UNIVERSIDADE DA CORUÑA

FACULTAD DE INFORMÁTICA

DEPARTAMENTO DE

TECNOLOGÍAS DE LA INFORMACIÓN Y LAS COMUNICACIONES

TESIS DOCTORAL

MARCO METODOLÓGICO PARA EL DESARROLLO DE MODELOS DE INTEGRACIÓN DE DATOS BIOMÉDICOS. UN CASO PRÁCTICO DE INTEGRACIÓN DE DATOS BIOMÉDICOS EN LOS DISTINTOS NIVELES DE INFORMACIÓN EN SALUD

José Antonio Seoane Fernández

Directores:

Dr. Julián Dorado de la Calle

Dr. Fernando José Martín Sánchez

A Coruña, abril 2012

Agradecimientos

En primer lugar quiero darle las gracias al grupo Rnasa-IMEDIR y a sus responsables por su apoyo en la realización de esta tesis doctoral y por proporcionarme los medios necesarios para la realización de la misma.

Especialmente al profesor Julián Dorado que, además de codirector y tutor de esta tesis, ha guiado toda mi carrera investigadora, y al profesor Fernando Martín codirector de este trabajo.

A la Xunta de Galicia la concesión de la beca Isabel Barreto, que ha financiado mi etapa predoctoral, a la Universidade da Coruña que ha financiado mis estancias predoctorales en el extranjero, y al proyecto COMBIOMED del Instituto de Salud Carlos III, fruto del cual ha surgido este trabajo.

Al profesor José Luis Oliveira y a su grupo de investigación en la Universidade de Aveiro por acogerme durante mi estancia en Aveiro, y a Fernando Martín y a sus grupos de investigación, tanto en Madrid como en Melbourne, por su gran hospitalidad.

A Alba Primo, Alba Cabarcos, María García, Sonsoles Quintela, Vanessa Aguiar e Ignacio Rey por su colaboración algunos de los desarrollo de esta tesis.

Quiero agradecer especialmente a todos los miembros del grupo Rnasa-IMEDIR, pasados y presentes, por su apoyo y todas las horas que hemos compartido.

A mis amigos, por su ayuda y paciencia.

A mi familia, especialmente a mis padres que me han servido de referencia, a Merce que me animó a seguir creciendo desde el primer momento y a Antonio que ha sido mi mentor y amigo desde siempre.

A todos vosotros, muchas gracias.

A mi familia

Resumen

Los avances en las nuevas tecnologías ómicas (genómica, proteómica, transcriptómica, metabolómica, etc.) han posibilitado la obtención de enormes cantidades de datos relativos al funcionamiento de los procesos biológicos. Sin embargo, la generación de conocimiento a partir de estos datos no ha avanzado al mismo nivel debido a la dificultad para gestionar estos datos de manera que puedan ser correctamente interpretados, este hecho se debe no sólo a la ingente cantidad de datos generados, sino también a la complejidad de estos.

Además de las dificultades que representa la interacción entre los diversos datos ómicos, también existe una interacción entre éstos y los distintos niveles de información en salud (molecular, celular, tejido, órgano, individual y población), que se manifiesta como un problema de fragmentación de información. La integración de todos estos niveles de información en una entidad coherente es esencial para el avance de la medicina personalizada.

Todos estos tipos de datos biomédicos se encuentran almacenados en fuentes de datos distribuidas, heterogéneas y con diversos modelos de datos. El proceso de integración de datos biomédicos consiste en comprender la complejidad y heterogeneidad de los diversos tipos de datos, analizar las posibles soluciones tecnológicas, evaluar las características del proceso de integración, seleccionar cuidadosamente las fuentes de datos que deben ser integradas y proponer un modelo común para que estos datos puedan ser interpretados de manera conjunta.

En este trabajo se propone un marco metodológico para el desarrollo de sistemas de integración de datos biomédicos entre los distintos niveles de información en salud que facilite la tarea de desarrollo de sistemas de información aplicables a la medicina personalizada.

Para validar este marco metodológico se ha aplicado a tres escenarios de integración de información biomédica en medicina personalizada: un sistema de anotación de historia clínica electrónica, una aplicación de ayuda al diseño de estudios epidemiológicos y un sistema de integración de información de pacientes trasplantados.

Abstract

Advances in new omics technologies (genomics, proteomics, transcriptomics, metabolomics, etc.) have made obtaining huge amounts of data concerning biological processes possible. However, generating knowledge from these data has not advanced at the same level due to the difficulty of managing these data so that they could be interpreted correctly, not only because of the huge amount of data generated but also because of their complexity.

Apart from the difficulties of interaction between these omics data, there also exist an interaction between them and the different levels of health information (molecular, cellular, tissue, organ, individual and population), manifested as a problem of information fragmentation. Integrating all these levels of information into a coherent entity is essential for the advance of personalized medicine.

All these types of biomedical data are stored in distributed, heterogeneous data sources with different data models. The integration of biomedical data consists in understanding the complexity and heterogeneity of the different data types, analyzing possible technological solutions, evaluating the characteristics of the integration process, carefully selecting the data sources to be integrated and proposing a common data model for these data can be understand jointly.

In this thesis, a methodological framework for the development of biomedical data integration systems between the different levels of health information is proposed. This framework facilitates the development of information systems applicable to personalized medicine.

In order to validate this methodological framework, it has been applied to three scenarios of biomedical integration systems in personalized medicine: an electronic health record annotation system, an epidemiological association design support tool and an information integration system for transplant patients.

Resumo

Os avances das novas tecnoloxías ómicas (xenómica, proteómica, trasnciptómica, metabolómica, etc.) posibilitaron a obtención de enormes cantidades de datos relativos ó funcionamento dos procesos biolóxicos. Sen embargo, a xeración de coñecemento a partir de estes datos non avanzou ó mesmo nivel debido a dificultade para xestionar eses datos de xeito que podan ser correctamente interpretados, non só por a inxente cantidade de datos xerados senón tamén por a complexidade destes.

Amais das dificultades que representa a interacción entre os diversos datos ómicos, tamén precisa da interacción entre eles e os distintos niveles de información en saúde (molecular, celular, tecido, órgano, individual e poboación), que se manifesta como un problema de fragmentación da información. A integración de todos estes niveles de información nunha entidade coherente é esencial para o avance da medicina personalizada.

Todos estes tipos de datos biomédicos encóntranse almacenados en fontes de datos distribuídas, heteroxéneas e con diversos modelos de datos. O proceso de integración de datos biomédicos consiste en comprender a complexidade e heteroxeneidade dos diversos tipos de datos, analizar as posibles solucións tecnolóxicas, avaliar as características do proceso de integración, seleccionar coidadosamente as fontes de datos que deben ser integradas e propor un modelo común para que estes datos podan ser interpretados de xeito conxunto.

Nesta tese proponse un marco metodolóxico para o desenvolvemento de sistemas de integración de datos biomédicos entre os distintos niveles de información en saúde, que facilite a tarefa de desenvolver sistemas de información aplicables a medicina personalizada.

Para validar este marco metodolóxico aplicouse a tres escenarios de integración de información biomédica en medicina personalizada; un sistema de anotación de historia clínica electrónica, unha aplicación de axuda o deseño de estudios epidemiolóxicos e un sistema de integración de información de pacientes trasplantados.

