccionado

Cuadro 3.67: Vistas usadas en el caso de uso “Eliminar *Cluster*”

- **Caso de uso: Buscar *Cluster* por ID.**

Cuando un administrador quiere buscar a un *Cluster* por su ID.

- Precondiciones:
 - ◊ Ninguna.
- El escenario del caso de uso se puede ver en el Cuadro 3.68.

Paso	Actor	Acción
1	Administrador	Selecciona la opción de gestión de <i>Clusters</i>
2	Aplicación	Muestra un listado de los <i>Clusters</i> de la base de datos
3	Administrador	Introduce el ID del <i>Cluster</i> que desea buscar en el espacio de búsqueda
4	Aplicación	Busca y muestra los <i>Clusters</i> con ese ID

Cuadro 3.68: Caso de uso “Buscar *Cluster* por ID

- Postcondiciones:
 - ◊ Ninguna.
- Posibles errores:
 - ◊ Error en la conexión a la base de datos.
- Las vistas usadas se muestran en el Cuadro 3.69.

Vista	Descripción
Vista de Clusters	Contiene una vista de los <i>Clusters</i> dados de alta

Cuadro 3.69: Vistas usadas en el caso de uso “Buscar *Cluster* por ID”

- **Caso de uso: Buscar *Cluster* por SEQ.**

Cuando un administrador quiere buscar a un *Cluster* por su SEQ.

- Precondiciones:
 - ◊ Ninguna.
- El escenario del caso de uso se puede ver en el Cuadro 3.70.

Paso	Actor	Acción
1	Administrador	Selecciona la opción de gestión de <i>Clusters</i>
2	Aplicación	Muestra un listado de los <i>Clusters</i> de la base de datos
3	Administrador	Introduce el campo SEQ del <i>Cluster</i> que desea buscar en el espacio de búsqueda
4	Aplicación	Busca y muestra los <i>Clusters</i> con ese SEQ

Cuadro 3.70: Caso de uso “Buscar *Cluster* por SEQ”

- Postcondiciones:
 - ◊ Ninguna.
- Posibles errores:
 - ◊ Error en la conexión a la base de datos.
- Las vistas usadas se muestran en el Cuadro 3.71.

Vista	Descripción
Vista de <i>Clusters</i>	Contiene una vista de los <i>Clusters</i> dados de alta

Cuadro 3.71: Vistas usadas en el caso de uso “Buscar *Cluster* por SEQ”

3.5.3.3. Diseño

Con los requisitos del cliente, se diseñó el nuevo esquema de la base de datos. En comparación con el esquema original de la Figura 3.1 de la página 15, destaca la ausencia de la tabla *Genes*, debido a que el cliente ya no la necesita más. Se han fusionado *ContigSEQ* y *Fulllengther*, integrándolas en la tabla *Contig* (para aumentar eficiencia en las consultas y facilitar el editado de los datos al cliente, por petición suya). También, se ha puesto la tabla *Expression* como un atributo de *Unigenes*. Todos estos cambios pueden observarse en la Figura 3.21.

Además, la Figura 3.22 contiene todo lo relacionado con la gestión de usuarios, grupos y permisos. La tabla *Auth_user* es la que contiene todos los datos de los usuarios, los permisos están en (*Auth_permission*) y los grupos en (*Auth_group*). Las relaciones entre estas instancias son de muchos a muchos (Un grupo puede tener varios usuarios que pueden estar en varios grupos, etc). A la hora de realizar el diseño, se utilizarán clases intermedias para plasmar las relaciones muchos a muchos de una forma plausible.

De este modo, el diagrama de clases completo, incluyendo la totalidad de las entidades es el mostrado en la Figura 3.23. El paso del ER al diagrama de clases de la parte biológica es inmediato, sin embargo la parte de usuarios y permisos es algo mas compleja. Como se comentó con anterioridad, se han sustituido las relaciones muchos a muchos por dos relaciones uno a muchos con una clase intermedia. Ésta clase contiene un identificador único simple y establece las relaciones entre las dos entidades que comunica. Los permisos pueden asignarse tanto a usuarios como a grupos, y los permisos de los que disfruta un usuario son siempre aditivos (si el usuario tiene permisos de lectura y está en un grupo que le da permisos de modificación, el resultado es que disfruta de los dos). Además de los atributos que se pueden observar, Django proporciona una serie de métodos de obtención de los mismos que no se representaron por ser demasiados. Pueden consultarse en la documentación de la API de Django [25].

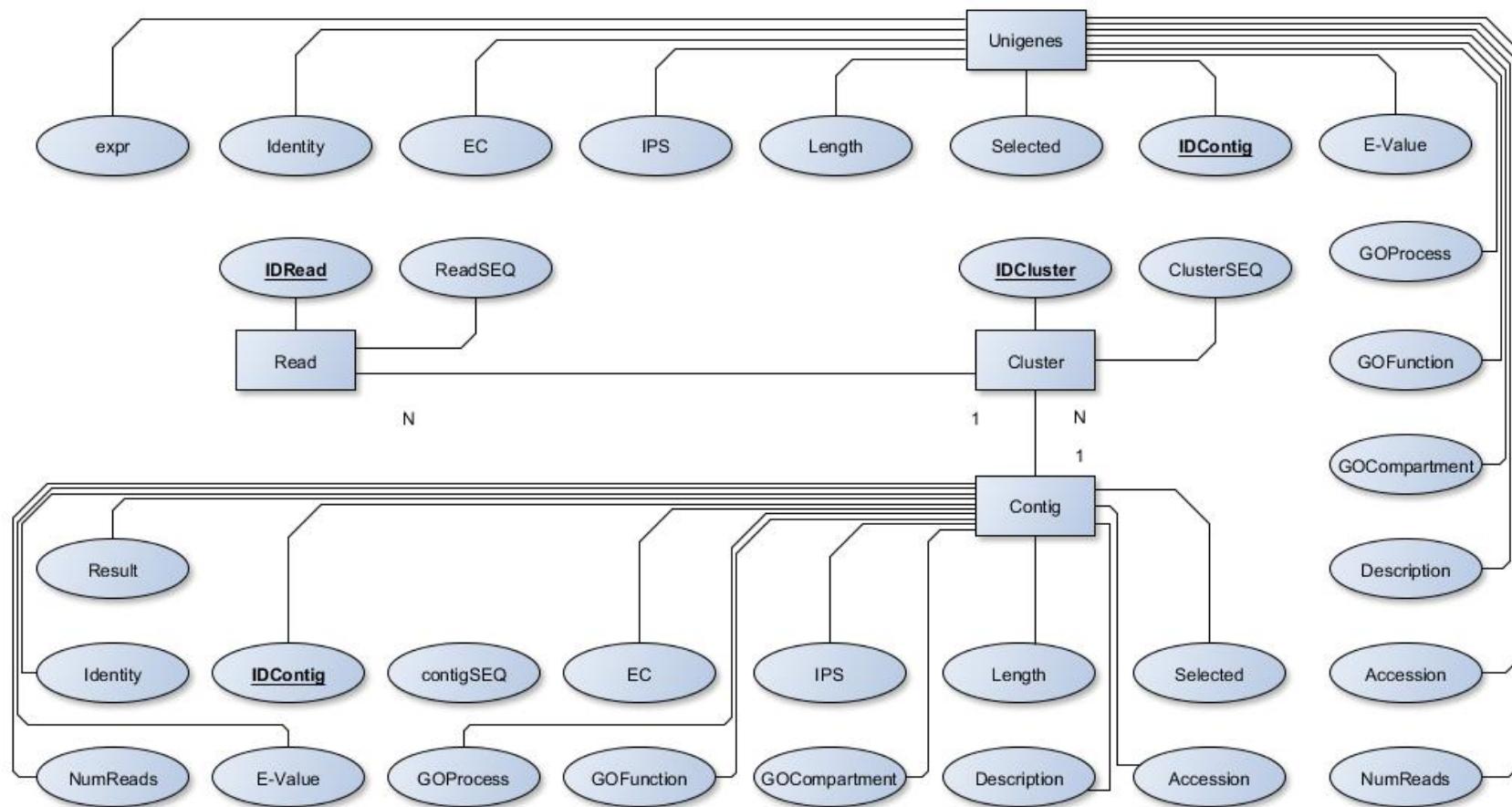


Figura 3.21: Diagrama Entidad-Relación de las entidades principales

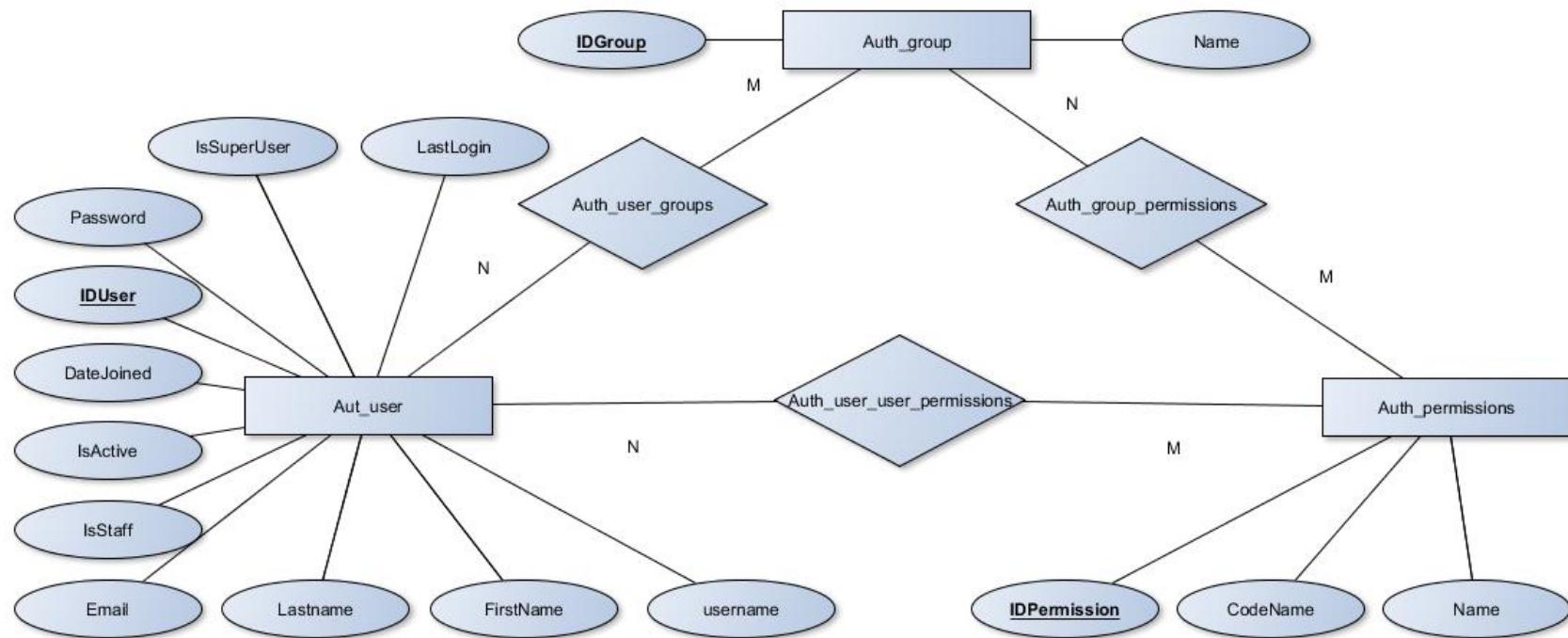


Figura 3.22: Diagrama Entidad-Relación de usuarios, grupos y permisos

3.5. Desarrollo iterativo

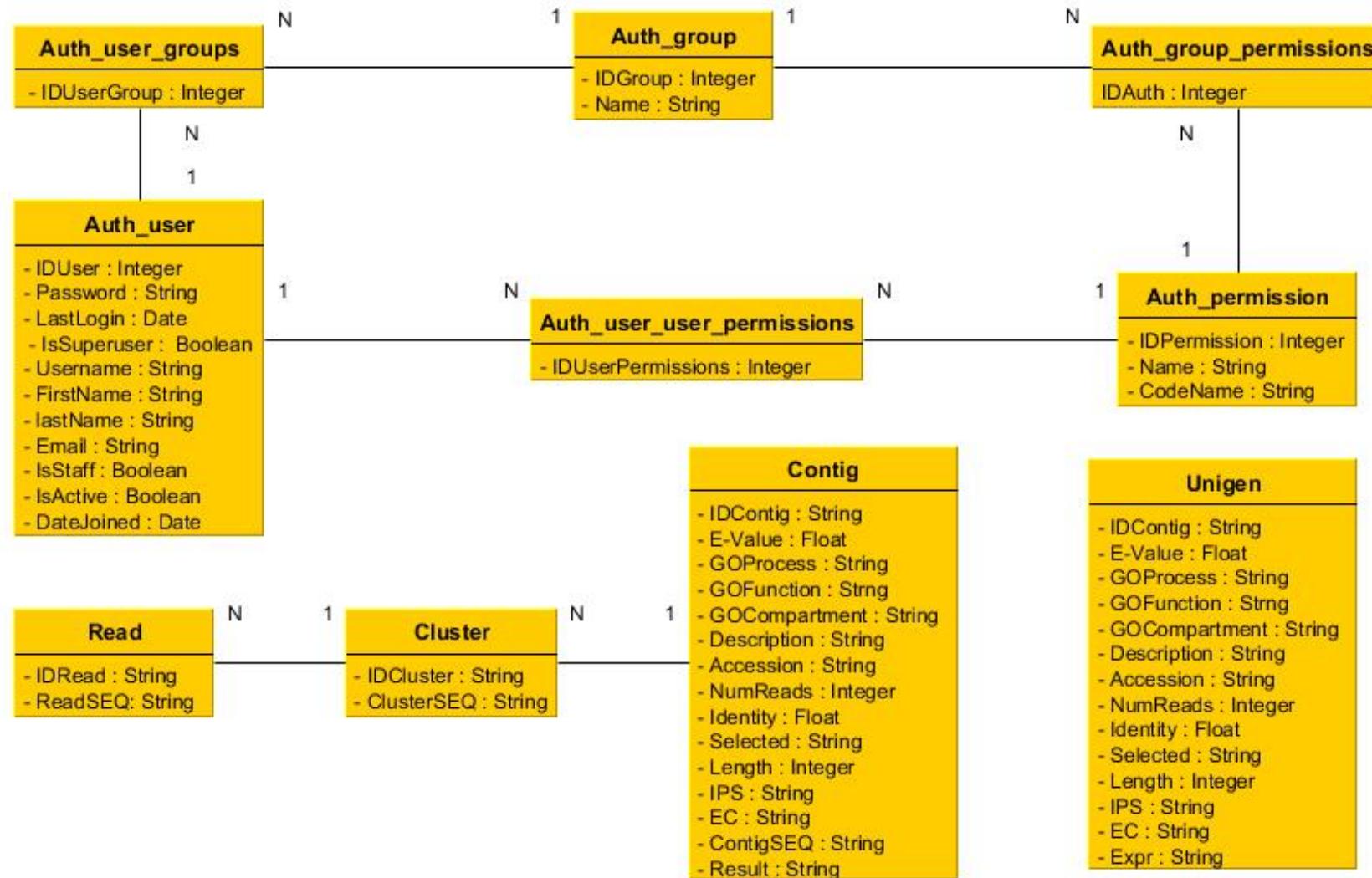


Figura 3.23: Diagrama de clases Final

3.5.3.4. Implementación

Con respecto a la parte biológica de la base de datos (*Reads*, *Clusters*, etc) los cambios introducidos en esta iteración no tuvieron casi impacto negativo en las funcionalidades ya desarrolladas. Se sincroniza el proyecto y Django se adapta automáticamente a los cambios en la estructura de las entidades.

Django proporciona facilidades para ayudar a la implementación de un menú de administración, por ejemplo, automatizando la creación de un sistema de *login* y la encriptación de las contraseñas. De haber tenido que implementar de cero todas las funcionalidades de este *Sprint*, habría contenido para un trabajo de fin de grado entero. En todo caso, a lo largo del desarrollo hubo muchas cosas que no fueron tan automáticas como podría parecer y hubo que lidiar con ellas. Para permitir búsquedas en este menú de administrador hubo que sobreescribir las clases de la API que Django proporciona para satisfacer las necesidades del cliente.

Además, al trabajar sobre los modelos de Django, éste automatiza la creación de un historial de cambios y de acciones recientes. Al ser una funcionalidad extra, se le preguntó al propietario del producto si el cliente deseaba también esa funcionalidad y dio su aprobación.

La Figura 3.24 muestra el seguimiento de *Scrum* realizado en Trello. La imagen a) muestra las nuevas tareas añadidas al *product backlog*. Se intentarán realizar todas ellas en el presente *Sprint*, como se pude ver en la figura b). La última imagen refleja el estado de las listas al finalizar el *Sprint*.

El diagrama de trabajo pendiente al finalizar el *Sprint* puede consultarse en la Figura 3.25.

3.5.3.5. Pruebas

Debido al hecho de que la gestión de usuarios, *Reads*, *Clusters* y *Contigs* se ha realizado con la ayuda de Django, probar su funcionamiento interno carece de sentido.

En su lugar, se ha optado por realizar pruebas de usuario o *User Acceptance Testing* (UAT). Éstas se han realizado en diferentes navegadores (Firefox y Google Chrome) y con los miembros del grupo de investigación de CHROMEVOL, que serán los usuarios finales de la plataforma. Se han definido dos tipos de per-

3.5. Desarrollo iterativo

(a) Product Backlog

(b) Sprint Backlog

Las capturas muestran la interfaz de Trello con tres paneles principales:

- PRODUCT BACKLOG:** Muestra una lista de items con prioridad (naranja) y tareas pendientes (azul). Los items incluyen: Menú administrador con sistema de login (5), Gestión usuarios (2), Gestión grupos (2), Gestión READS (2), Gestión CLUSTERS (2), y Gestión CONTIGS (2).
- SPRINT BACKLOG:** Un panel vacío titulado "Add a card...".
- DONE:** Un panel vacío titulado "Add a card...".

(c) Resultado del segundo Sprint

La captura muestra los mismos paneles que en (a) y (b), pero con cambios en el estado de los items:

- PRODUCT BACKLOG:** Los items permanecen sin cambios.
- SPRINT BACKLOG:** Vacío.
- DONE:** Muestra los resultados del segundo Sprint:
 - Menú administrador con sistema de login: Completado (3/3), 5 tareas.
 - Gestión usuarios: Completado (4/4), 2 tareas.
 - Gestión grupos: Completado (4/4), 2 tareas.
 - Gestión READS: Completado (4/4), 2 tareas.
 - Gestión CLUSTERS: Completado (4/4), 2 tareas.
 - Gestión CONTIGS: Completado (4/4), 2 tareas.

Figura 3.24: Capturas de la herramienta Trello del tercer Sprint

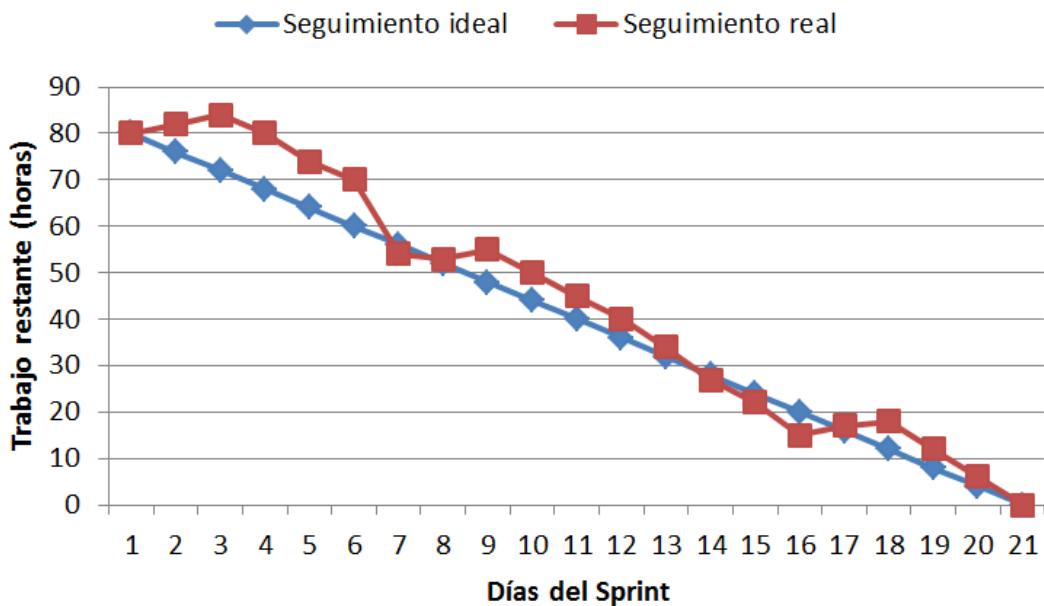


Figura 3.25: Diagrama de trabajo pendiente del tercer *Sprint*

files: usuarios acostumbrados a trabajar con la web antigua y usuarios que nunca trabajaron con ella.

Las tareas propuestas se pueden observar en el Cuadro 3.72.

Los usuarios 2, 3 y 4 son ajenos a la plataforma y los usuarios 5, 6 y 7 realizaron tareas previamente sobre la antigua web. Se ha medido el tiempo que tardaron en realizar las tareas asignadas por primera vez y después de unos minutos una segunda. Éstas se compararon con un usuario de control experto en la web (el autor de el presente proyecto), que se identificó como usuario 1. Destacar que, a pesar de haberles entregado un manual de usuario, ninguno de ellos tuvo necesidad de consultarla en ningún momento y realizaron todas las tareas sin ninguna complicación destacable.

El resultado de las pruebas puede verse en las Figuras 3.26 y 3.27. El primero refleja el tiempo invertido en realizar la totalidad de las tareas por cada uno de los usuarios, y en el segundo aparece desglosado el tiempo promedio en realizar cada tarea.

De estos resultados se pueden extraer las siguientes conclusiones:

- La herramienta resulta muy intuitiva de usar debido a que ninguno de los

Identificador	Tarea
Tarea 1	Añadir un nuevo usuario
Tarea 2	Eliminar un usuario
Tarea 3	Modificar un usuario
Tarea 4	Añadir un nuevo <i>Read</i>
Tarea 5	Modificar un <i>Cluster</i>
Tarea 6	Eliminar un <i>Contig</i>

Cuadro 3.72: Tareas de las pruebas de usuario

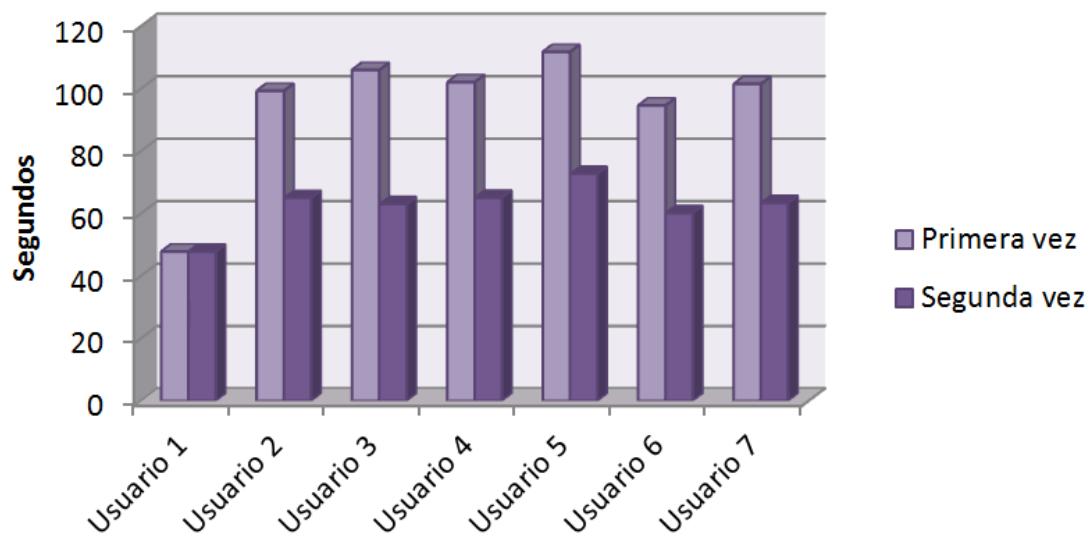


Figura 3.26: Tiempo invertido por los usuarios para completar todas las tareas

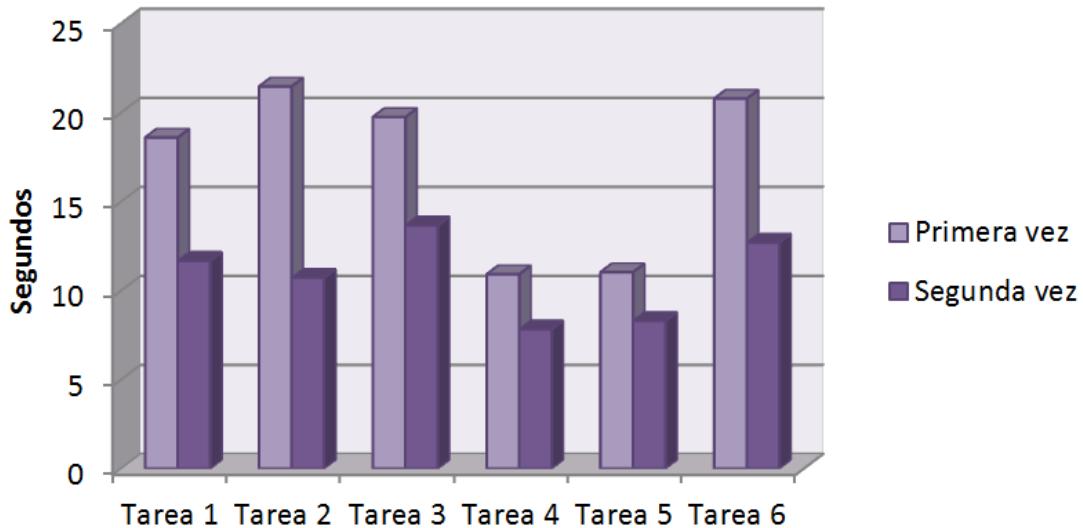


Figura 3.27: Tiempo promedio de cada tarea

usuarios tuvo que recurrir al manual para resolver una tarea.

- Además, la considerable diferencia entre la primera y la segunda prueba denota una rápida curva de aprendizaje.
- El hecho de que no haya diferencias significativas entre los usuarios habituados a la antigua web es debido a que estaban probando funcionalidades que la otra no poseía, por lo que eran tan ajenos a éstas como los nuevos.

Además, durante la realización de las pruebas no se detectó ningún fallo en el funcionamiento de la herramienta.

3.5.4. Desarrollos adicionales

Esta sección engloba todas las características y funcionalidades extra que el cliente no había expresado verbalmente pero el equipo de desarrollo consideró oportuno realizar.

3.5.4.1. Internacionalización

A pesar de que multitud de herramientas traducen el contenido de las páginas web de forma automática, su efectividad es muy inferior a una traducción



Figura 3.28: Logo

específica. El relativamente bajo coste de realizar esta traducción llevó al equipo de desarrollo a realizar una traducción específica para todos los contenidos de la web. Django proporciona una cómoda (una vez que se sabe usar) forma de implementar i18n (internationalization). Éste es el proceso de diseñar software de manera tal que pueda adaptarse a diferentes idiomas y regiones sin la necesidad de realizar cambios de ingeniería ni en el código.

Se dispuso de tal forma que el sistema detectase automáticamente el idioma del sistema operativo del usuario para acomodar esta traducción al mismo, lo que se conoce como l10n (localization) [26]. A la hora de analizar las necesidades del cliente, el equipo de desarrollo se dio cuenta de que el sistema está dirigido a investigadores de Miami, siendo la mayoría hispanohablantes trabajando en equipos de laboratorio en inglés. Es por ello que este automatismo podría no ser todo lo beneficioso que se pretendía. Por lo que se decidió añadir un botón para cambiar el idioma entre Español e Inglés bajo demanda. Además, se diseñó el botón de tal forma que resultaría muy sencillo añadir más idiomas al mismo.

3.5.4.2. Logo renovado

El equipo de desarrollo quería remarcar la transición entre el sistema antiguo y el nuevo, de manera que también se diseñó un nuevo logo para terminar de añadir modernismo a la web. Se puede observar en la Figura 3.28.

3.5.4.3. Panel de acciones recientes

Django proporciona de una forma sencilla un panel de acciones recientes en el menú de administrador. Al grupo de desarrollo le pareció buena idea incluirlo debido al poco tiempo que requería. En él pueden verse las acciones realizadas sobre las tablas, con enlaces a las mismas.

Capítulo 4

Análisis de datos

Índice general

4.1. Introducción	91
4.1.1. Definiciones	92
4.1.2. Objetivos	92
4.1.3. Descripción de datos experimentales	93
4.1.4. Trabajos relacionados	96
4.1.5. Elección de la tecnología	97
4.2. Análisis	98
4.2.1. Aproximación 1	99
4.2.2. Aproximación 2	100
4.2.3. Aproximación 3	102
4.3. Discusión	107

En este capítulo se detalla todo el proceso del análisis de los datos, desde un exhaustivo análisis de la base de datos, con la elección de la tecnología, pasando por los objetivos hasta las diferentes aproximaciones realizadas.

4.1. Introducción

Esta sección engloba toda la parte previa al análisis propiamente dicho.

4.1.1. Definiciones

A continuación se incluyen una serie de definiciones que se han considerado necesarias para una mejor comprensión de esta segunda parte del proyecto.

- **Bioinformática:** Rama de la ciencia que consiste en la aplicación de tecnologías de computadores, estadística, matemáticas y procesos ingenieriles a la gestión y análisis de datos biológicos. Engloba todos los algoritmos y técnicas utilizadas para solucionar o investigar problemas sobre escalas de tal magnitud que sobrepasan el discernimiento humano.
- **Ontología:** Exhaustivo y riguroso esquema conceptual dentro de uno o varios dominios dados, que se diseña con la finalidad de facilitar la comunicación y el intercambio de información entre diferentes sistemas y entidades.
- **Gene Ontology (GO):** El proyecto Ontología Génica, o GO, provee un vocabulario controlado que describe el gen y los atributos del producto génico en cualquier organismo. Puede ser, en general, dividido en dos partes. La primera es la ontología por sí misma mientras que la segunda es la anotación, es decir, la caracterización de los productos génicos usando términos de la ontología.
- **Librería (de tránscritos):** Colección de fragmentos de ARN de un solo organismo bajo unas condiciones experimentales específicas.
- **Gene enrichment:** Método para identificar genes o proteínas que están sobre-expresados o inhibidos entre múltiples librerías.
- **Clustering:** Técnica de agrupación de una serie de vectores de acuerdo con un criterio, por lo general de distancia o similitud. La cercanía se define en términos de una determinada función de distancia, la cual se trata de minimizar entre los elementos del propio clúster.

4.1.2. Objetivos

Los objetivos que se pretenden conseguir con esta etapa de análisis es identificar los *Contigs* relacionados entre sí y caracterizar esta relación. Cada uno de estos *Contigs* posee diversas etiquetas en función de las tareas que lleva a cabo

dentro de la célula, y serán estas etiquetas las que se usarán para agruparlos. Los resultados obtenidos serán transmitidos al grupo CHROMEVOL para su análisis e interpretación biológica, con el objetivo último de encontrar relaciones entre perfiles de expresión génica y los niveles de biotoxina en el agua de mar. Ésto derivará en la obtención de potenciales marcadores genotóxicos.

4.1.3. Descripción de datos experimentales

En esta sección se explicarán detalles sobre los datos utilizados, tanto pertinentes a su extracción como a su interpretación.

4.1.3.1. Creación de la base de datos

La ya existente base de datos de CHROMEVALOA fue creada con la intención de evaluar la contaminación del ácido okadaiko en el entorno marino, y está basada en el transcriptoma del molusco *Mytilus galloprovincialis*. La representación de este ácido puede verse en la figura 4.1

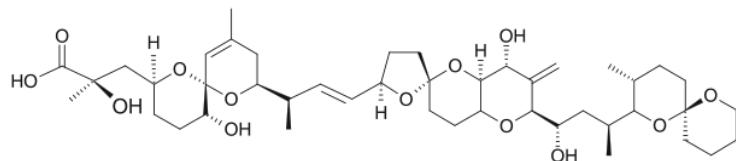


Figura 4.1: Estructura del ácido okadaiko.

Para ello, se dispuso de dos poblaciones de mejillones, una tratada (NORM_MGT) y otra de control (NORM_MGC); ambas forman las librerías que son el objeto del presente estudio. La población tratada fue introducida en un entorno contaminado con un alga productora de ácido okadaico (*Prorocentrum lima*) con la intención de observar los daños producidos, comparándola con la de control. La Figura 4.2 es un esquema de este diseño experimental.

Para cada población se trajeron lecturas del ADN (*Reads*) de su glándula digestiva. Éstas lecturas se agruparon por similaridad en *Clusters*, los cuales se representan mediante un *Contig* o secuencia de consenso. Además, estos *Contigs* poseen diversos atributos de interés biológico. Los *Unigenes* son *Contigs* asociados

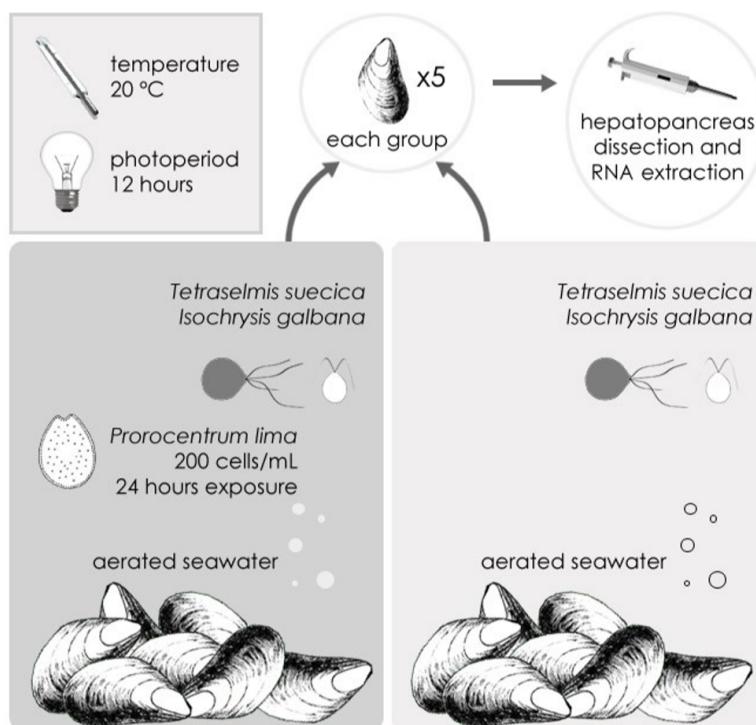


Figura 4.2: Proceso de extracción génica de las dos poblaciones de moluscos.

a la cromatina. Se trata, por lo tanto, de datos reales provenientes de un estudio del área de las ciencias marinas.

La estructura de la base de datos puede verse en la Figura 3.21 de la página 82.

4.1.3.2. Explicación de los datos

Se tienen *Contigs* (secuencias de ADN) etiquetadas con sus correspondientes *GOTerms* (términos GO), obtenidos mediante un proceso de *Gene enrichment*. Este etiquetado se ha realizado con los datos proporcionados por la iniciativa *Gene Ontology*, que ha identificado miles de términos y funciones y los ha clasificado en una enorme jerarquía disponible globalmente. Así, la base de datos está formada por una serie de *Contigs* con atributos *booleanos*. Cada uno se refiere a un *GOTerm* distinto, que indica si está o no anotado. A su vez, cada *GOTerm* desciende de los llamados términos raíz, que son los siguientes:

- *Biological Process*: El proceso celular que ayuda a llevar a cabo.

- *Biological Function*: Hace referencia a la función que realiza la secuencia.
- *Cellular Compartment* o *Cellular Component*: El lugar u orgánulo que se ve afectado.

En la Figura 4.3 se puede ver un pequeño ejemplo de la estructura de *Gene Ontology*. Cabe destacar que el árbol tiene una profundidad e índice de ramificación muchísimo más grande.