Índice de contenidos

1 Introducción 1

1.1 Organización de la memoria 3

2 Fundamentos 5

2.1 Datos biomédicos 5

2.1.1 Datos genómicos 5

2.1.2 Investigación 'ómica' 6

2.1.3 Complejidad de los datos biológicos 7

2.1.4 Impacto de las ómicas en salud 9

2.1.5 Niveles de información en salud 10

2.2 Integración de datos 14

2.2.1 Caracterización de un sistema de integración 15

2.2.2 Clasificación de los sistemas de integración 18

2.2.3 Metadatos para Sistemas de Integración 20

2.3 Criterios de clasificación para sistema de integración 22

2.3.1 Tipos de componentes 22

2.3.2 Integración abierta / cerrada 23

2.3.3 Modelo de datos en sistemas de integración 26

2.3.4 Tipos de integración semántica 26

2.3.5 Transparencia 27

2.3.6 Paradigma de consultas 28

2.3.7 Desarrollo Botom-up vs Top-down 28

2.3.8 Integración material vs integración virtual 30

2.3.9 Acceso lectura-escritura o solo-lectura 30

2.3.10 Métodos de acceso 31

2.4 Arquitectura de los sistemas de integración 31

2.4.1 Sistemas *warehouse* 31
2.4.2 Sistemas de integración federados 38
2.4.3 Otros modelos de federación 44
2.5 Integración de datos biomédicos 47
3 Estado de la cuestión 51
3.1 Introducción 51
3.2 Sistema de navegación por links 51
3.3 Warehouse 54
3.4 Sistemas federados 60
3.5 Discusión 67
4 Hipótesis 71
4.1 Planteamiento 72
4.2 Objetivos 72
4.2.1 Objetivos específicos 72
5 Análisis conceptual 73
5.1 Análisis 73
5.1.1 Factores conceptuales 73
5.1.2 Factores técnicos 77
5.1.3 Soluciones 78
5.2 Análisis de la arquitectura 82
5.2.1 Modelos de navegación por links 82
5.2.2 *Data Warehouse* vs Modelo Federado 83
5.2.3 Otros modelos de federación 86
6 Marco metodológico propuesto 89
6.1 Análisis de requisitos 89
6.2 Análisis de casos de uso 89

6.3 Análisis de los datos a integrar 89
6.3.1 Contenido de los datos 89
6.3.2 Interfaz 90
6.3.3 Actualizaciones 90
6.3.4 Modelo de datos 90
6.3.5 Anotación semántica 90
6.3.6 Autonomía, heterogeneidad y distribución 91
6.4 Selección de la arquitectura del sistema de integración 92
6.4.1 Modelos de Navegación por links 93
6.4.2 Data warehouse 93
6.4.3 Modelo federado 93
6.5 Tipo de modelo de integración 94
6.5.1 Integración abierta vs cerrada 94
6.5.2 Estrategias top-down vs bottom-up 95
6.6 Tipo de integración semántica 96
6.7 Procesado, planificación y ejecución de consultas 97
6.8 Métrica y costes 98
6.8.1 Factores de decisión 100
6.8.2 Costes por fuente 101
6.8.3 Aplicación de la métrica 101
6.9 Aspectos tecnológicos 102
6.9.1 Seguridad y privacidad de los datos 103
6.10 Validación 104
7 Resultados 107
7.1 Sistema de visualización y anotación de historia clínica electrónica . 107
7.1.1 Presentación del caso 107

7.1.2 Análisis 109
7.1.3 Desarrollo 125
7.1.4 Pruebas 130
7.1.5 Discusión 135
7.2 Aplicación de ayuda al diseño de estudios epidemiológicos 136
7.2.1 Presentación del caso 136
7.2.2 Análisis 138
7.2.3 Desarrollo 156
7.2.4 Pruebas 159
7.2.5 Discusión 162
7.3 Sistema de integración de pacientes con trasplante de riñón 164
7.3.1 Presentación del caso 164
7.3.2 Análisis 165
7.3.3 Desarrollo 181
7.3.4 Pruebas 191
7.3.5 Discusión 192
7.4 Discusión del sistema de integración 194
8 Conclusiones 199
9 Conclusions 203
10 Futuros Desarrollos 205
Referencias 207
ANEXO I 219

Índice de figuras

Figura 1: Relación entre genómica, transcriptómica, proteómica y metabolómica 7
Figura 2: Niveles de información en salud 11
Figura 3: Niveles de información en salud + nano 12
Figura 4: Relación entre niveles de información en salud y las disciplinas de informática biomédica (Kuhn 2008) 13
Figura 5: Integración de esquemas cerrada 24
Figura 6: Integración de esquemas abierta 25
Figura 7: Esquema de data warehouse 34
Figura 8: Tres capas de un sistema federado 38
Figura 9: Arquitectura sistema de bases de datos federada 40
Figura 10: Arquitectura mediador 41
Figura 11: Arquitectura de integración con workflows de Towari et al. 2007 45
Figura 12: Ejemplo de workflow de Taverna 46
Figura 13: Portal SRS del EBI 52
Figura 14: Ejemplo del card "Leukemia" en DIseaseCard 53
Figura 15: Información KEGG sobre el card "Leukemia" 53
Figura 16: Arquitectura warehouse con mediador del DataFoundry 54
Figura 17: Análisis del pathway de la Warfarina en PharmGKB 55
Figura 18: Arquitectura del Atlas 57
Figura 19: Portal BioMart 60
Figura 20: Arquitectura usando esquemas virtuales del Ontofusion 65
Figura 21: Modelo de hubs jerárquicos del Linkhub 66
Figura 22: Arquitectura del módulo interfaz del sistema de anotación de historia clínica 126
Figura 23: Arquitectura del módulo gestor de búsquedas 128
Figura 24: Arquitectura módulo de integración 130
Figura 25: Portal de widgets 131
Figura 26: Opciones de usuario autenticado 131
Figura 27: Wiki 131

Figura 28: Foro 132
Figura 29: Búsqueda por id. de historia clínica 132
Figura 30: Visualización de historia clínica 133
Figura 31: Visualización de historia clínica con información resaltada 133
Figura 32: Historia clínica con enlaces a "trombosis" 134
Figura 33: Fuentes bibliográficas sobre "fiebre" 135
Figura 34: Arquitectura escenario 2 157
Figura 35: Ejemplo de ejecución escenario 2 160
Figura 36: Ejecución del workflow en Taverna 161
Figura 37: Resultados de la consulta 162
Figura 38: Ejemplo de aplicación del Ontology Server y del Data Source Server a un problema de neurociencia. Marenco et al. JAMIA 2009 181
Figura 39: Configuración de fuentes de datos en el DSS 182
Figura 40: Extracción automática de los metadatos en el DSS 183
Figura 41: Configuración de la ontología de OMIM 183
Figura 42: Edición de la ontología OMIM con el OntoEditor 185
Figura 43: Esquema de la ontología OMIM 185
Figura 44: Esquema de la ontología RTO que modela GO y GEO 186
Figura 45: Ontología general (RTO) 187
Figura 46: Diagrama de secuencia de funcionamiento del ontology server 189
Figura 47: Arquitectura del sistema completo 190
Figura 48: Visualización de la consulta almacenada 192
Figura 49: Consulta generada prueba 1 220
Figura 50: Resultados prueba 1 221
Figura 51: Resultados prueba 2 222
Figura 52: Resultados prueba 3 223
Figura 53: Resultado prueba 4 225
Figura 54: Resultados prueba 5 226
Figura 55: Resultados prueba 6 228
Figura 56: Resultados prueba 7 229
Figura 57: Resultados prueba 8 231
Figura 58: Resultados prueba 9 232