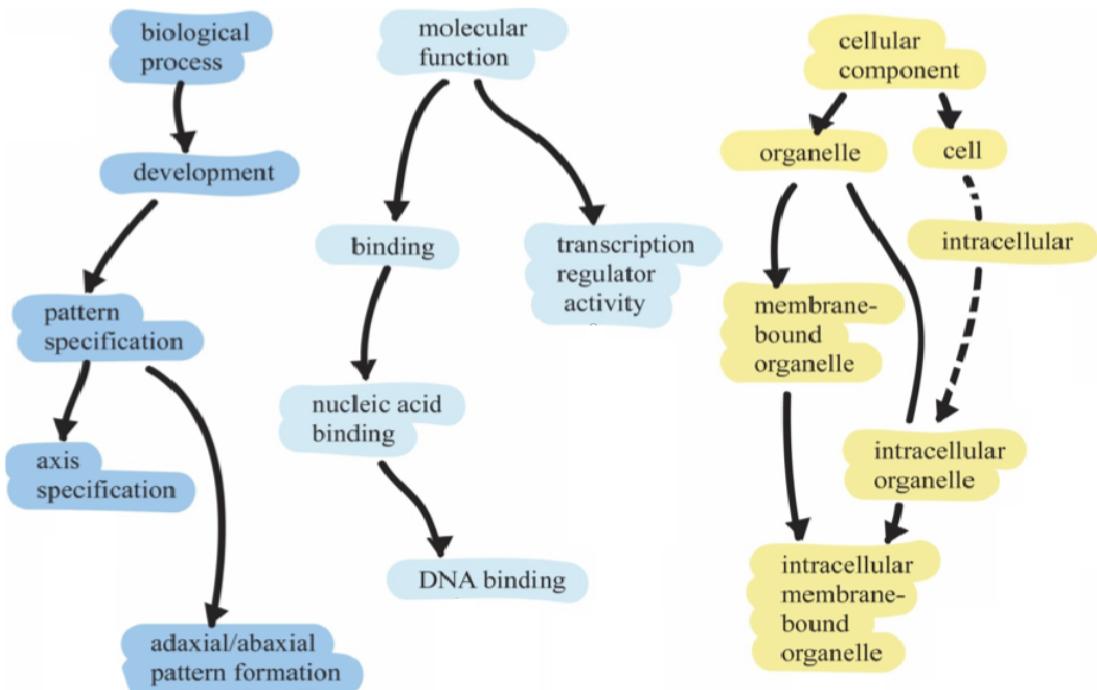


Figura 4.3: Diagrama de muestra de la estructura de Gene Ontology.

La Figura 4.4 procede de la página web oficial de *Gene Ontology*. En ella se muestran los ancestros del *GOTerm Nucleus*. Se puede observar que está clasificado como Cellular Compartment, además de una larga lista de hijos de la misma. Cabe destacar que el hecho de que un *GOTerm* tenga dos padres directos no supone ninguna contradicción, debido a la naturaleza totalmente relacionada de los mismos.

id	name	term_type	distance	ancestor_id
4815	intracellular membrane-bounded organelle	cellular_component	1	4815
21993	membrane-bounded organelle	cellular_component	2	21993
21995	intracellular organelle	cellular_component	2	21995
21992	organelle	cellular_component	3	21992
23119	intracellular part	cellular_component	3	23119
310	cellular_component	cellular_component	4	310
23157	cell part	cellular_component	4	23157
42841	all	universal	5	42841
297	intracellular	cellular_component	4	297
4662	cell	cellular_component	5	4662
4672	nucleus	cellular_component	0	4672
23157	cell part	cellular_component	5	23157
310	cellular_component	cellular_component	6	310
42841	all	universal	7	42841

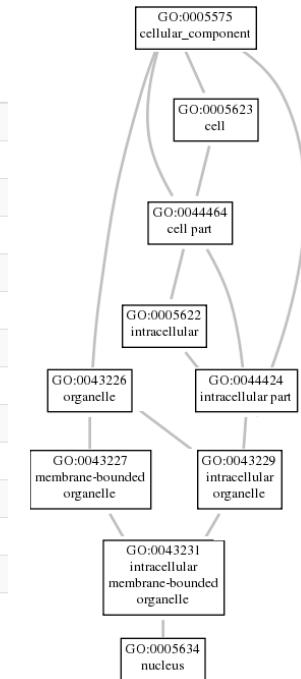


Figura 4.4: Ancestros del GOTerm Nucleus.

4.1.4. Trabajos relacionados

Existen esfuerzos previos para realizar clustering sobre datos de *Gene Ontology*. Prueba de ello es la disponibilidad de herramientas para que los biólogos analicen sus datos. Una de las más conocidas es DAVID [27, 28], que permite multitud de funciones además de enriquecer las secuencias con términos de su base de datos. Se consideró la posibilidad de utilizar esta herramienta, que simplificaría enormemente la tarea de análisis. El problema es que el dominio del presente trabajo está muy acotado, y DAVID es una herramienta muy genérica, lo que produciría anotaciones no relevantes para el problema que se está tratando. Sin embargo, esta herramienta ha contribuido a realizar importantes investigaciones en el campo de la bioinformática. En [29] y [30] se pueden consultar ejemplos en los que se utilizó DAVID para hacer *clustering* sobre datos genéticos. El problema de la presente investigación es, como se expone en [31], el escaso conocimiento disponible sobre la secuenciación del ADN y las proteínas de bivalvos. Ésto constituye una importante barrera a la hora de analizar los datos, por lo que no existen herramientas que automaticen de forma eficaz el anotado y análisis de

sus términos. Concretamente, sobre la especie tratada en este proyecto (*Mytilus galloprovincialis*) no se conoce ningún intento de agrupar términos ni de realizar un análisis parecido al que se pretende.

4.1.5. Elección de la tecnología

En este apartado se explicarán las herramientas y técnicas utilizadas y las ventajas frente a otras opciones existentes.

4.1.5.1. *K-Means*

K-Means es un método de agrupamiento (*clustering*) que tiene como objetivo la partición de un conjunto de n observaciones en k grupos en el que cada instancia pertenece al grupo más cercano a la media. Intenta minimizar la suma de los cuadrados de las distancias dentro de cada *cluster* (WCSS) mediante la fórmula:

$$\operatorname{argmin}_S \sum_{i=1}^k \sum_{x \in S_i} \|x - \mu_i\|^2$$

donde μ_i es la media de los puntos en el grupo S_i . Existen multitud de diferentes distancias aplicables (Euclídea, Manhattan, Hamming, etc). En éste proyecto se usará la distancia Hamming (número de bits diferentes) debido a su adecuación a datos binarios.

K-Means es un método ampliamente utilizado en minería de datos para agrupar mediciones. El resultado es una partición del espacio de datos en celdas de Voronoi, que determinan una serie de polígonos alrededor de un conjunto de puntos de control. Su uso es especialmente indicado cuando no se tiene conocimiento de la distribución de las muestras, debido a que es un método no supervisado. Se usará en este proyecto debido a esta última razón ya que obtiene mejores resultados que los mapas autoorganizativos o SOM (*Self-Organizing Maps*) con valores booleanos.

4.1.5.2. Matlab

Existen numerosas herramientas de análisis de grandes cantidades de datos, las más famosas son R y Matlab. En este trabajo se ha optado por la segunda.

Aunque funcionan de forma muy similar, las ventajas de Matlab frente a R pueden sintetizarse en los siguientes puntos:

- Entorno de programación más cómodo.
- Generalmente más eficiente.
- Mayor número de extensiones.
- Mejor soporte para las ciencias, mientras que R está mas orientado a la estadística.

Matlab es un lenguaje de programación de alto nivel, con un entorno interactivo utilizado por millones de ingenieros y científicos en todo el mundo. Permite explorar y visualizar ideas, así como realizar tareas en procesamiento de señales e imagen, comunicaciones, sistemas de control y finanzas computacionales entre otros. Es especialmente eficiente en el tratamiento de grandes matrices, algo esencial en este proyecto. Además, dispone de infinidad de *toolboxes* con librerías especializadas en diferentes áreas. En este proyecto se han usado las librerías referentes a *K-Means*, en concreto la conocida *Statistics and Machine Learning Toolbox*.

4.2. Análisis

En la presente sección se explicarán las diferentes aproximaciones realizadas.

El cliente quiere agrupar los *Contigs* por similaridad de *GOterms*. Como no se dispone de información acerca de cómo deberían agruparse esos datos, se utilizarán técnicas de clustering, concretamente *K-Means*.

A la hora de analizar los datos se evaluó concienzudamente la posibilidad de utilizar los datos disponibles en la base de datos de AmigGO (aplicación web que permite realizar consultas a la jerarquía de *GeneOntology*). Dispone de una API pública para consultar los ancestros y descendientes de un término, lo cual facilita la obtención de relaciones entre los mismos, además de producir un análisis de mayor calidad. Finalmente no se optó por esta aproximación debido al enorme gasto de tiempo que supondría, para todos los términos disponibles, almacenar las interrelaciones entre ellos en una matriz computable por un ordenador. Así,

se consideró que un análisis con tal nivel de detalle constituiría un proyecto por sí mismo, por lo que se decidió descartarlo. Sin embargo, todo el tiempo y esfuerzo invertido en investigar y sopesar esta información fue comunicada al grupo CHROMEVOL para que en futuras investigaciones pudieran hacer uso de ella.

Así, se decidió que el formato de presentación de los datos fuera booleano, de forma que indique qué *Contigs* están etiquetados con cada término. Se disponen de 41.019 instancias y de 6.558 *GoTerms*.

4.2.1. Aproximación 1

Para esta primera aproximación se analizaron los datos en bruto, sin realizar ningún tipo de proceso de reducción de dimensionalidad más que la eliminación de términos duplicados. De esta manera se pretendía comprender mejor la naturaleza de los datos y sopesar los caminos a seguir a continuación.

Se construyeron tres grupos de datos binarios para cada uno de los diferentes tipos de *GO Terms* (ver Sección 4.1.3), cada una con sus vectores identificativos de los términos correspondientes.

Debido a que *K-Means* es un algoritmo con una componente aleatoria, no es recomendable ejecutarlo solo una vez ya que el resultado varía para cada ejecución. Es por ésto que dos elementos no tienen necesariamente que volver a coincidir en el mismo *cluster*. Al no tener ninguna pista de cuantos grupos debería haber ni de cuántos elementos deberían asignarse a cada *cluster*, se optó por ejecutar *K-Means* para cada uno de los tres grupos de datos, con diferentes valores de *K* (de 5 a 5500), repitiendo el proceso 100 veces. Para cada par de instancias se guardó cuantas veces coincidían en el mismo *cluster*. Debido a que la presente es una primera aproximación al problema, la medida de similitud que se utilizó fue la distancia euclídea.

Los resultados obtenidos indicaban una medida de la relación entre los distintos *Contigs*. Se enviaron al grupo CHROMEVOL como una primera aproximación para que realizasen una evaluación y determinasen los siguientes pasos a seguir. Como resultado de dicho intercambio, el grupo manifestó su deseo de acotar los datos ante el enorme tamaño de los presentes.

4.2.2. Aproximación 2

La anterior aproximación resultó muy pesada de ejecutar, tardando varios días, por lo que el grupo CHROMEVOL decidió simplificar lo máximo posible los datos. Al haber un número excesivamente grande de *Contigs* en la base de datos, no todos son de interés para sus objetivos, se decidió analizar tan solo los *Unigenes*. Conviene recordar que éstos son los *Contigs* asociados con la cromatina, que es en lo que se centra el grupo CHROMEVOL, y por lo tanto les resultan más relevantes. Además, se redujo el conjunto de análisis solamente al término *Biological Process* (una de las tres grandes categorías de *GeneOntology*), por ser considerado el más relevante para ellos.

Inicialmente, se disponía de 1124 *Unigenes*, cada uno con 1308 posibles *Biological Process* asociados. Pero al analizar más detenidamente los datos resultó evidente que podían reducirse todavía más, debido a que había *Unigenes* que no tenían ningún *Biological Process*. De este modo se pudo reducir su número hasta 428.

Al tratarse de atributos binarios, la distancia que más se adecua es la Distancia de Hamming (número de bits diferentes entre las dos cadenas a comparar).

Una vez simplificados los datos, se procedió a realizar la operación de *clustering*, con diferentes parámetros para K (de 10 a 420) y 20 repeticiones. Esto generó una inmensa cantidad de datos debido a que, para cada operación de *clustering*, se guardó la siguiente información:

- A qué *cluster* pertenece cada instancia.
- La localización del centroide de cada *cluster*.
- La suma de las distancias de cada punto a su centroide.
- La distancia de cada punto a todos los centroides.

A continuación se buscaron los *clusters* para los cuales todos los elementos tienen una distancia a su centroide menor que un umbral determinado, lo que ocasiona que estén poco dispersos (que se parezcan mucho entre ellos). Este umbral fue establecido de forma relativa, como un 1% de la distancia del punto más alejado a su centroide, debido a que los datos parecían estar muy cerca al mismo (siendo esto lo que se buscaba).

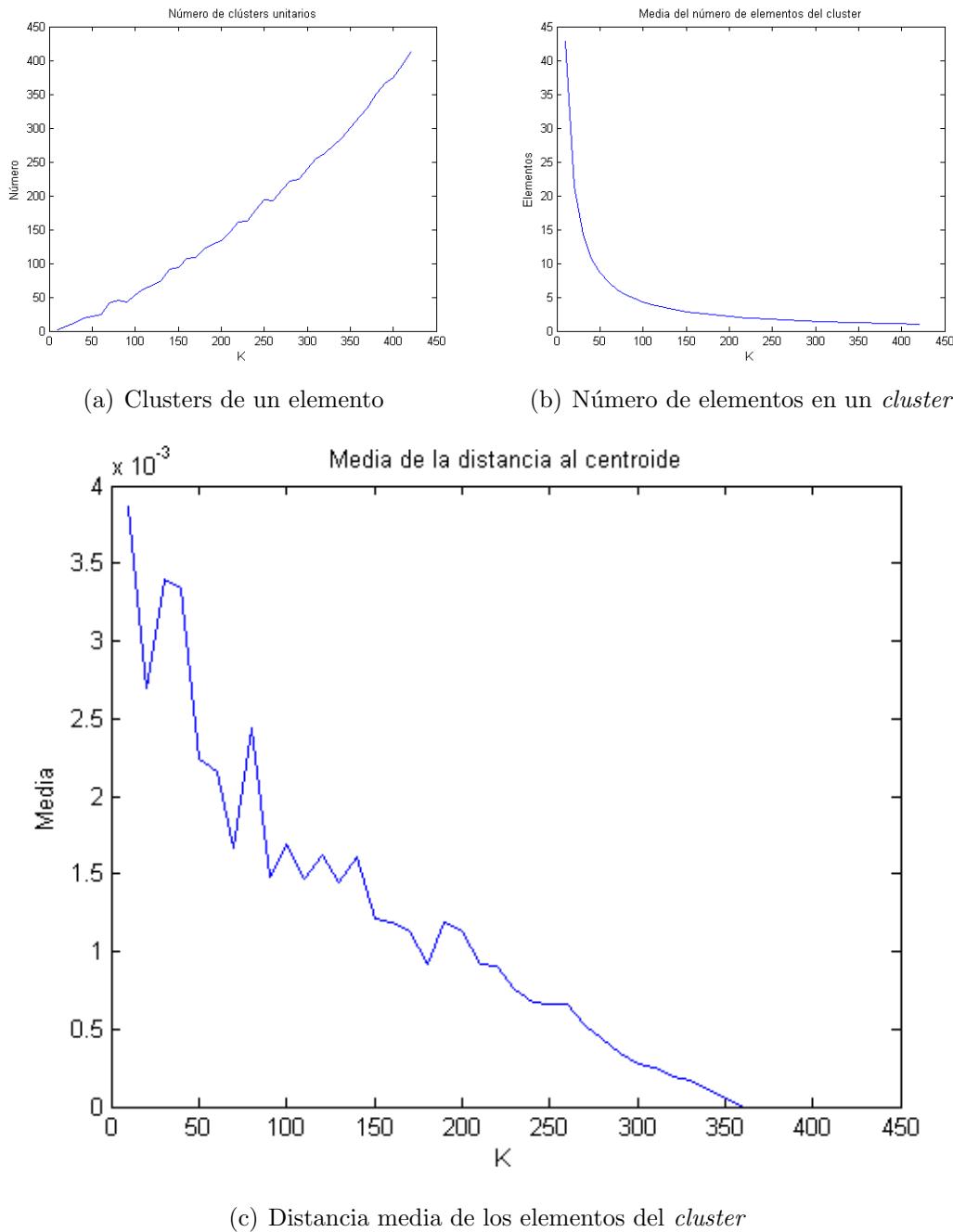


Figura 4.5: Figuras resultado de la segunda aproximación

En la Figura 4.5 se pueden observar los gráficos extraídos. El a) y el b) resultan evidentes: cuanto más se aumenta K (mayor número de *clusters*), más se disminuye el número medio de elementos en cada uno, aumentando al mismo tiempo el número de *clusters* unitarios (que solo tienen una instancia). Con respecto a estos últimos, cabe destacar su gran número en comparación con K . Ésto es bueno debido a que la base de datos tiene instancias dispersas (que siempre caen en *clusters* unitarios) y también algunos cúmulos aislados, que son los que se quiere identificar. En cambio el diagrama c) muestra que, según de aumenta K , la distancia media centroide-instancias de los *clusters* con más de un elemento disminuye. Solamente se tuvieron en cuenta los *clusters* no unitarios debido a que un *cluster* unitario tiene distancia cero por definición. A medida que aumenta K hay un mayor número de *clusters* de un elemento, por lo que los que tienen más de una instancia serán muy parecidos entre sí, derivando en una distancia mucho menor. El que la distancia global sea realmente pequeña sugiere muy poca varianza dentro de un mismo *cluster*, que es lo que es lo que se busca con esta técnica. Sin embargo, al observar los datos detenidamente, resultaba evidente que había instancias repetidas. Es por esto que a partir de un valor de K de 350 la distancia media cae a cero, ya que todos los *clusters* no unitarios están formados por instancias idénticas (con distancia cero).

El hecho de que la mayoría de las distancias medias fuese cero era debido que los atributos de estas instancias eran idénticos. De este modo siempre caían en los mismos *clusters* las instancias iguales entre sí, evitando que se manifestaran posibles similitudes ocultas, que es el objetivo del presente análisis. Al comprobar en la base de datos estas instancias, se observó que estaban etiquetadas con los mismos *GOTerms* pero el resto de sus campos eran diferentes, por lo que la base de datos no tenía ningún problema.

4.2.3. Aproximación 3

El objetivo de esta aproximación es evidente teniendo en cuenta los resultados de la aproximación anterior: se pretende agrupar esas instancias idénticas para hacer *clustering* sobre una instancia de cada tipo.

Así, en una etapa de pre-procesado se obtuvieron las instancias únicas, descartando las repetidas.

Seguidamente se procedió a realizar el *clustering* de los datos simplificados,

repitiendo 20 veces cada experimento. Debido a que se tienen casi 100 instancias menos, se realizaron experimentos con un K menor que en la aproximación anterior (de 5 a 285), lo que influye en los resultados. Las gráficas de la distribución de las instancias en los *clusters* pueden verse en la Figura 4.6. Los gráficos a) y b) son muy parecidas a los de la Figura de la aproximación anterior. Con respecto a c), la forma global de la misma es similar debido a que la distribución de la base de datos es idéntica. Sin embargo, la distancia es mayor que en el caso anterior. Ésto es debido al haber eliminado las instancias repetidas que caían en *clusters* de varios elementos idénticos y con distancia cero.

Una vez realizada la operación de *clustering*, se volvieron a filtrar los grupos interesantes. Al igual que se hizo en la aproximación anterior, se buscaron los *clusters* para los cuales todos los elementos tuvieran una distancia a su centroide menor que un umbral. En la aproximación anterior, este umbral se estableció como un 1 % de la distancia del punto más alejado a su centroide (por lo que el resultado eran los *clusters* con instancias idénticas en él). En este caso, al no haber repetidos, el umbral se estableció como un 50 % de la distancia del punto más alejado a su centroide. De este modo se obtuvieron una serie de *clusters* prometedores (con distancias mayores que cero).

Los datos de esa operación se enviaron al grupo CHROMEVOL para que analizaran si esos resultados tenían significado biológico. Debido a la complejidad de los resultados, los investigadores estimaron que era necesario realizar un post-procesado de los datos proporcionados, que aportara información adicional para facilitar su interpretación.

Así, se ideó un método para calcular la importancia relativa de cada atributo en el *cluster*. Partiendo de la conocida posición de cada centroide y de cada instancia en el espacio N dimensional, y de la distancia de cada instancia a cada centroide, se fue eliminando cada atributo y recalculando la distancia centroide-instancias. Ésta se dividió entre la consecuente suma de la variación de la distancia de todos los atributos para ese *cluster*, con el objetivo de conseguir la importancia relativa de cada atributo para cada *cluster*. Esta necesidad de calcular la importancia de cada atributo limita la posibilidad de realizar seguidamente una aproximación con un análisis de componentes principales (PCA). Ésta es una técnica estadística utilizada para reducir la dimensionalidad de un conjunto de datos y ordenarlas por importancia, para así eliminar las que no aporten información al

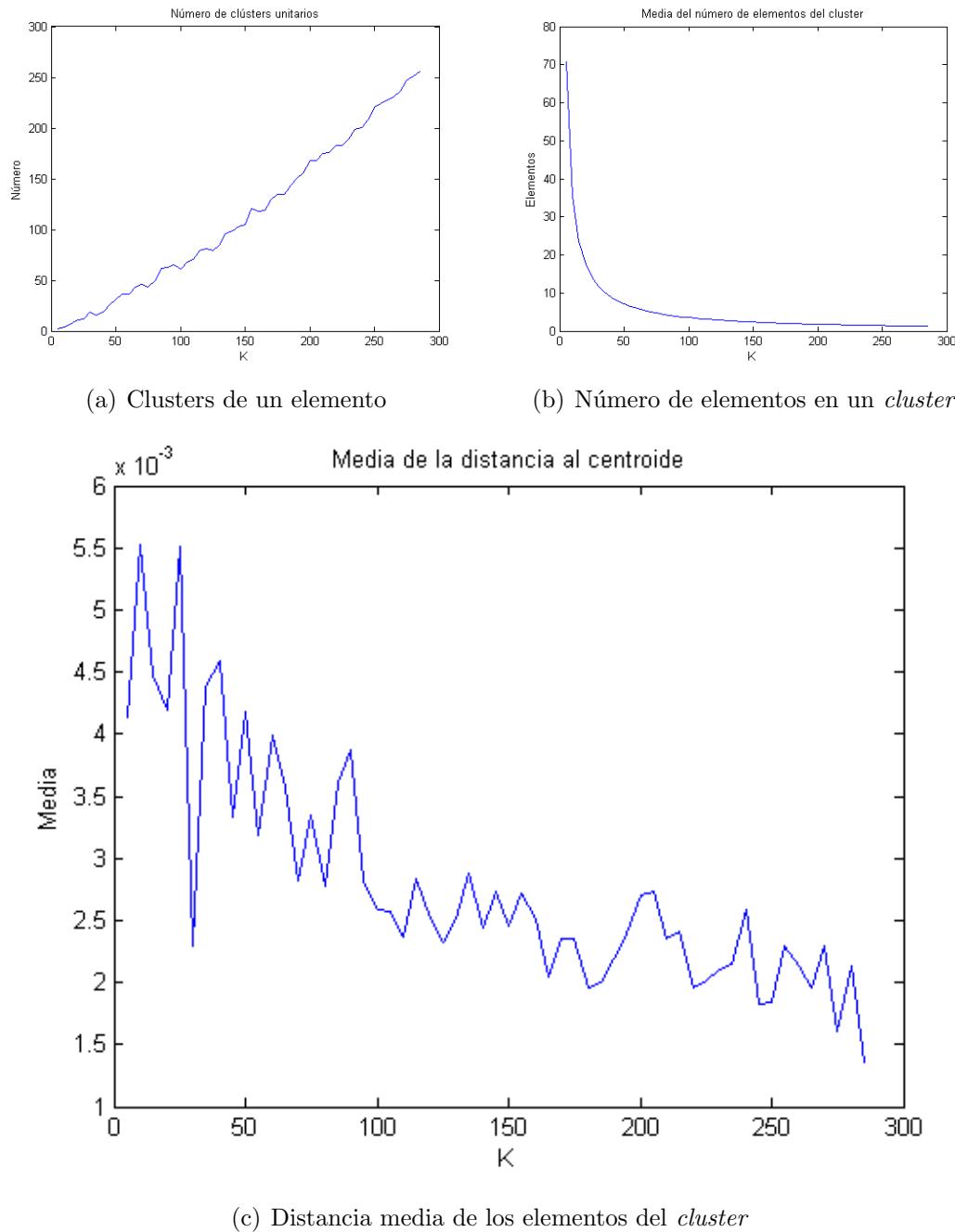


Figura 4.6: Figuras resultado de la tercera aproximación

experimento. No se usará en este proyecto debido a que estas técnicas pierden la identidad de cada atributo, reemplazándolo por una combinación lineal de todos ellos, lo que impediría realizar una posterior clasificación por importancia de los mismos.

Además, el grupo CHROMEVOL quería una medida de similaridad entre *Unigenes* que debía de ser independiente del K usado y del número de instancias en los grupos. Así, se unieron todos los *clusters* resultantes de esta aproximación (para K desde 5 a 300 y 20 repeticiones por operación). El grupo resultante se ordenó por la distancia media de todas las instancias del *cluster* con respecto a su centroide, eliminando los duplicados (debido a que muchos grupos se repiten a lo largo de las ejecuciones).

De este modo se obtuvieron los siguientes datos:

- Las instancias de todos los *clusters* realizados.
- Las distancias medias al centroide de todos los *clusters*.
- Importancia relativa de cada atributo en cada *cluster*.
- Correspondencia de cada identificador a cada *Unigen*.

Las Figuras 4.7 y 4.8 muestran una visión global de la distribución de los datos de forma independiente de K , en función de la distancia. Se puede observar que la gran mayoría de los *clusters* se agrupan en una distancia pequeña, debido a que hay muchos *clusters* con pocas instancias. Además, el número de elementos de cada *cluster* alcanza su máximo con un valor de distancia muy alto. Ésto es debido a que con un valor de K muy pequeño, se agrupan muchas instancias muy poco similares (con una distancia alta) en un mismo *cluster*.

Es el grupo CHROMEVOL el que tendrá que decidir la distancia a partir de la cual filtrará los *clusters* por ser demasiado genéricos o poco informativos.

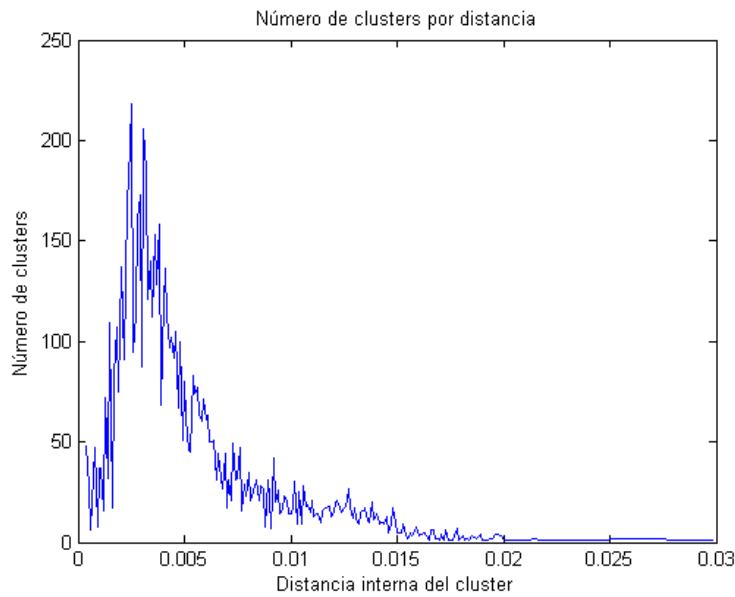


Figura 4.7: Número de *clusters* en función de la distancia.

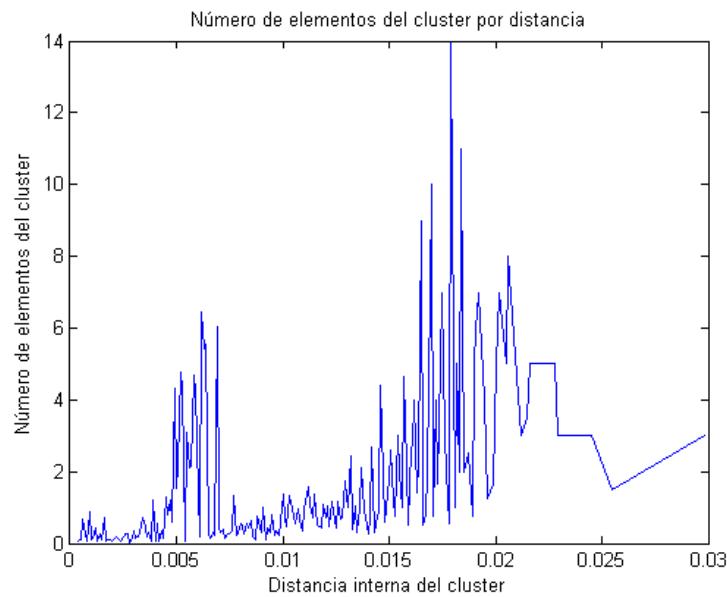


Figura 4.8: Número de elementos de cada *clusters* en función de la distancia.

En los Cuadros 4.1 y 4.2 se pueden observar varios ejemplos significativo de los resultados de este proceso. No se muestra la totalidad de los mismos debido

a su gran tamaño. Sin embargo, pueden consultarse en el CD entregado adjunto a la presente memoria, en el cual se adjuntan también los resultados de todas las aproximaciones, el código utilizado y las tablas originales.

El Cuadro 4.1 muestra los *clusters* obtenidos, ordenados por su distancia interna. Además, también se muestra el identificador de las instancias de cada *cluster* y la distancia media a su centroide.

Por su parte, en el Cuadro 4.2 se muestra, para cada *cluster*, los atributos que lo definen y su importancia relativa. Se han seleccionado algunos de los mostrados en el Cuadro 4.1 para facilitar su comprensión.

4.3. Discusión

Debido a que se buscaban fuertes similaridades se trabajó con un número de *clusters* muy elevado. Ésto propiciaba que todos los elementos agrupados en él fueran lo más parecidos posibles, ya que las instancias diferentes quedaban aisladas en *clusters* unitarios. El número de estos *clusters* de un elemento aumentaba conforme lo hacía K , por lo que la media de elementos de cada *cluster* y la distancia dentro de cada uno de ellos disminuía. Las instancias agrupadas bajo estas condiciones lo hacían de forma reiterada con independencia de las ejecuciones, debido a su gran similaridad.

Los resultados obtenidos se transmitieron al grupo CHROMEVOL y, a la fecha de entrega del presente proyecto, se continúa esperando los resultados. Así, aunque resta extraer las conclusiones biológicas sobre los datos expuestos, se han completado sin embargo todos los objetivos del análisis informático. Se espera que dichas conclusiones contribuyan a aumentar el conocimiento disponible acerca de las relaciones entre perfiles de expresión génica y los niveles de AO en el agua de mar.

CLUSTER	INSTANCIAS	DISTANCIA
1	NORM_MGC_c10101, NORM_MGC_c16258, NORM_MGC_c2047	$3,822623 \times 10^{-4}$
66	NORM_MGC_c4296, NORM_MGT_c1212, NORM_MGT_c2042, NORM_MGT_c3815, NORM_MGT_c4464, NORM_MGT_c449, NORM_MGT_c4726, NORM_MGT_c1791, NORM_MGT_c81	$9,556574 \times 10^{-4}$
146	NORM_MGT_c1430, NORM_MGT_c3253, NORM_MGT_c3542, NORM_MGT_c934	$9,556574 \times 10^{-4}$
202	NORM_MGC_c5230, NORM_MGC_c5738, NORM_MGC_c5903, NORM_MGC_c7942, NORM_MGT_c3633	$9,556574 \times 10^{-4}$
1763	NORM_MGC_c2980, NORM_MGC_c3847, NORM_MGC_c5676, NORM_MGT_c2521	$2,675840 \times 10^{-3}$
2711	NORM_MGC_c13824, NORM_MGC_c5364, NORM_MGT_c2018	$1,529051 \times 10^{-3}$
3335	NORM_MGC_c1058, NORM_MGC_c1954, NORM_MGC_c9117, NORM_MGC_c15854, NORM_MGC_c6621, NORM_MGC_c2291	$3,822629 \times 10^{-3}$
4166	NORM_MGC_c1201, NORM_MGC_c145, NORM_MGC_c16053, NORM_MGC_c1681, NORM_MGC_c1725, NORM_MGC_c1945, NORM_MGC_c5915, NORM_MGC_c8636, NORM_MGT_c22, NORM_MGT_c4806	$4,510703 \times 10^{-3}$
5931	NORM_MGC_c1272, NORM_MGC_c13971, NORM_MGC_c4679, NORM_MGT_c118, NORM_MGT_c527	$5,691471 \times 10^{-3}$
7300	NORM_MGC_c14294, NORM_MGC_c2281, NORM_MGC_c33, NORM_MGC_c73, NORM_MGC_c9690, NORM_MGT_c141, NORM_MGT_c4677, NORM_MGT_c527	$2,064220 \times 10^{-2}$

Cuadro 4.1: Tabla de distancias de cada *cluster*

CLUSTER	NOMBRE	IMPORTANCIA
1	DNA RECOMBINATION	18,18 %
	DNA REPAIR	18,18 %
	MITOTIC CHROMOSOME CONDENSATION	18,18 %
	CELL DIVISION	18,18 %
	SISTER CHROMATID COHESION	18,18 %
	OXIDATION-REDUCTION PROCESS	9,09 %
66	TRANSLATION	20 %
	RIBOSOME BIOGENESIS	60 %
	TRANSLATIONAL ELONGATION	20 %
146	POSITIVE REGULATION OF ACTIN FILAMENT POLYMERIZATION	11,11 %
	RESPONSE TO UNFOLDED PROTEIN	11,11 %
	PROTEIN FOLDING	44,44 %
	PROTEIN PEPTIDYL-PROLYL ISOMERIZATION	22,22 %
	NUCLEAR TRANSPORT	11,11 %
202	NUCLEOSOME ASSEMBLY	10 %
	NEGATIVE REGULATION OF TRANSCRIPTION FROM RNA POLYMERASE II PROMOTER	10 %
	REGULATION OF TRANSCRIPTION, DNA-DEPENDENT	10 %
	STEROID HORMONE MEDIATED SIGNALING PATHWAY	10 %
	APOPTOTIC PROCESS	10 %
	MULTICELLULAR ORGANISMAL DEVELOPMENT	40 %
	HISTONE H2A MONOUBIQUITINATION	10 %

Cuadro 4.2: Importancia de los atributos de cada *Cluster*

Capítulo 5

Conclusiones y trabajo futuro

Índice general

5.1. Conclusiones	111
5.1.1. Objetivos	111
5.1.2. Comparación con trabajos relacionados	112
5.1.3. Metodología	113
5.1.4. Problemas encontrados	114
5.2. Trabajo futuro	114

NEste capítulo se expondrán las conclusiones obtenidas a la fecha de la finalización del proyecto y una sección en la que se evaluarán las posibles extensiones al mismo.

5.1. Conclusiones

Se analizarán las conclusiones extraídas de las dos vertientes del proyecto.

5.1.1. Objetivos

Los objetivos planteados al comienzo del presente proyecto se han satisfecho en su totalidad. Además, los múltiples cambios en los requisitos y las nuevas funcionalidades añadidas se han completado sin excepción. La parte del desarrollo web se encuentra preparada para comenzar la fase de producción y así sustituir

a la antigua. Con respecto al análisis de los datos genéticos, se han enviado los prometedores resultados obtenidos a los responsables del grupo CHROMEVOL para su interpretación biológica, de la cual se está a la espera. Las conclusiones que se extraigan a partir de estos datos contribuirán a la producción de test moleculares para la detección y evaluación de los efectos genotóxicos del ácido okadaico.