Figura 59: Resultados prueba 10 234

Índice de tablas

Tabla 1: Valores de importancia de los factores de decisión en arquitectura escenario 1 110
Tabla 2: Valores de importancia de los factores de decisión en modelo de integración escenario 1..... 111
Tabla 3: Valores de importancia de los factores de decisión en estrategia de integración escenario 1..... 111
Tabla 4: Valores de importancia de los factores de decisión en tipo de integración semántica escenario 1 111
Tabla 5: Valores de importancia de los factores de decisión en gestión de consultas escenario 1..... 112
Tabla 6: Valoración arquitectura escenario 1 118
Tabla 7: Valoración modelo integración escenario 1 119
Tabla 8 Valoración estrategia escenario 1..... 119
Tabla 9 Valoración técnica de integración semántica escenario 1 120
Tabla 10: Valoración gestión de consultas escenario 1..... 121
Tabla 11: Impacto arquitectura escenario 1 121
Tabla 12: Impacto modelo de integración escenario 1 122
Tabla 13: Impacto estrategia de integración escenario 1 122
Tabla 14: Impacto integración semántica escenario 1..... 123
Tabla 15: Impacto gestión de consultas escenario 1..... 124
Tabla 16: Importancia de los factores de decisión en arquitectura escenario 2 140
Tabla 17: Importancia de los factores de decisión en tipo de integración escenario 2140
Tabla 18: Importancia de los factores de decisión en estrategia de integración escenario 2..... 140
Tabla 19: Importancia de los factores de decisión en integración semántica escenario 2 141
Tabla 20: Importancia de los factores de decisión en gestión de consultas escenario 2 141
Tabla 21: Valoración arquitectura escenario 2 146
Tabla 22: Valoración modelo de integración escenario 2 148

Tabla 23: Valoración estrategia de integración escenario 2 149
Tabla 24: Valoración tipo de integración semántica escenario 2 150
Tabla 25: Valoración gestión de consultas escenario 2..... 151
Tabla 26: Impacto arquitectura escenario 2 151
Tabla 27: Impacto modelo de integración escenario 2..... 152
Tabla 28: Impacto estrategia de integración escenario 2 153
Tabla 29: Impacto tipo de integración semántica escenario 2 153
Tabla 30: Impacto gestión de consultas escenario 2..... 154
Tabla 31: Importancia de los factores de decisión en arquitectura escenario 3 166
Tabla 32: Importancia de los factores de decisión en modelo de integración en escenario 3..... 167
Tabla 33: Importancia de los factores de decisión en estrategia de integración en escenario 3..... 167
Tabla 34: Importancia de los factores de decisión en tipo de integración semántica en escenario 3..... 167
Tabla 35: Importancia de los factores de decisión en gestión de consultas en escenario 3 168
Tabla 36: Valoración arquitectura escenario 3 172
Tabla 37: Valoración del modelo de integración en escenario 3 173
Tabla 38: Valoración de estrategia de integración en el escenario 3 174
Tabla 39: Valoración de tipos de integración semántica en el escenario 3 175
Tabla 40: Valoración de la gestión de consultas en el escenario 3 175
Tabla 41: Impacto arquitectura escenario 3 176
Tabla 42: Impacto modelo de integración escenario 3 177
Tabla 43 Impacto estrategia de integración escenario 3 177
Tabla 44: Impacto integración semántica escenario 3..... 178
Tabla 45: Impacto gestión de consultas escenario 3..... 179

1 Introducción

En los últimos años la investigación biomédica se ha transformado debido al crecimiento, tanto en escala como en complejidad, de los nuevos datos, producido por el desarrollo tecnológico en las técnicas de alto rendimiento (*high-throughput*) y el abaratamiento de los experimentos utilizando estas técnicas. Esta nueva aproximación de investigación "-ómica" representa una gran oportunidad para el desarrollo de nuevas aproximaciones de investigación y obtención de nuevo conocimiento, sin embargo, tiene también una lectura negativa.

Los datos obtenidos a partir de estas técnicas de alto rendimiento "-ómicas", entre las que se incluye la genómica, proteómica, metabolómica, etc. permitirán ampliar como nunca antes el conocimiento de los procesos biológicos y su aplicación translacional en salud. Sin embargo, toda esta gran cantidad de datos generados deben ser correctamente almacenados, organizados e integrados para poder interpretarlos de manera correcta.

Estos desafíos tecnológicos no son fáciles de solucionar, y de ellos depende la correcta interpretación de toda esta información. El hecho de no solucionarlos puede conllevar el riesgo de perder importantes descubrimientos potenciales o lo que es peor, llegar a conclusiones erróneas.

La gestión y el almacenamiento de los datos biomédicos se ha llevado a cabo hasta ahora de una manera relativamente desorganizada, distribuyéndolos en una gran cantidad de bases de datos (Galperin and Cochrane, 2011) con diferentes formatos (Wang et al., 2005) (Brazma et al., 2006) y diferentes métodos de acceso. Esta distribución y heterogeneidad conlleva que el proceso de integración sea aún más complicado.

El proceso de integración de datos biomédicos tiene, además de la dificultad dependiente de esta distribución y heterogeneidad, otro problema inherente a los propios datos biomédicos que es su complejidad biológica. Los sistemas de integración normalmente han sido desarrollados para integrar datos que han sido modelados por humanos, mientras que los sistemas de integración de datos biológicos deben modelar

sistemas que han surgido del proceso de la evolución, lo que provoca un gran problema a la hora de definir los conceptos que se intentan modelar, ya que los sistemas biológicos tienden a expresar propiedades que no son definibles formalmente y que no son naturalmente descomponibles (Bernstam et al., 2010).

Para solucionar el problema de la integración de datos biomédicos se han desarrollado diferentes soluciones durante los últimos 15 años, que pueden resolver adecuadamente alguno de los pasos del proceso de integración, sin embargo no se ha abordado el problema de desarrollar un modelo genérico de integración de datos biomédicos.

Esto es debido principalmente a la gran cantidad de distintos tipos de problemas de integración que pueden encontrarse, tanto en la integración de datos ómicos, como en su aplicación en la medicina translacional. No existe un modelo genérico de integración que permita solucionar cualquier tipo de problema de integración de datos biomédicos.

El desarrollo de un modelo de integración de datos biomédicos para un problema en concreto requiere entender la complejidad de los tipos de datos que se van a tratar, analizar todas las posibles soluciones, evaluar cada una de las diferentes características del proceso de integración y seleccionar cuidadosamente todas las fuentes de datos que deben ser integradas.

Sería por tanto de gran utilidad contar con algún tipo de metodología que permita de una manera formal analizar los tipos de datos a integrar, así como seleccionar correctamente las distintas características del proceso de integración por medio de algún tipo de métrica. De esta manera, el ingeniero podría desarrollar un sistema de integración siguiendo una serie de pasos y teniendo en cuenta una serie de factores necesarios o aconsejables tanto para el sistema de integración como para la selección de las fuentes.

La utilización de este tipo de metodología permitiría agilizar el proceso de desarrollo, minimizar los errores de diseño y los tiempos de desarrollo o evolución del sistema de integración de datos biomédicos.

1.1 Organización de la memoria

A continuación se describe la organización de la memoria a través de los distintos capítulos:

Este primer capítulo sirve para introducir el ámbito del problema de integración de datos en el que se desarrollará esta tesis.

En el segundo capítulo se muestran los fundamentos de la integración de datos, haciendo especial mención a las especiales características de los datos biomédicos, que incluyen los datos de los distintos niveles de información en salud, desde moleculares a poblacionales.

En el tercer capítulo se hace un recorrido por las distintas aproximaciones de integración de datos biomédicos encontradas en la bibliografía. Se dividen los distintos desarrollos en función del tipo de arquitectura que se utilizó.

En el cuarto capítulo se plantea la problemática introducida en el primer capítulo y a continuación se formula la hipótesis y los objetivos del siguiente trabajo.

En el capítulo quinto se realiza un análisis de las particularidades de la integración de datos biomédicos, identificando los principales factores técnicos y conceptuales y proponiendo una serie de soluciones.

En el capítulo sexto se plantea el marco metodológico en el que se fundamenta la presente tesis, describiendo cada uno de los pasos del marco metodológico.

En el séptimo capítulo se muestran los resultados de aplicación de la metodología en tres distintos escenarios relacionados con la integración de datos en medicina personalizada.

En el capítulo octavo se ofrece la discusión de los resultados obtenidos en el capítulo anterior y el análisis del funcionamiento del marco metodológico.

En el capítulo noveno se plantean las conclusiones derivadas del presente trabajo, tanto en castellano como en inglés, siguiendo la normativa para optar a la mención de Doctor Internacional.

Finalmente en el capítulo décimo se plantean las futuras líneas de investigación sobre los resultados alcanzados en este trabajo.

2 Fundamentos

En 1985, en el informe de la National Academy of Science "Models for Biomedical Research: A New Perspective" (NRC and Research, 1985), los autores argumentaban que la investigación biomédica había llegado a un punto en el que "nuevas generalizaciones y leyes de alto orden biológico se están acercando, pero pueden ser ocultadas por la simple masificación de los datos". Los autores proponían la creación de una "Matriz de Conocimiento Biológico", en la que los datos, la información y el conocimiento se estructurasen y almacenasen para proporcionar una visión integrada de la biología. Durante las siguientes décadas, los datos que se generaron excedieron las expectativas del informe, comenzando la generación de grandes cantidades de datos biológicos.

2.1 Datos biomédicos

Con la elucidación del Proyecto Genoma Humano (Lander et al., 2001) (Venter et al., 2001) y sus implicaciones funcionales, la comunidad biomédica se está encontrando con un creciente número de datos y de información que están teniendo una relevancia y repercusión cada vez mayor en la práctica clínica. La situación tradicional que se ha dado hasta ahora ha sido la realización de estudios en los distintos niveles de información sobre la salud (población, paciente, órgano o tejido, celular y molecular o genético), en los que no se ha integrado la información y datos procedentes de los estudios que se realizaban en paralelo, pero en otro nivel. Esta situación se está revirtiendo a medida que cada vez están disponibles una mayor cantidad de datos y se están comenzando a demostrar las interacciones entre todos los niveles, desarrollándose iniciativas de integración de la información procedente de estos estudios, como único modo de realizar aproximaciones integradas para la investigación de enfermedades.

2.1.1 Datos genómicos

Hace más de dos décadas que se ha reconocido la importancia de determinar la secuencia del genoma humano, pero no ha sido más que el primer paso en la era de la genómica. Durante ese tiempo también se creía que el objetivo de descifrar el genoma estaba al alcance de la mano tanto con las técnicas de secuenciación existentes como con las técnicas de *mapping* y clonado a gran escala que se desarrollaron. La

secuenciación del genoma humano comenzó caracterizando una serie de marcadores para ensamblar el mapa de ligamiento y desarrollando un mapa físico del genoma. El ensamblado del genoma se realizó gracias a la disponibilidad de marcadores uniformemente distribuidos a lo largo del genoma. El uso de enzimas de restricción y el análisis de largos fragmentos de ADN mediante electroforesis facilitaron la construcción de este mapa físico. El desarrollo de vectores capaces de dar cabida a largos fragmentos genómicos permitió el ensamblaje del genoma humano como un array de grandes clones de ADN. La secuenciación comenzó con un consorcio internacional financiado por el Departament of Energy, que continuó como el Human Genome Project en el National Institute of Health (NIH) de los Estados Unidos. Simultáneamente, una iniciativa privada, a manos de Craig Venter del Institute of Genomic Research y Celera Genomics perseguía el mismo objetivo. Mientras que el consorcio liderado por el NIH usaba el array de clones ordenados para secuenciar el genoma, el grupo Venter empleó una técnica de secuenciar pequeñas partes aleatoriamente y luego utilizar análisis computacional para ensamblar las secuencias solapadas. Utilizando estas dos aproximaciones, el borrador del genoma humano se presentó en 2001.

El siguiente paso en esta dirección fue estudiar la diversidad genética en los humanos, identificando varios haplotipos presentes en la población, secuenciando genomas representativos. Durante el transcurso de esta empresa, se desarrollaron nuevas técnicas de secuenciación, como la pirosecuenciación y la secuenciación por síntesis de una sola molécula. Estos métodos han hecho el análisis de secuencias más rápido y barato y, actualmente, es posible secuenciar un genoma completo por menos de 1000€.

2.1.2 Investigación 'ómica'

En esta era post-genómica, el modelo de la vida está siendo descrito no sólo con la secuenciación del genoma, sino también con la evaluación simultanea de grandes cantidades de ARN mensajero (transcriptómica), ARN interferente (interferómica), proteínas (proteómica), interacciones de proteínas (interactómica), modificaciones del ADN (variómica) y la cromatina (epigenómica) y metabolitos (metabolómica), entre otras.

Los desarrollos de estas disciplinas han producido un cambio en la manera en la que se hace la biología. La biología molecular estaba basada en el estudio laborioso de un gen, de una proteína o un proceso a lo largo de toda la carrera del investigador. Actualmente la biología se ha transformado en una gran parte, en una operación de generación de datos de alto rendimiento (*high-throughput*). Actualmente, las operaciones de análisis se llevan a cabo por medio de robots programables que son microminiaturizados (en un futuro *nanominiaturizados*), que permiten la obtención altamente paralela de grandes cantidades de datos en un pequeño espacio de tiempo.

Las diferentes tecnologías ómicas se relacionan unas con otras siguiendo el dogma de la biología como se muestra en la Figura 1.