5.1.2. Comparación con trabajos relacionados

Con respecto a la parte del desarrollo web, y tomando como base el listado de características mejorables de la antigua, se realizará un resumen de las razones por las que el presente proyecto supera a la herramienta anterior.

- Tecnología mas moderna: Python es el lenguaje preferido de los actuales investigadores en bioinformática, como lo era Perl hace años.
- Escalabilidad añadida: Se ha seguido el patrón de diseño MVC, además de abstraer la tecnología de la base de datos. Se ha puesto especial hincapié en mejorar las pésimas prácticas de programación de la web antigua. Se ha establecido una jerarquía entre los HTML, permitiendo reutilizar la mayoría de este código. Se han incluido comentarios en todo lo que podría ocasionar dudas a posibles desarrolladores futuros, además de que el diseño modular implementado permitirá añadir nuevas funcionalidades de forma sencilla.
- Contenidos actualizados: Los enlaces que integran las herramientas bioinformáticas funcionan tal y como se espera, realizando las acciones que el cliente expuso en sus peticiones.
- Acceso eficiente: Se han añadido diversas optimizaciones con respecto a la web antigua; por ejemplo, el acceso a los contenidos de la web es ahora paginado, pudiéndose ordenar por diversos campos y retroceder y avanzar por los resultados mostrados. Además, se han implementado mejoras en la eficiencia de las consultas a la base de datos.
- Corrección de errores: Se han corregido errores de la web antigua para que, por ejemplo, no haya posibilidad de dejar campos en blanco a la hora de llamar a BLAST.

- Interfaz renovada: La diferencia entre la calidad de la interfaz gráfica de las dos webs es notable. El renovado diseño de la nueva deja claro el esfuerzo invertido en el apartado gráfico. La sensación de navegar por la nueva web es propia de las páginas más modernas, y el nuevo logo refleja mucho mejor el objetivo de la misma, aportando más dinamismo.
- Funcionalidades añadidas: El presente proyecto incluye tanto las características básicas de la antigua web como las siguientes:
 - Búsqueda por cualquier campo de las tablas.
 - Navegación paginada.
 - Posibilidad de ordenar las filas de las tablas.
 - Sistema de autentificación.
 - Gestión de usuarios y sus permisos
 - Gestión de *Contigs*, *Reads* y *Clusters*.

Por el contrario, debido a que no existen intentos previos para explorar los datos del grupo CHROMEVOL, la parte de análisis de datos del presente proyecto no incluye ninguna comparativa.

5.1.3. Metodología

La metodología planteada en el anteproyecto no se ha seguido en absoluto, debido a que al comenzar el proyecto se decidió cambiarla por una metodología ágil. Esta decisión se confirma como completamente acertada porque, si bien inicialmente el cliente requería una web sencilla que simplemente mostrara unas tablas, mas tarde sus pretensiones fueron creciendo gradualmente hasta llegar a lo que es actualmente: una completa herramienta para la gestión de datos biológicos. Si se llega a continuar con la metodología en espiral planteada inicialmente, el costo de todos los cambios en los requisitos hubiera sido realmente grave. Aún así, a pesar de todo lo que prometen las metodologías ágiles, hubo un sobrecoste debido a estos continuos cambios. El proyecto habría sido mucho más sencillo, menos costoso y con un entregable idéntico si se hubieran tenido los requisitos claros desde el primer momento.

5.1.4. Problemas encontrados

El principal problema de este proyecto ha sido la falta de conocimiento por parte del autor de la presente memoria de lo que se esperaba de él desde un primer momento.

Con respecto a la web, el hecho de que el propietario del producto estuviera en EEUU dificultó en gran medida la obtención de requisitos. Los constantes cambios de los mismos, muchos de ellos no plasmados en esta memoria por simplicidad, han llevado a replantear el diseño en multitud de ocasiones. Además, el hecho de tener que haber extraído la estructura de una base de datos existente complicó las cosas debido a que no se tenía la certeza de estar trabajando sobre las tablas correctas hasta bien avanzado el proyecto. Ésto, unido al hecho de no comprender completamente el significado biológico de las tablas ni las relaciones entre ellas desde el primer momento llevó a replantear su estructura en multitud de ocasiones. La baja calidad del código de la web original hizo perder mucho tiempo intentando entender su funcionamiento, debido a que había funcionalidades que a simple vista no deberían funcionar y que sin embargo sí lo hacían.

Con respecto al análisis de datos, el hecho de empezar el mismo sin una estructura de los datos bien definida y sin comprender el significado y las relaciones entre las entidades lo dificultó mucho. Sin embargo, el principal problema de la parte analítica fue el poco tiempo que se le pudo dedicar debido a la mayor importancia que se le dio a la otra parte del proyecto, la de la web.

Hay que destacar, además, el tiempo invertido en conseguir una base suficiente de conocimientos biológicos que permitiese analizar el problema con conocimiento de causa. La limitada formación en biología con la el autor del presente proyecto partía supuso un importante impedimento, sobre todo en los inicios del mismo.

5.2. Trabajo futuro

Esta sección contiene algunas de las posibles formas de completar el trabajo realizado en el presente proyecto.

Con respecto a la herramienta web hay que destacar que se ha abierto la posibilidad de expansión de la misma de muchas formas. Las características adicionales que se podrían implementar y en las que el cliente estaría interesado son

las siguientes:

- Realización de un *pipeline* integrado con la aplicación, para llevar a cabo un proceso de *gene enrichment* basado en la herramienta TProfiler. Este proceso es una técnica utilizada para interpretar conjuntos de genes usando el sistema de clasificación de *Gene Ontology*, en el cual se les asigna una caracterización predefinida dependiendo de sus características funcionales. Esto sería de gran utilidad para el grupo de investigación debido a que actualmente no dispone de un medio para automatizar la caracterización de sus datos.
- Integración de nuevas fuentes de información como AmiGO o KEGG. Esto permitiría a los investigadores que utilizasen la herramienta una mayor rapidez y eficiencia en el acceso a contenido adicional sobre los datos genéticos que contiene.
- Inclusión de la herramienta GOblet [32] como parte del framework de bioinformática que se está desarrollando. Ésta es una aplicación web que permite anotar secuencias anónimas con los términos de GO, por lo que sería útil para actualizar las anotaciones existentes y permitir a los usuarios anotar las suyas propias.
- Integrar la herramienta Trinotate [33] de forma que se ejecute periódicamente en el servidor con el objetivo de actualizar los *Accessions* de BLAST para los *Contigs*.

Con respecto a la parte del análisis de datos, las posibilidades del trabajo futuro son inmensas. Los resultados presentados en este proyecto son realmente buenos para el limitado tiempo del que se disponía, aunque la calidad del análisis se podría aumentar. La posibilidad de utilizar los datos disponibles en la base de datos de AmiGO y KEGG, con sus relaciones y jerarquías entre términos, permitiría relacionar instancias cuyos atributos se encuentran dentro de la misma ruta metabólica, entre otras ventajas.

Además, se podría desarrollar una base de datos que indique las relaciones entre los diferentes *Unigenes* e incluso *Clusters*. Ésta se integraría en la web realizada en la primera parte del proyecto, permitiendo: seleccionar instancias

y mostrar el grado de relación entre ellas, elaborar diagramas completos de las rutas metabólicas en las que actúan, establecer umbrales, etc.

Cabe destacar que la parte de análisis de los datos genéticos continuará probablemente en el futuro. Este inicio ha sentado las bases de lo que podría ser una Tesis Doctoral en el área de la bioinformática. El interés tanto científico como económico de los resultados que se puedan extraer de posteriores análisis parecen asegurar alguna financiación para su realización en un futuro no muy lejano.

Los resultados y las conclusiones extraídos del análisis de datos biológicos serán enviados para su revisión y posterior publicación en diversas revistas científicas de alto impacto.

Apéndices

Apéndice A

Manual de usuario

Este manual contiene todo lo necesario para que un usuario pueda sacar todo el partido que la herramienta ofrece.

A.1. Requisitos *Hardware* y *Software*

Al tratarse de una aplicación web sólo es necesario disponer de una conexión a Internet y un navegador web. Estos requisitos hacen que cualquier persona pueda acceder al portal sin apenas limitaciones espaciales o temporales.

A.2. Descripción de pantallas visibles al usuario estándar

En este punto se describen todas las pantallas con las que el usuario podrá interactuar y las transiciones entre las mismas.

La Figura A.1 muestra el panel de navegación presente en todas las pantallas de la web. Sus elementos son los siguientes:

- **HOME**: seleccionado este botón se redirige a la pantalla inicial.
- **BLAST**: lleva a la pantalla para llamar a la herramienta BLAST.
- **BROWSE**: seleccionando esta pestaña se redirige a la vista de los *Contigs* de la base de datos.
- **EXPRESSION** lleva a una vista de los *Unigenes* de la base de datos.

- **About** redirige a la página del grupo CHROMEVOL.



Figura A.1: Panel de navegación

A.2.1. Pantalla inicial

La Figura A.2 es la pantalla que verá el usuario al entrar en la web. Se puede observar los elementos principales comunes a todas las páginas de la web. Contiene una breve descripción de la web y una imagen de un *microarray* (superficie sólida a la cual se une una colección de fragmentos de ADN). Además incluye una serie de elementos que se repetirán en todas las pantallas que están marcados en rojo. Éstos son:

- **1:** Logo de la web.
- **2:** Barra de navegación, indicando la pestaña actualmente seleccionada.
- **3:** Logo del grupo CHROMEVOL, con un enlace a la página del mismo.
- **4:** Logo del grupo RNASA, con un enlace a la página del mismo.
- **5:** Logo de la web, con un enlace a la página principal.
- **6:** Selector de idioma; Inglés actualmente seleccionado.
- **7:** Pestaña de *About*, lleva a la página oficial del grupo CHROMEVOL.

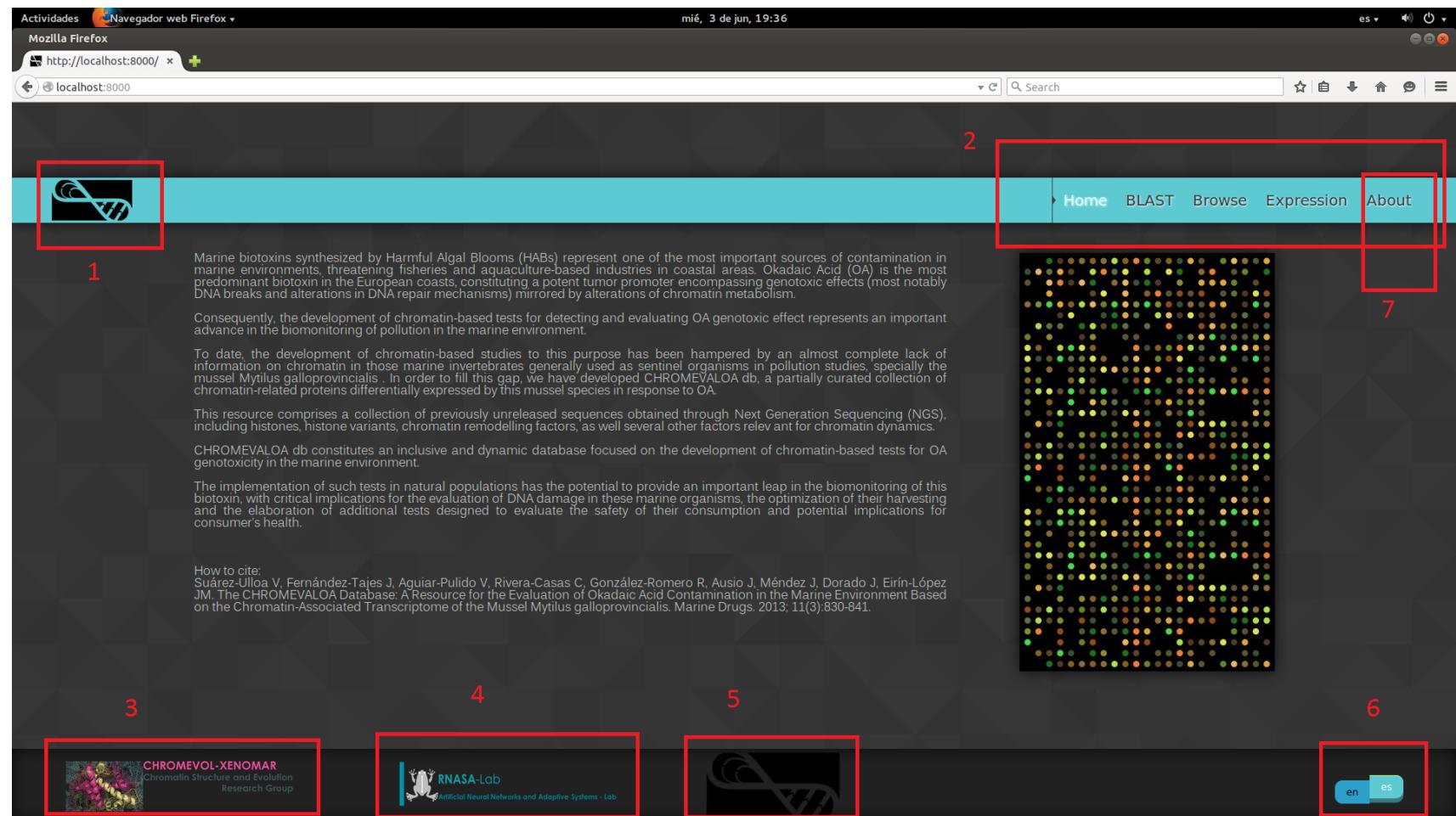


Figura A.2: Pantalla inicial

A.2.2. *Browse*

La Figura A.3 muestra los campos más importantes de la tabla *Contigs*. Ésta permite ordenar y realizar búsquedas por todos los campos. Además de estar enlazada con diversas páginas. Los elementos marcados en rojo representan lo siguiente:

- **1:** Botón para descargar la tabla completa. (ver Figura A.4 como ejemplo).
- **2:** Barra de búsqueda, que permite seleccionar campo e introducir el texto de búsqueda. Puede verse un ejemplo en la figura A.5
- **3:** Botón de editar, lleva a la página de administrador para ese *Contig*.
- **4:** ID del *Contig*, lleva a la página de detalles de la Figura A.6.
- **5:** *Accession* del *Contig*, realiza una búsqueda en la página del NCBI.
- **6:** Al seleccionar la cabecera de una columna se ordenará por la misma. Al volver a seleccionarla se hará en el orden inverso.
- **7:** Navegación por la tabla, permite ir para delante y para detrás.
- **8:** Información sobre la posición, indica qué parte de la base de datos total se está mostrando.

A. Manual de usuario

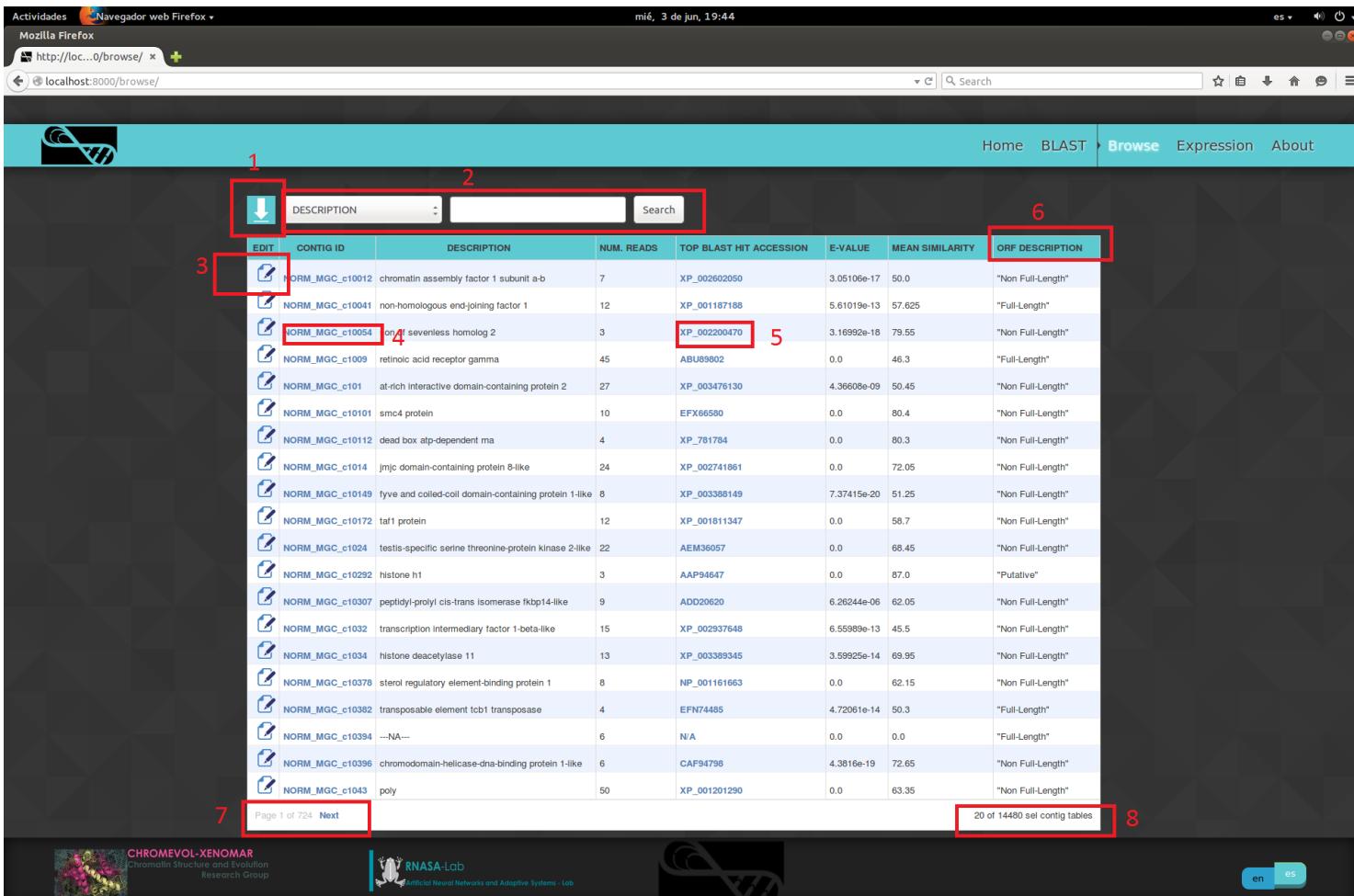


Figura A.3: Pantalla Browse: visualizar los *Contigs*

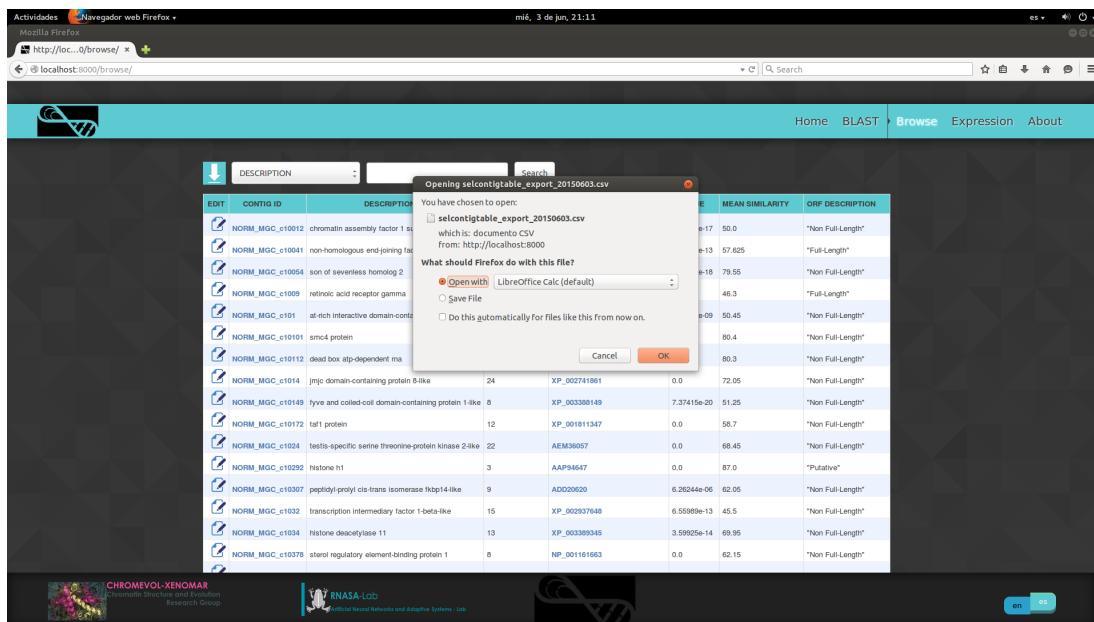
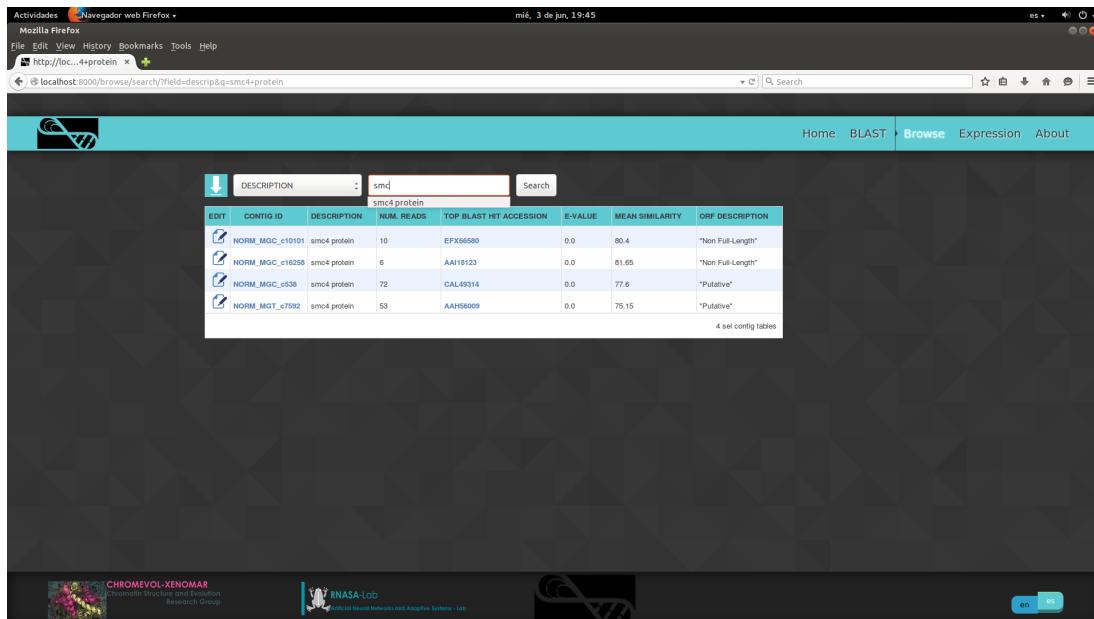


Figura A.4: Pantalla de descarga de la tabla completa

A.2.2.1. Búsqueda de *Contigs*

Figura A.5: Pantalla de los resultados de la búsqueda de *Contigs*

En la Figura A.4 puede verse un ejemplo de los resultados de una búsqueda de *Contigs* por su descripción.

A.2.2.2. Detalles del *Contig*

La Figura A.6 muestra la secuencia del *Contig* seleccionado previamente. Además de descargarse puede visualizarse su alineamiento con los *Reads* que lo conforman seleccionando la opción de Jalview. Ésta lleva a la Figura 3.28.

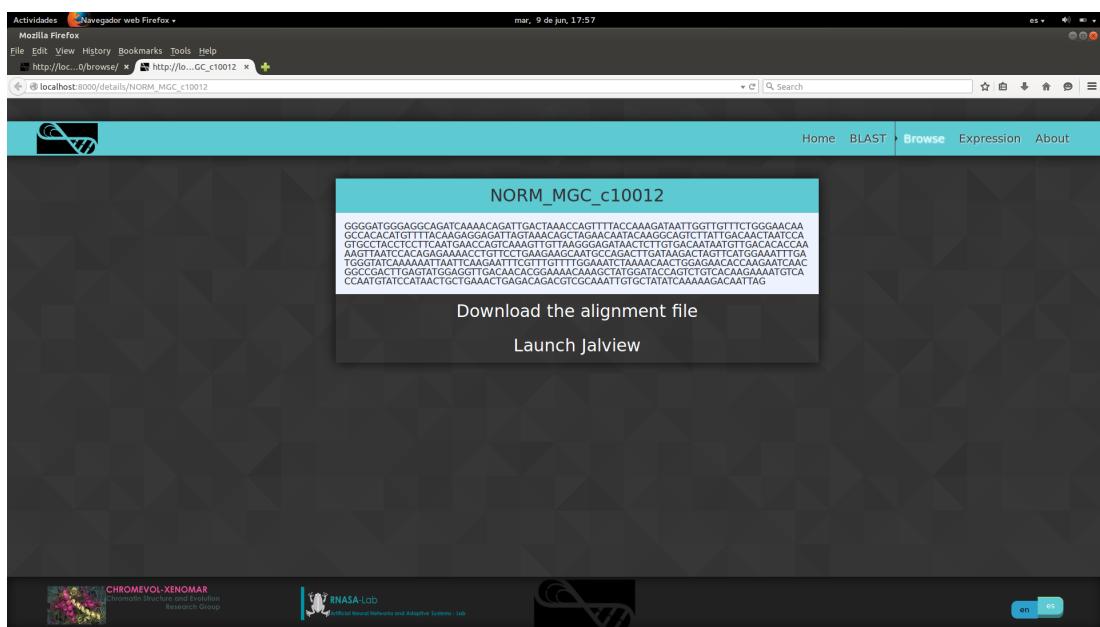


Figura A.6: Pantalla que muestra la secuencia del *Contig* seleccionado

A.2.3. Jalview

La Figura A.7 muestra la herramienta Jalview descargada y cargada con el fichero de alineamiento descargado previamente.

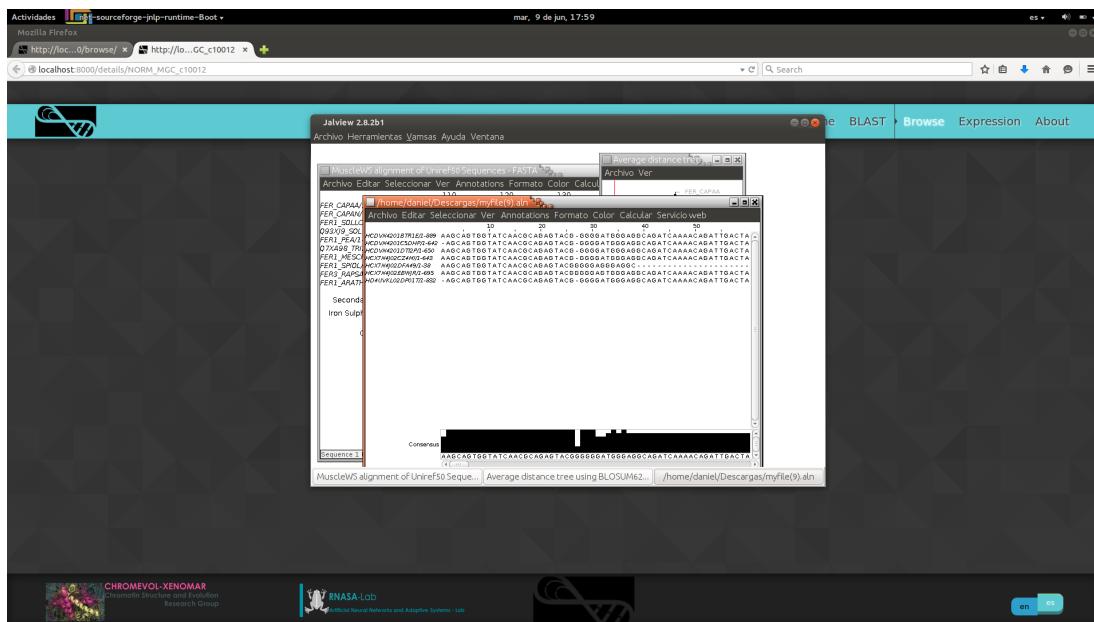


Figura A.7: Pantalla de visualización de Jalview

A.2.4. Expression

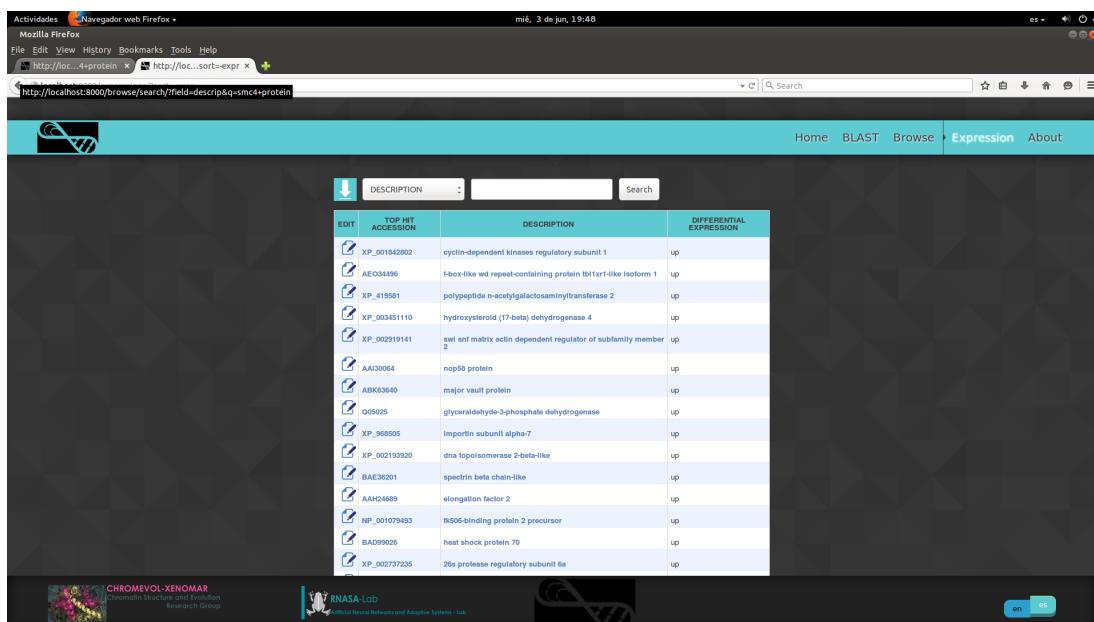


Figura A.8: Pantalla Expression: visualizar la expresión de los Unigenes

La Figura A.8 muestra algunos campos de la tabla *Unigenes*. Además de permitir las mismas operaciones que la pestaña anterior, al seleccionar la descripción de alguno de ellos se realiza una búsqueda en todos los *Contigs* con esa misma descripción.

A.2.5. BLAST

La Figura A.10 muestra la pantalla de selección de los parámetros de configuración de la llamada a BLAST. Los elementos pueden desglosarse en los siguientes:

- **1:** Entrada de la secuencia problema.
- **2:** Fichero a subir en vez de usar la caja de texto de arriba.
- **3:** Seleccionar el formato de BLAST:
 - BLASTN: Acepta una secuencia de nucleótidos.
 - TBLASTN: Acepta una secuencia de proteínas.
 - TBLASTX: Acepta una secuencia de nucleótidos traducidos.
- **4:** Formato del fichero de entrada:
 - FASTA
 - EMBL
 - GENBANK
 - RAW
- **5:** Base de datos contra la que comparar:
 - NORM_TREATED: Población de mejillones expuestos al AO.
 - NORM_CONTROL: Población de moluscos de control.
 - NORM_LIBS: Usar las dos a la vez.
- **6:** Confianza de la comparación.
- **7:** Máximo número de resultados (Por orden de confianza).
- **8:** Botón para realizar la consulta.

- **9:** Ayuda sobre el tipo de BLAST a utilizar, enlaza con la página mostrada en la Figura A.11.
- **6:** Ayuda sobre la base de datos a utilizar, enlaza con la página mostrada en la Figura A.12.

La Figura A.9 Muestra la información obtenida de la llamada a BLAST. Puede ser realmente diferente en función de los parámetros de entrada.

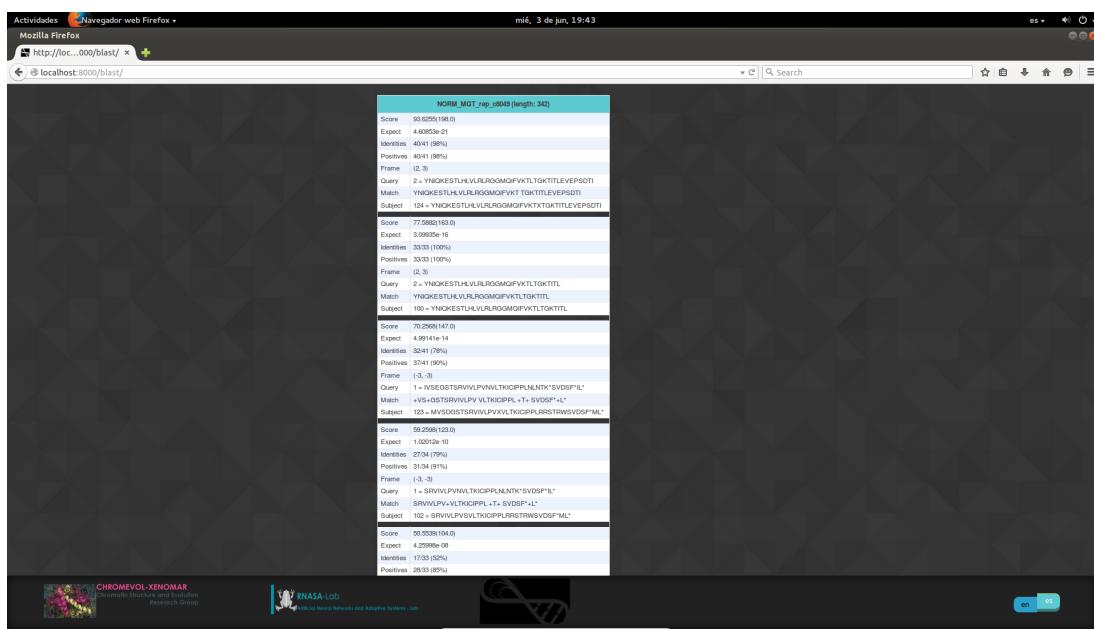


Figura A.9: Pantalla de los resultados de BLAST

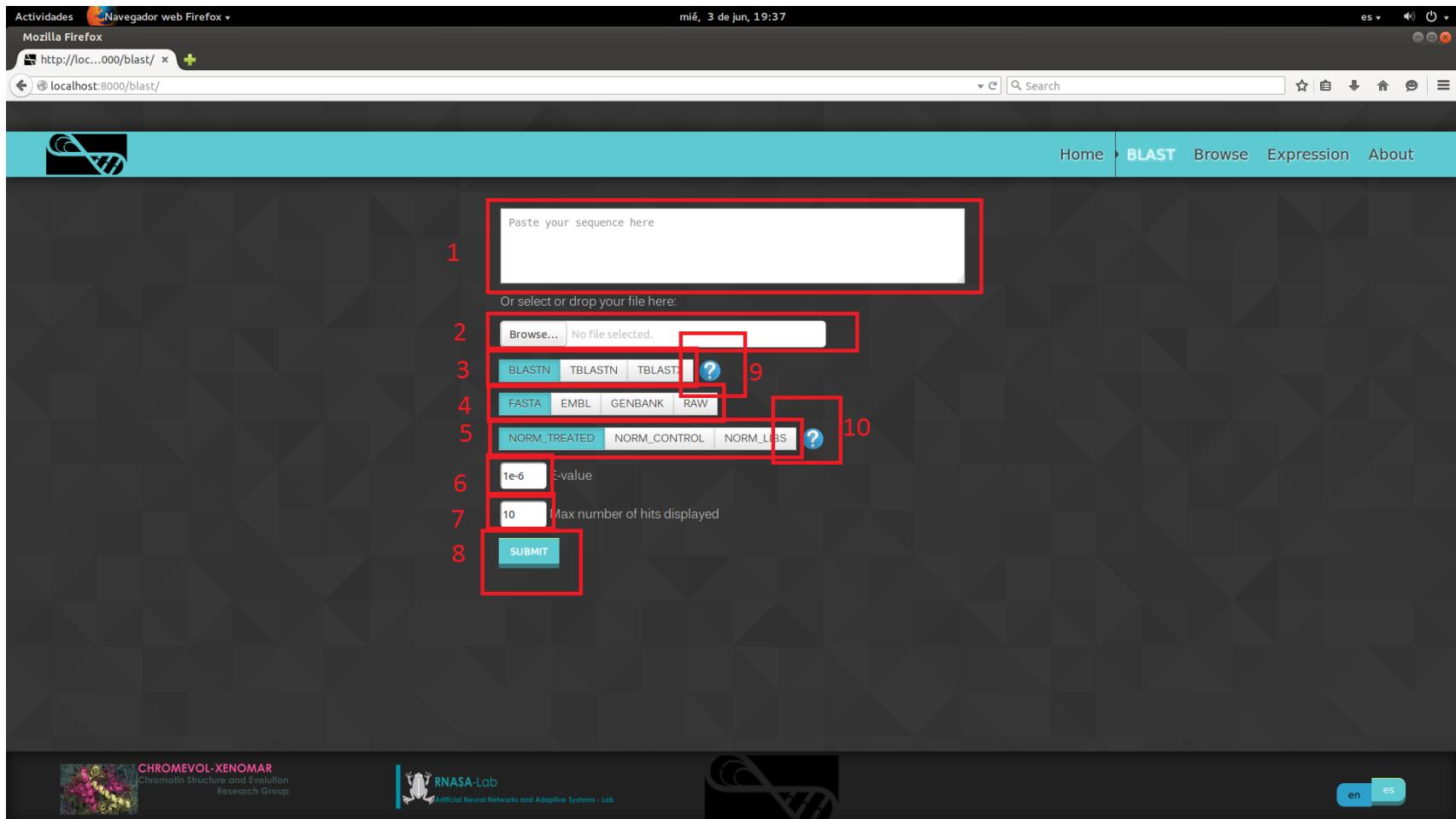


Figura A.10: Pantalla de configuración de BLAST

A.2.5.1. *Program Help*

La Figura A.11 Muestra ayuda acerca del tipo de BLAST a utilizar en función de la secuencia de entrada. Cada tipo de BLAST acepta un formato distinto de *query*.