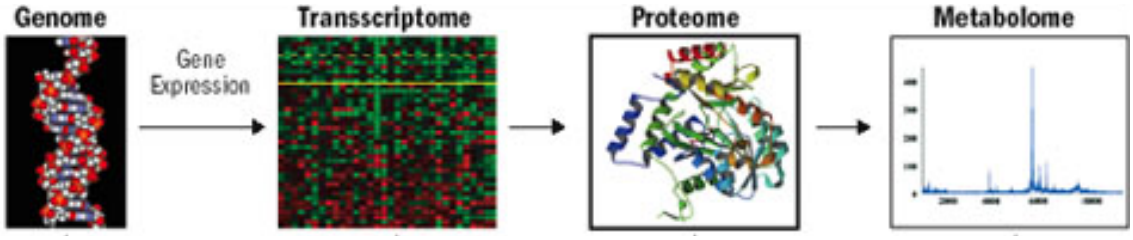


Figura 1: Relación entre genómica, transcriptómica, proteómica y metabolómica

Todos estos desarrollos reflejan lo que puede ser denominada la "investigación ómica". Además de las mencionadas anteriormente, existen otras como las kinómica (de las kinasas), CHOmica (de los carbohidratos), farmacogenómica, genómica estructural, farmacometilómica, bibliómica, etc. Etimológicamente, el diccionario Webster define '-oma' como una entidad, grupo o masa abstracta. La investigación ómica en biología es el estudio de entidades en un agregado (ADN, ARN, proteína y otros componentes moleculares como una célula, un tejido o un organismo). Lo esencial de la investigación ómica radica en que requiere un cambio de mentalidad con respecto a la investigación biológica clásica de un gen o proceso. Esto generalmente lleva a saber poco de mucho, en lugar de mucho de poco (Weinstein, 2002).

2.1.3 Complejidad de los datos biológicos

Es bien conocido que la biología es compleja. Hace varios años se creía que había muchos más genes de los que realmente existen. Sin embargo, el problema de la complejidad en la biología no se refiere sólo a los números, sino a los niveles. La mayoría de las proteínas funcionan en forma de complejos macromoleculares o

máquinas moleculares, y deben transportase entre los compartimentos celulares o activarse o bloquearse en múltiples lugares de fosforilación. Pero quizá el aspecto más importante (y menos conocido) que define la complejidad de la biología, que la hace tan complicada de entender, es el hecho de que los modelos biológicos no fueron desarrollados por un ingeniero. Los sistemas biológicos han evolucionado orgánicamente, el proceso evolutivo construye sistemas de una manera orgánica por prueba y error que eventualmente puede volverse irreversible si es suficientemente exitoso.

Esta complejidad en los sistemas biológicos trabaja como un todo, dado el gran número de componentes individuales y las interacciones entre ellos, a veces se habla del proceso evolutivo como milagro. Una base incorrecta puede provocar homeostasis y sabotear el organismo. Ningún humano podría diseñar jamás una máquina con tantas partes móviles independientes, siendo el transbordador espacial algo trivial al lado de cualquier organismo. Pero la noción de la vida como un milagro aleja el punto de vista del proceso evolutivo. Una vez que surge el proceso evolutivo, existen nichos biológicos esperando ser ocupados, y estos serán ocupados por algún tipo de forma de vida. Lo que vemos alrededor de nosotros son los linajes supervivientes de un torneo de 4.000 millones de años de duración. Como en los torneos deportivos, algunos jugadores genómicos están obligados a sobrevivir en cada ronda del juego. En el caso de la evolución biológica, los nuevos linajes genéticos se van separando continuamente para competir en la siguiente ronda, pero la conclusión permanece. Unos sobreviven y lo hacen porque han evolucionado para trabajar suficientemente bien para vencer a sus competidores a través del tiempo evolutivo. Es porque son un producto de la selección evolutiva, por lo que los sistemas biológicos no han sido optimizados de acuerdo a un conjunto definido de criterios o, al menos, a un conjunto de criterios que se puedan definir. La regla de la parsimonia funciona perfectamente en la biología, si se puede imaginar una manera más simple de que una función pueda realizarse o que una estructura pueda diseñarse, se puede estar seguro de que un sistema biológico no lo hará de esa manera.

Este argumento surge en varios autores cuando se enfrentan, desde distintos campos con las ómicas. Desde el campo de la biología, Weinstein (Weinstein, 2002)

intenta buscar un modelo de hipótesis que permita integrar las ómicas para comprender la complejidad de la biología, mientras que otros autores, desde la perspectiva de tratamiento de la información, como Bernstam (Bernstam et al., 2010), Blois (Blois, 1984) o Simon (Simon, 1996) intentan justificar la complejidad de la modelización de la biología en contraste con entes modelables artificiales.

2.1.4 Impacto de las ómicas en salud

Indudablemente, los distintos datos obtenidos a partir de la investigación ómica pueden ser analizados y comparados con el objetivo de producir conocimiento sobre las causas moleculares de las enfermedades y, finalmente, proporcionar nuevas soluciones diagnósticas y terapéuticas.

La medicina molecular representa un intento de definir tanto el estado normal como patológico, en términos de la presencia y regulación de entidades moleculares a seres vivos. Existen muchas aplicaciones potenciales de la medicina genómica a la salud, incluyendo el tratamiento de salud individualizado basado en genética, métodos predictivos para evaluar la predisposición a una enfermedad, nuevas dianas de fármacos, terapia génica o epidemiología genética para estudiar enfermedades en diferentes poblaciones. El campo de la medicina genómica se puede ver como un gran mosaico de disciplinas relacionadas, entre las cuales se pueden mencionar la genética moderna, que trata de identificar genes responsables de una enfermedad moderna, la farmacogenética y la farmacogenómica (Schadt et al., 2003), que buscan cómo los genes pueden afectar a la respuesta de un determinado fármaco, diseño de fármacos personalizados (Claus and Underwood, 2002), cuyo objetivo es usar toda la información química, biológica y clínica para un mejor diseño, selección y administración de fármacos a cierto paciente o el desarrollo del punto de atención al paciente, que trata de que los clínicos tengan toda la información genética y clínica para usar “a pie de cama” (Fuller et al., 1999, Bush et al., 2004, Overby et al., 2010, Cabarcos et al., 2010).

A pesar de las distintas disciplinas de la medicina genómica, y de todos los tipos de datos ómicos relacionados con ella, hay un hilo común: todas las disciplinas tratan de descubrir la relación entre un genotipo y un fenotipo. Un genotipo está definido por

el perfil genético, mientras que el fenotipo puede ser definido como las propiedades visibles de un organismo que se producen debido a la interacción del genotipo y el entorno. En el contexto de la medicina genómica la conexión entre el genotipo y el fenotipo puede ser definida como los polimorfismos o haplotipos relacionados con una enfermedad o con las respuestas del organismo al tratamiento a una enfermedad.

La medicina genómica definirá sin duda el futuro de la medicina, pero también el futuro de la atención sanitaria. Un ejemplo de esto es que en 2006, el por entonces senador por Illinois Barack Obama presentó en el congreso un proyecto de ley "Genomics and Personalized Medicine Act 2006" (Obama, 2007) en la que afirmaba *"…estamos en una nueva era en las ciencias de la vida, y la verdad de esta afirmación se puede ver en campos como la imagen médica, los nuevos fármacos, o incluso en el uso de tecnología de ADN para mejorar el medio ambiente y reducir los gases que provocan el efecto invernadero. Pero ningún área de investigación es una promesa mayor que el campo de la medicina personalizada*". A pesar de que esta propuesta, hasta la fecha, no ha salido adelante ninguna de las cuatro veces que se presentó, define las líneas de actuación de la administración americana con respecto a la importancia de la medicina genética en el sistema de salud a largo plazo.