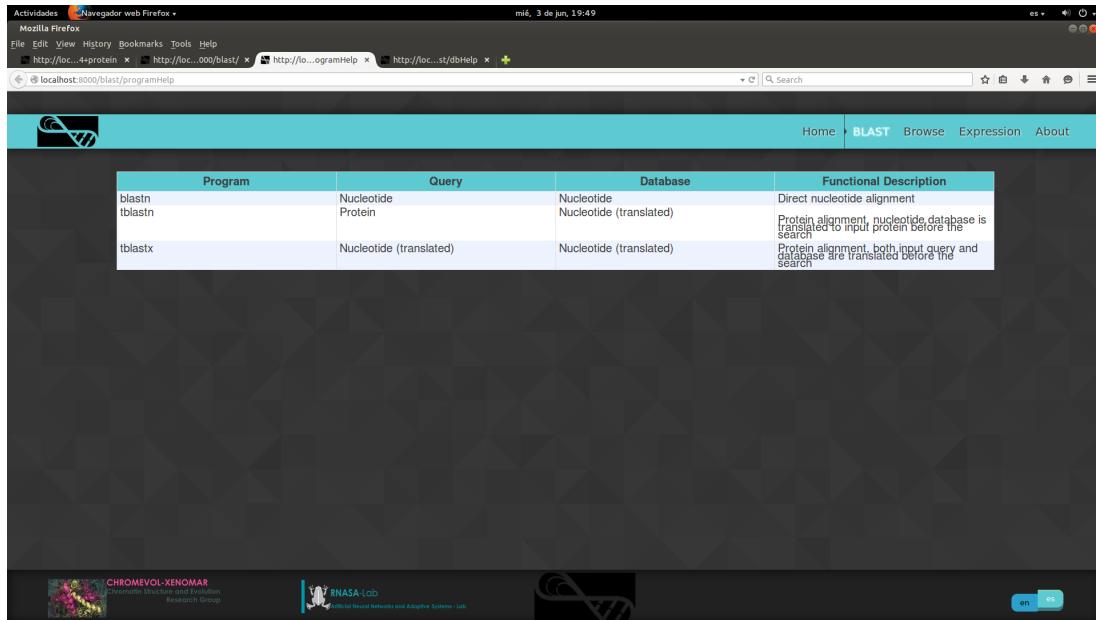


Figura A.11: Pantalla de ayuda sobre el tipo de BLAST a utilizar

A.2.5.2. *Database Help*

La Figura A.12 Muestra ayuda acerca de los contenidos de la base de datos a la hora de llamar a BLAST.

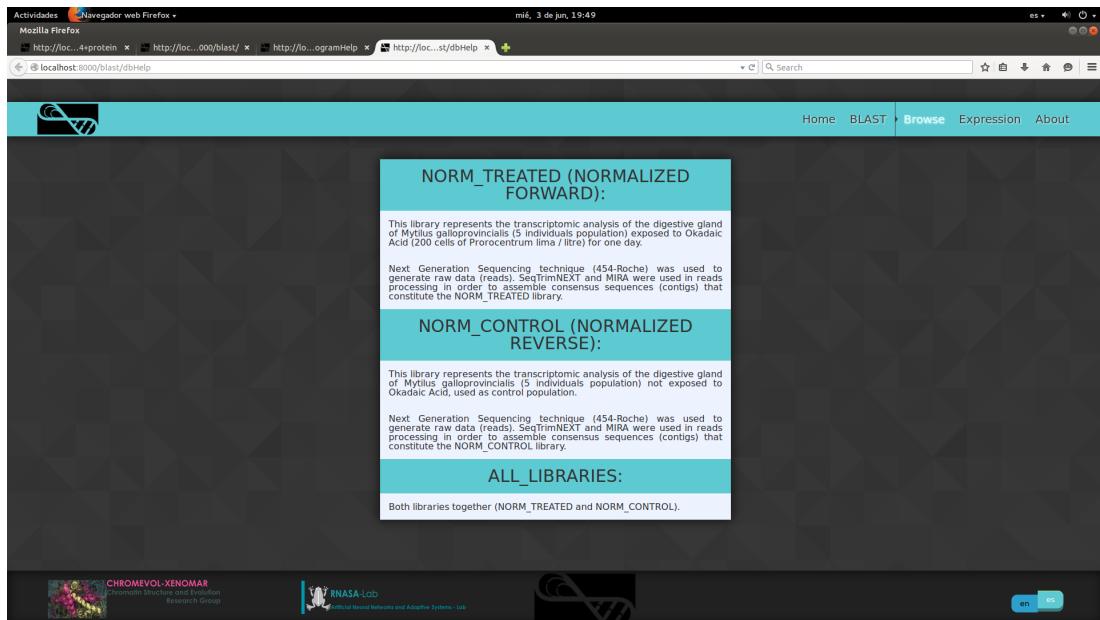


Figura A.12: Pantalla de ayuda sobre los contenidos de cada base de datos

A.2.6. Errores

Las Figuras A.13 y A.14 muestran ejemplos de cómo se verían los errores que podría visualizar el usuario.



Figura A.13: Error 404



Figura A.14: Error 500

A.3. Descripción de pantallas visibles a los administradores

En este punto se describen todas las pantallas a las que solamente un usuario con los permisos apropiados tendrá acceso.

A.3.1. Pantalla de *login*

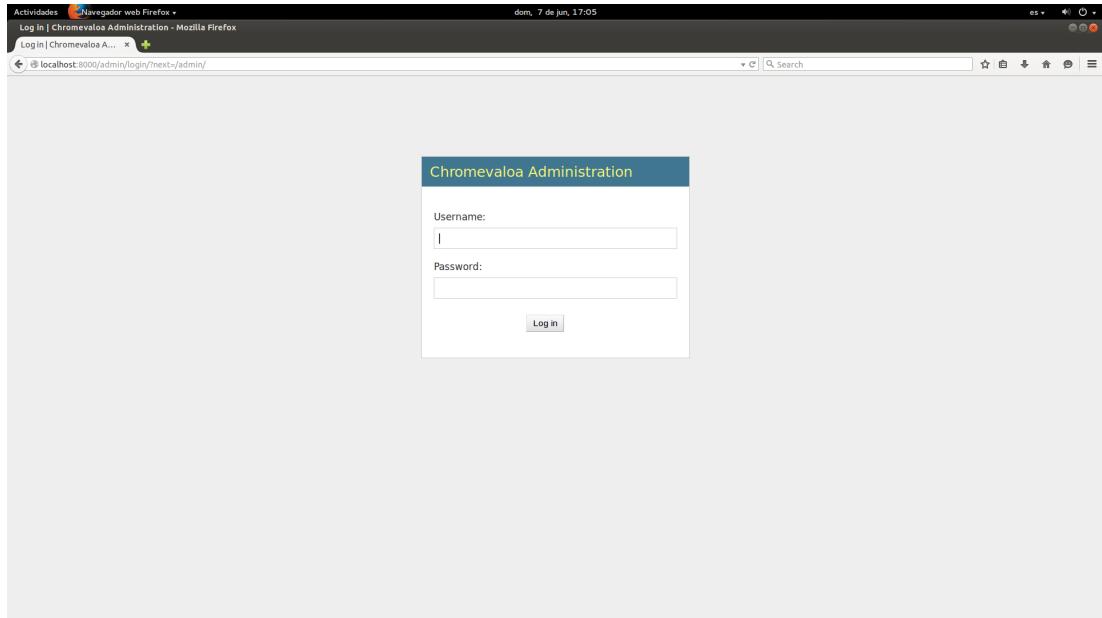


Figura A.15: Pantalla de identificación

La Figura A.15 muestra la primera pantalla que un usuario ve al entrar en el menú de administrador: se pide un usuario y contraseña válidos para poder navegar por el menú de administrador.

A.3.2. Pantalla de administración inicial

La Figura A.16 muestra la pantalla de inicio del menú administrador, en ella hay accesos directos a todas las entidades relevantes que el administrador puede editar.

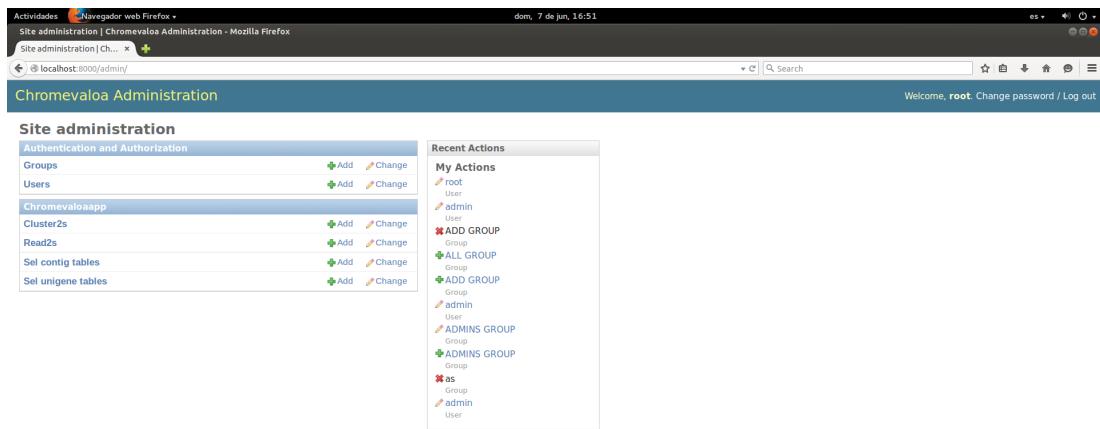


Figura A.16: Pantalla inicial de administrador

A.3.3. Pantalla de gestión de usuarios

La Figura A.17 muestra lo que el administrador ve al seleccionar la opción de gestión de usuarios: una lista de todos los usuarios, sobre la que puede realizar búsquedas, o seleccionarlos para editarlos, además de añadirlos. Ésta ventana y las correspondientes para el resto de entidades consta de las siguientes partes:

- **1:** Barra de búsqueda.
- **2:** Acción a realizar en las instancias seleccionadas.
- **3:** Instancias seleccionables de la clase en cuestión.
- **4:** Botón de añadir usuario.

- 5: Filtro de usuarios.
- 7: Panel de navegación.

Actividades Navegador web Firefox - Select user to change | Chromeveloa Administration - Mozilla Firefox

dom, 7 de jun, 16:52

Select user to change | ...

localhost:8000/admin/auth/user/

Search

Welcome, root. Change password / Log out

Chromeveloa Administration

Home > Authentication and Authorization > Users

Select user to change

Action: Search Go 0 of 2 selected

Username	Email address	First name	Last name	Staff status
<input type="checkbox"/> admin	da@ads.com			<input checked="" type="checkbox"/>
<input type="checkbox"/> root				<input checked="" type="checkbox"/>

2 users

Add user

Filter

By staff status

- All
- Yes
- No

By superuser status

- All
- Yes
- No

By active

- All
- Yes
- No

By groups

- All
- ADMINS GROUP
- ALL GROUP

Figura A.17: Pantalla de gestión de usuarios

Actividades Navegador web Firefox - Change user | Chromeveloa Administration - Mozilla Firefox

dom, 7 de jun, 16:54

File Edit View History Bookmarks Tools Help

Change user | Chromeveloa Administration

localhost:8000/admin/auth/user/1/

Welcome, root. Change password / Log out

Chromeveloa Administration

Home > Authentication and Authorization > Users > root

Change user

History

Username: root

Required. 30 characters or fewer. Letters, digits and @/_/-_ only.

Password: pbkdf2_sha256 iterations: 15000 salt: 22at3***** hash: pVAL4+*****

algorithm: pbkdf2_sha256 iterations: 15000 salt: 22at3***** hash: pVAL4+*****

Raw passwords are not stored, so there's no way to see this user's password, but you can change the password using this form.

Personal info

First name:

Last name:

Email address: da@ads.com

Permissions

Active

Designates whether this user should be treated as active. Uncheck this instead of deleting accounts.

Staff status

Designates whether the user can log into this admin site.

Superuser status

Designates that this user has all permissions without explicitly assigning them.

The groups this user belongs to. A user will get all permissions granted to each of his/her group. Hold down "Control", or "Command" on a Mac, to select more than one.

Groups:

Available groups

Chosen group

ADMINS GROUP

ALL GROUP

Choose all

Remove all

Figura A.18: Pantalla de editar usuarios 1

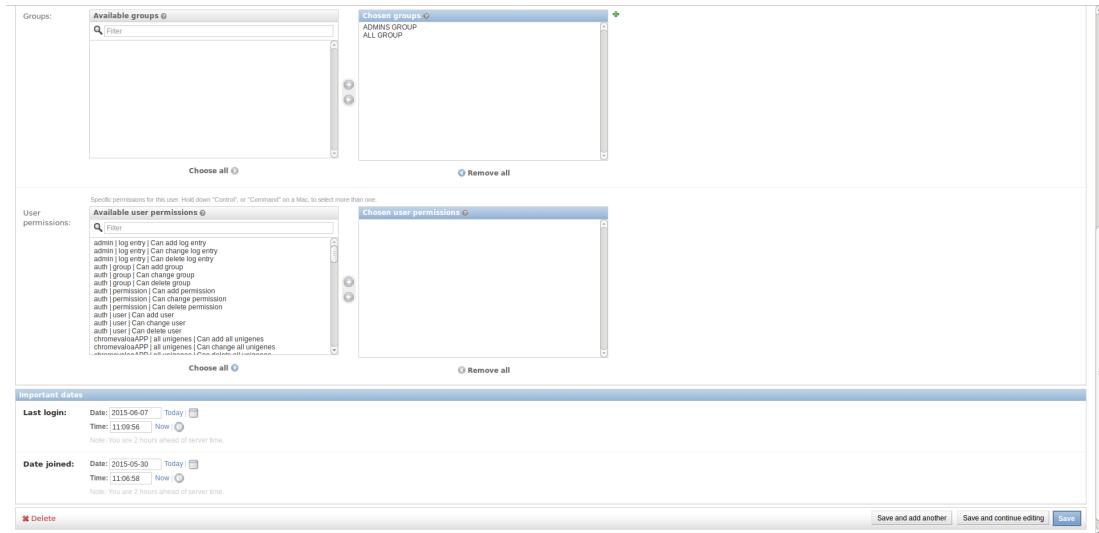


Figura A.19: Pantalla de editar de usuarios 2

Las Figuras A.18 y A.19 muestran todos los campos que un administrador puede editar en un usuario. La contraseña se muestra encriptada y para modificarla tiene que seleccionarla y se le redirige a la ventana mostrada en la Figura A.20

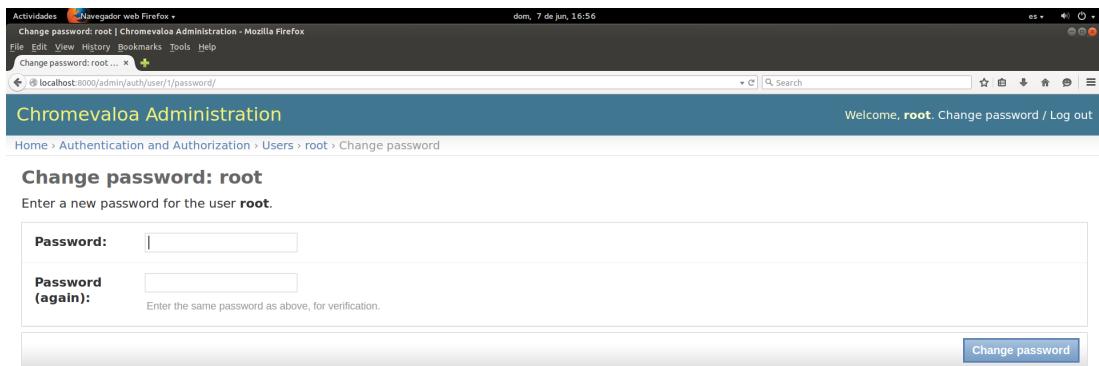


Figura A.20: Pantalla de cambio de contraseña de un usuario

A.3.4. Pantalla de gestión de grupos

La Figura A.21 muestra lo que el administrador ve al seleccionar la opción de gestión de grupos: una lista de todos los grupos, sobre la que puede realizar

búsquedas, o seleccionarlos para editarlos, además de añadirlos.