2.1.5 Niveles de información en salud

En "Information and Medicine: the nature of medical descriptions" (Blois, 1984), Blois establece una serie de niveles de información en salud, que van desde el nivel más bajo, el nivel molecular, al nivel más alto, el nivel poblacional (Figura 2). Estos niveles obviamente no están físicamente delimitados y existe un intercambio de información entre ellos, como defienden Kulikowski (Kulikowski, 2002) y Kuhn et al. (Kuhn et al., 2008). El desarrollo de nuevas tecnologías de alto rendimiento permite incrementar las posibilidades de cada uno de estos niveles (Martin-Sanchez et al., 2002).

Estos niveles incluyen el nivel molecular, que se refiere a los datos bioquímicos que se generan por medio de las tecnologías ómicas de alto rendimiento, pero también a los datos genéticos singulares. La interpretación de todos estos datos en el contexto correcto y la transformación de estos en resultados con significado para

aplicarlos en la práctica clínica es el objetivo más importante de la investigación biomédica.

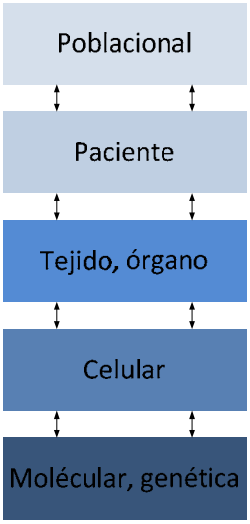


Figura 2: Niveles de información en salud

En el siguiente nivel se situarían los datos relacionados con la célula, que tienen que ver con el transporte, rutas metabólicas y que se pueden obtener con técnicas de imagen microscópica, microbiología o a partir de los datos del nivel inferior, a través del análisis de expresión o de análisis proteómico en una determinada célula.

El siguiente nivel es el de tejido/órgano en el que se analizan los datos de órganos o tejidos obtenidos a partir del estudio, por medio de técnicas de imagen o de biopsias de muestras patológicas, pero también a partir de datos de fisiología o de redes genéticas que afectan a un órgano concreto. El nivel de información de paciente suele ser el más estudiado e incluye todos los datos clínicos relacionados con el paciente que figuran en la historia clínica, técnicas de imagen convencionales o complejas, análisis de señales, análisis bioquímicos, etc. Con la llegada de las ómicas se pueden incluir también los perfiles genéticos de un paciente concreto, SNPs, etc. que lleven a la medicina personalizada.

El siguiente nivel es el que se refiere a información sobre el paciente. Tradicionalmente las enfermedades se han venido agrupando de acuerdo a las manifestaciones clínicas de los pacientes, más tarde, con los avances del conocimiento en medicina, los resultados de diversos tipos de análisis se añadieron a las manifestaciones clínicas. Ahora se puede añadir el conocimiento proporcionado por la

medicina molecular, redefiniendo las enfermedades usando los mecanismos causales y el genotipo del paciente, en lugar de únicamente el fenotipo.

Finalmente en el nivel poblacional, se encuentran los datos epidemiológicos relacionados con datos ambientales, indicadores de salud, etc. A estos hay que añadir los datos de epidemiología genética, disciplina que se encarga de gestionar y analizar los datos relevantes de interacción gen-ambiente que contribuyen al desarrollo de enfermedades o a la respuesta al tratamiento a estas.

Además de los niveles anteriores, en los últimos años se ha considerado añadir un nivel por debajo del nivel de información molecular, que puede ser considerado el nivel de información atómico (Figura 3) (Martin-Sanchez and Maojo, 2009). Este nivel está formado por información a nivel atómico obtenido a partir de los avances en la nanotecnología.

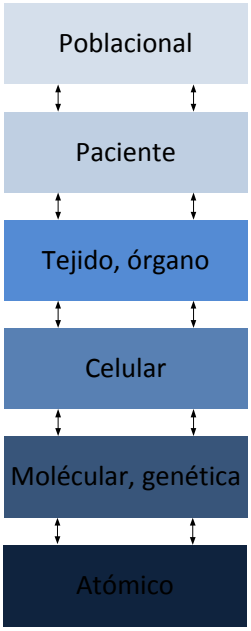


Figura 3: Niveles de información en salud + nano

Como se ha visto, tanto a nivel de las distintas ómicas como a nivel de la interacción entre estas ómicas y los distintos niveles de información en salud, existe un problema de fragmentación del conocimiento proporcionado por cada uno de estos niveles de información. La integración de estos niveles de información en una única

entidad coherente es esencial para que la comunidad de investigadores biomédicos puedan comprender mejor las ciencias de la vida.

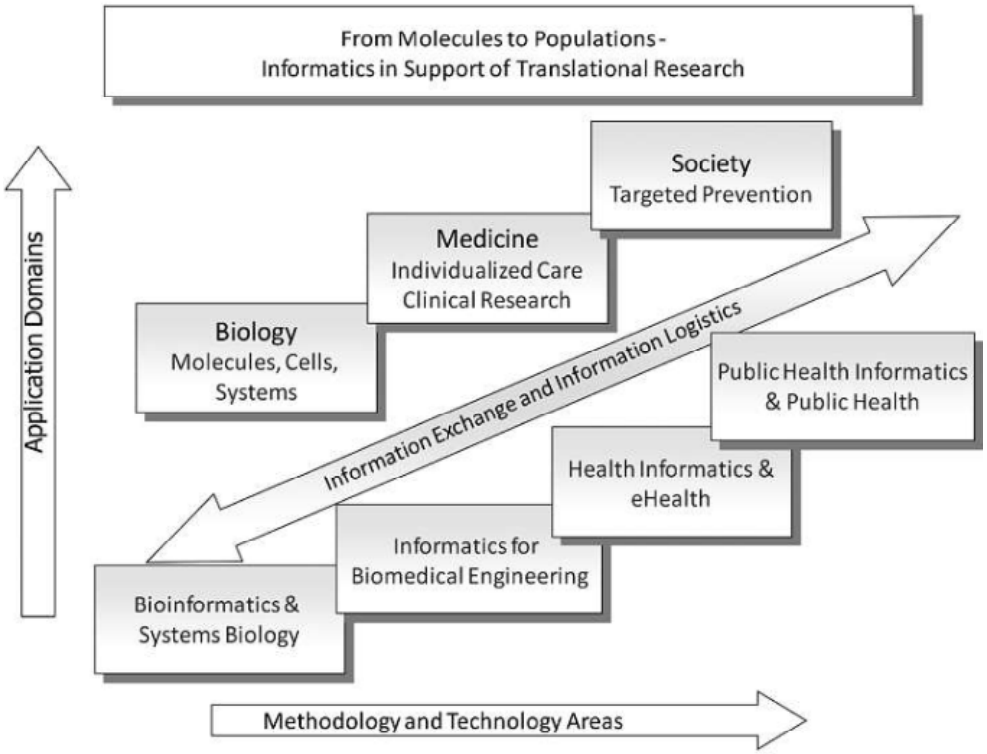


Figura 4: Relación entre niveles de información en salud y las disciplinas de informática biomédica (Kuhn 2008)

La informática biomédica ha de desempeñar un papel preponderante en la integración de estos nuevos datos y estos nuevos enfoques a la hora de realizar sus aplicaciones sanitarias. Desde un punto de vista clásico la informática en salud pública ha sido capaz de enfrentarse y resolver problemas a niveles poblacionales, las aplicaciones de la informática médica ha manejado eficazmente los niveles tanto de la enfermedad como del paciente y las disciplinas de procesamiento de imágenes médicas han desarrollado herramientas de gestión y análisis de imágenes en la realización de técnicas no invasivas para el estudio de órganos y tejidos. Por otro lado, la bioinformática ha proporcionado herramientas para analizar la información a nivel molecular y genético. En la Figura 4 extraída de (Kuhn et al., 2008), se muestran las interacciones entre las diferentes disciplinas de informática biomédica y estos niveles de información en salud.

2.2 Integración de datos

La integración de datos que están almacenados en diferentes sistemas es un tema de investigación activo desde hace muchos años. Mientras que en los 80 la investigación se centró principalmente en obtener los datos distribuidos sobre diversas aplicaciones propietarias incompatibles sobre una sola base de datos centralizada, en los noventa ha habido una necesidad de combinar los datos almacenados en diferentes bases de datos. Algunos de los primeros trabajos en sistemas de datos multi-bases de datos se publicaron en el 85 (Heimbigner and McLeod, 1985). Posteriormente, el término base de datos federada emergió para caracterizar técnicas que proveen de acceso integrado a un conjunto de bases de datos distribuidas, heterogéneas y autónomas (Litwin et al., 1990) (Sheth and Larson, 1990). En (Sheth and Larson, 1990) se define la arquitectura clásica de bases de datos federadas. En 1992 Inmon (Inmon, 1992) define los almacenes de datos o *data warehouse* como "*una colección de datos clasificada por temas, integrada, variable en el tiempo y no volátil que se utiliza como ayuda al proceso de toma de decisiones por parte de quienes dirigen una organización*".

A lo largo de los años los requisitos de la integración de datos han cambiado. En paralelo, las técnicas y métodos de intercambio de información se han mejorado. Sin embargo, las diferentes fuentes, unidas al amplio acceso a los datos, han creado nuevos problemas. Por ejemplo, la integración de esquemas no es una buena aproximación si los datos no están descritos en los esquemas, en lugar de a través de formatos, y la solución clásica de consultas falla si todos los datos no están accesibles a través de lenguajes de consulta. Por otro lado, la integración de datos normalmente se intenta sin informar a las fuentes de datos que están sido utilizadas por un sistema de integración de datos. Esto lleva el problema de la autonomía a una nueva dimensión. Cualquier cambio en la fuente o en el esquema, no se notifica a los sistemas de integración. Esto es especialmente sensible en entornos científicos (Mattos et al., 1999), donde los datos no se almacenan únicamente en bases de datos, sino en diferentes tipos de ficheros. Es en estos casos donde las nuevas técnicas de integración son más necesarias.

Al mismo tiempo, muchos paradigmas de desarrollo de software han cambiado. Las aplicaciones centralizadas y monolíticas son consideradas del pasado, mientras que los sistemas modernos son construidos incrementalmente a base de componentes distribuidos. Las arquitecturas cliente-servidor son sustituidas por arquitecturas de n capas que intercambian datos en diferentes protocolos. Por otro lado, los sistemas software deben ser, desde un primer momento, construidos pensando en su evolución futura, debido a muchas causas, como puede ser que sus desarrollos requieren de una gran inversión y son críticos en las empresas. La experiencia ha mostrado que la evolución del software es muy compleja, por lo tanto, los sistemas software deben estar preparados para reaccionar al cambio. Esto es particularmente crítico en los sistemas de integración de datos que tienen muchas potenciales fuentes de cambio: las fuentes de datos que integran.

Estas fuentes de datos han modificado los requisitos de la integración de bases de datos. El artículo de Wiederhold (Wiederhold, 1992) definía un escenario de un conjunto de componentes de integración ligeros denominados mediadores, que accedían a fuentes de datos u otros mediadores a través de protocolos interactivos. Las fuentes de datos son encapsuladas por adaptadores software, denominados *wrappers*, que tienen la habilidad de presentar los datos de manera que el cliente (el mediador), los necesita.

2.2.1 Caracterización de un sistema de integración

Los sistemas de integración pueden ser caracterizados de acuerdo a las dimensiones de autonomía, heterogeneidad y distribución (Busse et al., 2000), además de su capacidad y flexibilidad para la evolución:

- Autonomía: Cuando los sistemas de integración están situados en múltiples módulos, también denominados componentes, la autonomía de los componentes integrados debe ser un tema crítico. Se pueden distinguir según (Ozsu, 1991) varios tipos de autonomía:
    - Autonomía de diseño: Un componente es independiente de otros en su diseño, relativo a temas como el modelo de datos, nombre de conceptos, etc. La autonomía de diseño persigue la autonomía de

cambiar el diseño en cualquier punto, en cualquier momento, con la particularidad de que es complicado manejarla en infraestructuras de componentes.

- Autonomía de comunicaciones: Se da cuando un componente puede decidir independientemente con qué otro componente se comunica. En sistemas de integración la autonomía de comunicación implica que cada componente puede dejar o entrar en el sistema en cualquier momento.
- Autonomía de ejecución: denota la autonomía de cada componente en la ejecución o en la planificación de peticiones entrantes.

- Heterogeneidad: La heterogeneidad de un sistema ocurre debido al desarrollo autónomo de unos sistemas con respecto a otros debidos a diferentes modelos del mismo concepto, entornos tecnológicos distintos, requisitos particulares de aplicaciones, etc. Superar el problema de la heterogeneidad es uno de las principales tareas de la integración. En general se distinguen los siguientes tipos de heterogeneidad (Wiederhold, 1996):
  - Heterogeneidad sintáctica:
    - Heterogeneidad tecnológica referida a aspectos técnicos como hardware, sistemas operativos, métodos de acceso, protocolos, seguridad, etc.
    - Heterogeneidad de interfaz: ocurre cuando diferentes componentes son accesibles a través de diferentes métodos de acceso:
      - Heterogeneidad de lenguaje: diferentes lenguajes de consulta o restricciones del lenguaje.
      - Restricciones de consulta: Únicamente pueden realizarse algunos tipos de consulta, expresiones limitadas, etc.
      - Restricciones de enlace: Ciertos atributos deben ser especificados de manera que formen una consulta válida
  - Heterogeneidad del modelo: Diferentes modelos de datos tienen distinta semántica para sus conceptos. Por ejemplo, el modelo relacional no tiene herencia, en contraste con el modelo orientado a

objetos. Aunque la heterogeneidad en el modelo de datos es un conflicto semántico, se trata de manera separada porque muchos sistemas manejan aparte este problema y otros conflictos semánticos, ya que los modelos de datos contienen únicamente ciertos conceptos básicos. Por ejemplo, muchos sistemas de integración utilizan *wrappers* para acceder a las fuentes de datos originales, que ocultan los diversos modelos de datos, pero no la heterogeneidad semántica.