Figura A.21: Pantalla de gestión de grupos

Al seleccionar uno, se redirige a la pantalla mostrada en la Figura A.22

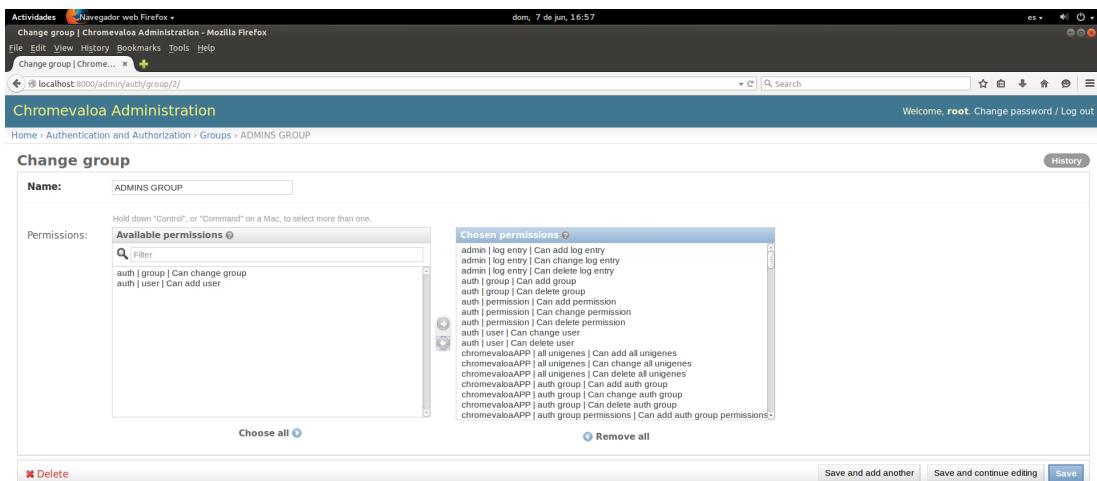


Figura A.22: Pantalla de editar grupos

A.3.5. Pantalla de gestión de *Reads*

La Figura A.23 muestra lo que el administrador ve al seleccionar la opción de gestión de *Reads*: Una lista de todos los *Reads*, sobre la que puede realizar búsquedas, editarlos o añadirlos.

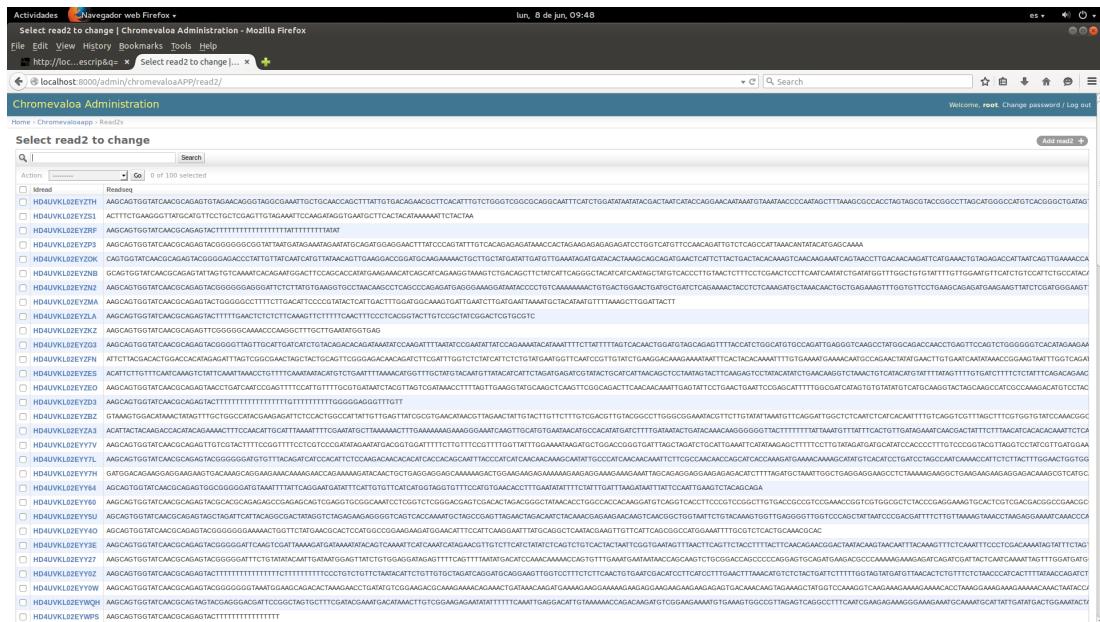


Figura A.23: Pantalla de gestión de *Reads*

Al seleccionar uno, se redirige a la pantalla mostrada en la Figura A.24

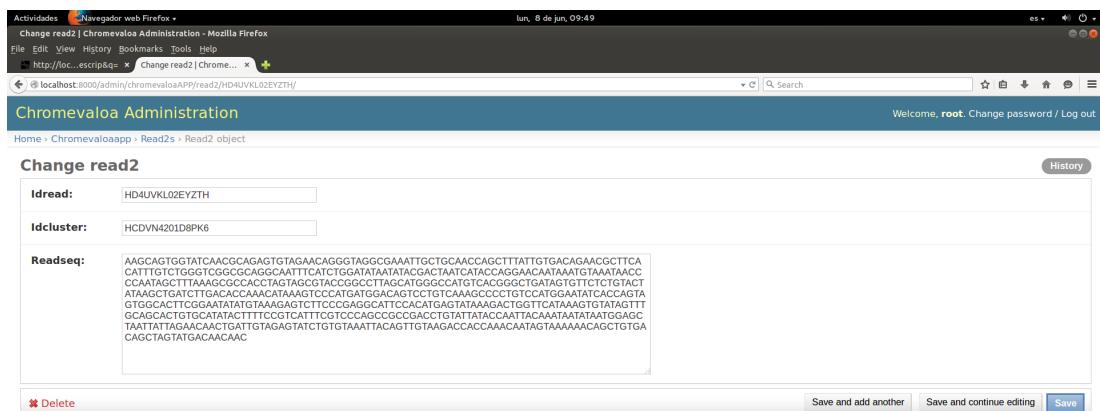


Figura A.24: Pantalla de editar *Reads*

Apéndice B

Guía de instalación

Esta guía contiene todos los pasos a seguir para la instalación y ejecución de la herramienta web en un sistema operativo Linux.

B.1. Instalación

Se deberán instalar las siguientes herramientas siguiendo las instrucciones indicadas. En el caso de adjuntarse un enlace a una web, hay seguir los pasos indicados en la misma. Todos los enlaces adjuntados son plenamente funcionales a fecha de entrega de la memoria. En caso de adjuntarse una lista de comandos, hay que ejecutarlos en un terminal instalando todas las dependencias necesarias. Se deberán seguir los pasos en el orden indicado, finalizando la instalación de una herramienta antes de proceder con la siguiente.

- **Python 2.7.** Esta herramienta se puede descargar e instalar siguiendo las instrucciones de la siguiente página:

<http://heliumhq.com/docs/installing-python-2.7.5-on-ubuntu>

- **MySQL Server.** Esta herramienta se puede descargar e instalar siguiendo las instrucciones de la siguiente página:

<https://help.ubuntu.com/12.04/serverguide/mysql.html>

- **pip.** Esta herramienta se puede instalar ejecutando los siguientes comandos:

```
sudo apt-get install python-pip python-dev build-essential  
sudo pip install --upgrade pip
```

```
sudo pip install --upgrade virtualenv
```

- **MySQL Python.** Esta herramienta se puede descargar e instalar ejecutando los siguientes comandos:

```
pip install MySQL-python
sudo apt-get install python-mysqldb
```

- **Django.** Esta herramienta se puede descargar e instalar ejecutando los siguientes comandos:

```
pip install -U Django
pip install django --upgrade
```

- **Extensiones de Django.** Esta herramienta se puede descargar e instalar ejecutando los siguientes comandos:

```
pip install django-queryset-csv
pip install django-tables2
```

- **BLAST.** Esta herramienta se puede descargar e instalar ejecutando los siguientes comandos:

```
apt-get install ncbi-blast+
```

- **Clustal Omega.** Esta herramienta se puede descargar e instalar ejecutando los siguientes comandos:

```
sudo apt-get install clustalo
```

B.2. Ejecución

En caso de haber completado la totalidad de requisitos, abrir un terminal en la carpeta raíz del proyecto y ejecutar

```
python manage.py runserver --insecure
```

Ésto lanzará el servidor en el puerto 8000 de localhost.

Apéndice C

Glosario de acrónimos

ADN *Ácido Desoxirribonucleico.*

API *Application Programming Interface.*

ARN *Ácido Ribonucleico.*

AO *Ácido Okadaico.*

BioRG *Bioinformatics Research Group.*

BLAST *Basic Local Alignment Search Tool.*

CGI *Common Gateway Interface.*

CHROMEVALOA *CHROMatin EVALuation of Okadaic Acid.*

CHROMEVOL *Chromatin Structure and Evolution Research Group.*

CSS *Cascade Style Sheet.*

DRY *Don't Repeat Yourself.*

DSP *Diarrhetic Shellfish Poisoning.*

DW *Data Warehouse.*

E-R *Entity Relationship.*

FIU *Florida International University.*

GNU-GPL *GNU General Public License.*

HP *Hewlett-Packard.*

HTML *HyperText Markup Language.*

IBM *International Business Machines Corp.*

I18N *Internationalization.*

KEGG *Kyoto Encyclopedia of Genes and Genomes.*

MTV *Model Template View.*

MVC *Model View Controller.*

NCBI *National Center for Biotechnology Information.*

OMG *Object Management Group.*

PCA *Principal Component Analysis.*

SGBD Sistema de Gestión de Bases de Datos.

SOM *Self-Organizing Maps.*

UAT *User Acceptance Testing.*

UML *Unified Modeling Language.*

URL *Uniform Resource Locator.*

W3C *World Wide Web Consortium.*

XHTML *eXtensible HyperText Markup Language.*

XML *eXtensible Markup Language.*

Apéndice D

Glosario de términos

ADN: Ácido nucleico que contiene las instrucciones genéticas usadas en el desarrollo y funcionamiento de todos los organismos vivos conocidos, además de ser el responsable de su transmisión hereditaria.

ARN: Ácido nucleico formado por una cadena de ribonucleótidos. Está presente tanto en las células procariotas como en las eucariotas, siendo lineal y de hebra sencilla.

ARNm: Ácido ribonucleico que contiene la información genética procedente del ADN del núcleo celular y lo lleva a un ribosoma en el citoplasma, es decir, actúa como plantilla para la síntesis de las proteínas.

Biomonitorización: Utilización de métodos que permiten evaluar la concentración de compuestos químicos o de sus metabolitos en muestras biológicas.

Cromatina: conjunto de ADN, histonas, proteínas no histónicas y ARN que se encuentran en el núcleo interfásico de las células eucariotas y que constituye el genoma de dichas células.

Gene Ontology: Iniciativa que provee de un vocabulario controlado que describe los genes y atributos del producto génico en cualquier organismo. Se divide en dos partes; la primera es la ontología por si misma y la segunda es la anotación (caracterización de productos génicos usando términos de la ontología).

Genoma: Conjunto de genes contenidos en los cromosomas, es decir, la totalidad de la información genética que posee un organismo en particular.

Genotoxicidad: capacidad de un agente físico, químico o biológico para causar daño al material genético; este daño incluye no sólo el causado sobre el ADN si no también sobre aquellos componentes celulares que se encuentran relacionados con la funcionalidad y comportamiento de los cromosomas dentro de la célula. Este daño puede ser de tipo mutágenico o carcinógenico.

KEGG (*Kyoto Encyclopedia of Genes and Genomes*): Colección de bases de datos de genomas, rutas enzimáticas, enfermedades, drogas y sustancias químicas. Se utiliza para bioinformática, investigación y educación. Incluye el análisis de datos genómicos, meta-genómicos y metabolómicos, así como el modelado y la simulación en biología de sistemas y en la investigación transaccional para el desarrollo de drogas.

Transcriptoma: Porción del genoma que es transcripto en ARNm en ciertas circunstancias.

Tráns crito: Moléculas de ARN que darán lugar a proteínas.

Bibliografía

- [1] L. A. Bucki, *OpenProj: The OpenSource Solution for Managing Your Projects*, 1st ed. Course Technology PTR, 2008.
- [2] P. W. G. Morris, *The Management of Projects*, 1st ed. Thomas Telford, 1997.
- [3] Why progressive estimation scale is so efficient for teams. [Online]. Available: <http://www.yakyma.com/2012/05/why-progressive-estimation-scale-is-so.html>
- [4] P. Stevens and R. Pooley, *Utilización de UML en Ingeniería del Software con Objetos y Componentes*, 2nd ed. Addison Wesley, 2007.
- [5] I. Jacobson, *Object-Oriented Software Engineering: A Use Case Driven Approach*, 4th ed. Addison Wesley, 1993.
- [6] R. Barker, *El modelo entidad-relación CASE*METHOD*. Díaz de Santos, 1994.
- [7] UML 2 Sequence Diagrams: An Agile Introduction. [Online]. Available: <http://www.agilemodeling.com/artifacts/sequenceDiagram.htm>
- [8] D. M. Beazley, *Python Essential Reference*, 4th ed. Addison Wesley, 2009.
- [9] S. Bassi, *Python for Bioinformatics*, 1st ed. Chapman Hall, 2009.
- [10] M. Pilgrim. Dive into HTML5. Accessed: 2015-05-08. [Online]. Available: <http://biopython.org/DIST/docs/tutorial/Tutorial.html>
- [11] O. B. Foundation. Biopython Tutorial and Cookbook. Accessed: 2015-05-08. [Online]. Available: <http://diveintohtml5.info>
- [12] P. Gasston, *The Book of CSS3: A Developer's Guide to the Future of Web Design*, 1st ed. No Starch Press, 2011.

- [13] C. D. Professionals. Pros and cons of using frameworks. Accessed: 2015-05-08. [Online]. Available: <http://www.1stwebdesigner.com/pros-cons-frameworks>
- [14] The Django Book. [Online]. Available: <http://www.djangoproject.com/en/2.0/index.html>
- [15] F. Buschman, R. Meunier, H. Rohnert, P. Sommerlad, and M. Stal, *Pattern-Oriented Software Architecture*, 4th ed., vol. 1.
- [16] D. S. Foundation. Django at a glance. Accessed: 2015-05-08. [Online]. Available: <https://docs.djangoproject.com/en/1.8/intro/overview>
- [17] django-tables2 - an app for creating html tables. [Online]. Available: <https://django-tables2.readthedocs.org>
- [18] 10 tips for using Trello as an effective agile scrum project management tool. [Online]. Available: <http://www.tommasonervegna.com/blog/2014/1/9/10-effective-tips-for-using-trello-as-an-agile-scrum-project-management-tool>
- [19] J. Community. Jalview. Accessed: 2015-05-08. [Online]. Available: <http://www.jalview.org>
- [20] D. Higgins, F. Sievers, and D. Dineen. Clustal Omega. Accessed: 2015-05-08. [Online]. Available: <http://www.clustal.org/omega>
- [21] M. Agostino, *Practical Bioinformatics*, 1st ed. Garland Science, 2012.
- [22] yEd Graph Editor: Hight-quality diagramas made easy. [Online]. Available: <http://www.yworks.com/en/products/yfiles/yed/>
- [23] P. DuBois, *MySQL*, 5th ed. Addison Wesley, 2013.
- [24] C. Victor Jongeneel, “The need for a human gene index,” *Bioinformatics*, vol. 16, no. 12, pp. 1059–1061, 2000. [Online]. Available: <http://bioinformatics.oxfordjournals.org/content/16/12/1059.short>
- [25] django.contrib.auth. [Online]. Available: <https://docs.djangoproject.com/en/1.8/ref/contrib/auth/>
- [26] i18n vs l10n - what's the diff? [Online]. Available: <https://blog.mozilla.org/l10n/2011/12/14/i18n-vs-l10n-whats-the-diff/>
-

- [27] D. W. Huang, B. T. Sherman, and R. A. Lempicki, "Bioinformatics enrichment tools: paths toward the comprehensive functional analysis of large gene lists," *Nucleic Acids Research*, 2008.
- [28] D. W. Huang, B. Sherman, and R. A. Lempicki, "Systematic and integrative analysis of large gene lists using DAVID bioinformatics resources," *Nature*, 2008.
- [29] A. Lingling and D. R. W., "Dynamic Clustering of Gene Expression," *ISRN Bioinformatics*, vol. 2012, p. 12, 2012. [Online]. Available: <http://dx.doi.org/10.5402/2012/537217>
- [30] J. Sivriver, N. Habib, and N. Friedman, "An integrative clustering and modeling algorithm for dynamical gene expression data," *Bioinformatics*, vol. 27, no. 13, pp. i392–i400, 2011. [Online]. Available: <http://bioinformatics.oxfordjournals.org/content/27/13/i392.abstract>
- [31] V. Suárez-Ulloa, J. Fernández-Tajes, V. Aguiar-Pulido, C. Rivera-Casas, R. González-Romero, J. Ausio, J. Méndez, J. Dorado, and J. M. Eirín-López, "The CHROMEVALOA Database: A Resource for the Evaluation of Okadaic Acid Contamination in the Marine Environment Based on the Chromatin-Associated Transcriptome of the Mussel *Mytilus galloprovincialis*," *Marine Drugs*, vol. 11, no. 3, p. 830, 2013. [Online]. Available: <http://www.mdpi.com/1660-3397/11/3/830>
- [32] D. Groth, H. Lehrach, and S. Henning, "GOblet: a platform for Gene Ontology annotation of anonymous sequence data," *Nucleic Acids Research*, 2004.
- [33] Trinotate: Transcriptome Functional Annotation and Analysis. [Online]. Available: <http://trinotate.github.io/>