- Heterogeneidad lógica:
  - Heterogeneidad semántica: Se refiere a la semántica de los datos y del esquema. En ocasiones varios esquemas de un mismo modelo pueden tener diferentes semánticas. Se partirá de que un esquema es un conjunto de relaciones y atributos. Estas relaciones y atributos tienen nombres que llevan una semántica implícita, que es el concepto que representan. Sin embargo, el esquema contiene únicamente el nombre, pero no el concepto. La interpretación de los nombres por personas distintas no tiene porqué coincidir, además de los conflictos semánticos, como que el mismo nombre denote diferentes conceptos (homónimos), diferentes nombres para el mismo concepto (sinónimos), divergencia en la interpretación del mismo concepto, etc. Los atributos también pueden tener el mismo significado, pero distintas unidades de medida. La semántica de los datos es definida a través del esquema bajo el cual se representan. Los valores no llevan ningún tipo de significado, pero se describen a través de la parte del esquema a la que pertenece. Esto no contradice el hecho de que existen sistemas en los que es necesario analizar los valores para extraer la semántica del esquema.
  - Heterogeneidad esquemática: Es la codificación de los conceptos en diferentes partes del modelo de datos. En un modelo relacional, existen 3 tipos de conflictos (Krishnamurthy et al., 1991): relación <-> nombre atributo, nombre atributo <->

valor atributo y relación <-> valor atributo. Tratar con la heterogeneidad esquemática requiere normalmente lenguajes de consulta que sean “sintácticamente de segundo orden (de lógica de predicados), pero semánticamente de primer orden”, debido a la necesidad de variables que cubran el rango de un atributo o nombre de tablas. Son semánticamente de primer orden debido a que siempre se refieren a un esquema concreto.

  - Heterogeneidad estructural: Existen elementos que tienen el mismo significado, están modelados con el mismo modelo de datos y son esquemáticamente homogéneos, pero se estructuran de distinta manera, como por ejemplo, atributos distribuidos en distintas tablas.

- Distribución: El tercer problema, ortogonal a la autonomía y a la heterogeneidad en las fuentes de datos es la distribución física de los datos. Desde que los ordenadores están conectados a algún tipo de red, es natural pensar en la combinación de aplicaciones y fuentes de datos que están físicamente localizadas en diferentes lugares pero conectadas a través de una red. La distribución se ha realizado utilizando diferentes métodos, desde sockets, RPC, Servicios Web, CORBA, .Net Remoting, etc. Usando tecnologías como por ejemplo, CORBA, Servicios Web o .Net Remoting, las aplicaciones se pueden desarrollar para que ignoren hasta cierto punto la localización física de los componentes. Por este motivo no se tratará la distribución como un problema separado, asumiendo que los datos se distribuyen por componentes que no comparten ni memoria ni disco.

2.2.2 Clasificación de los sistemas de integración

Basándose en dos de las dimensiones vistas anteriormente (distribución y heterogeneidad), se pueden separar tres grandes clases de sistemas de integración:

- Sistemas de integración Monolíticos, que funcionan como aplicaciones monolíticas en una sola máquina. Pudiendo ser, por un lado, sistemas que usan una base de datos para almacenar y gestionar los datos, los cuales están basados en un modelo de datos. Los datos se estructuran de acuerdo a un

esquema y se accede a ellos a través de un lenguaje de consultas. Por otro lado, pueden no usar sistemas de bases de datos, usando sistemas de ficheros, colecciones de documentos, etc. denominados semi-estructurados. Estos sistemas no suelen estar basados en un modelo de datos estándar y no suelen ofrecer un lenguaje de consultas.

- Sistemas de integración Distribuidos: Los datos están distribuidos físicamente sobre múltiples lugares conectados con algún tipo de red de comunicación
- Sistemas de integración Heterogéneos: Son una colección de sistemas de integración que difieren en aspectos sintácticos y lógicos, como pueden ser plataformas hardware, modelo de datos o semántica.

Si se añade una nueva dimensión a esta clasificación, la autonomía, a la hora de clasificar los sistemas de integración heterogéneos, se obtienen dos claras categorías:

- Sistemas de integración heterogéneos autónomos: El sistema de integración mantiene una autonomía de diseño, autonomía de comunicaciones y autonomía de ejecución. Dentro de esta categoría se encuentran los sistemas *warehouse* que se verán a continuación, debido a que todos sus componentes mantienen una autonomía de diseño y comunicaciones, y el sistema general tiene autonomía de ejecución.
- Sistemas de integración heterogéneos semiautónomos: Estos sistemas mantienen una autonomía de diseño y comunicación en cada una de las fuentes de datos originales y en el modelo del sistema, sin embargo no mantienen la autonomía de ejecución, ya que la ejecución del sistema global depende de la ejecución de las fuentes de datos originales. Dentro de esta categoría se encuentran los sistemas federados.

2.2.2.1 Evolución en los sistemas de integración

Además de las características de autonomía, heterogeneidad y distribución, la mayoría de los sistemas de integración más utilizados en la industria, ciencia y administración comparten una característica en común, están sujetos a continuos cambios y evolución. Su requisito principal es la capacidad de integrar fuentes de datos y sistemas de manera modular. Los conceptos, arquitecturas y desarrollo de los

sistemas de integración completos, así como de sus componentes deben tener un alto grado de capacidad de cambio y evolución.

Si dividimos el sistema de integración en dos partes, la parte inferior, que contiene a las fuentes de datos y los mecanismos de traducción, y la capa superior, que contiene la capa de integración y presentación al usuario, existen 2 principales razones para los requisitos de evolución:

- En el nivel inferior, habrá una gran cantidad de cambios, a partir de nuevas fuentes de datos y cambios en las antiguas. Estos componentes nuevos o modificados son diseñados en primer lugar con los requisitos locales.
- En el nivel superior, la necesidad de nuevos servicios de información aparecerá tan pronto como el valor de la estructura de la información adicional sea reconocido por un número relevante de usuarios y exista la idea de nuevas fuentes de datos.

A partir de las peticiones se obtienen los requisitos de la capa de integración. En particular debería de ser relativamente sencillo:

- Integrar nuevas fuentes, por ejemplo, para analizar su tipo, estructura y contenido, y hacerlo técnicamente disponible para la vista global usando los *wrappers* apropiados.
- Desarrollar nuevos servicios basados en el conocimiento de la información necesaria en el nivel superior, el conocimiento disponible en el nivel inferior y las relaciones entre ambos.
- Añadir nuevos componentes arquitectónicos en la capa de integración, por ejemplo, reconociendo patrones más generales que *wrapping* e integración y, consecuentemente, reconstruyendo el middleware.

2.2.3 Metadatos para Sistemas de Integración

Un concepto importante para tratar con la heterogeneidad y la distribución es el concepto de metadatos. Los metadatos son datos que describen otros datos o elementos de un sistema que ayudan a su documentación, reusabilidad e interoperabilidad. Como resultado, ayudan a desarrollar soluciones más flexibles.

En el contexto de los sistemas de integración heterogéneos, especialmente en la capa de integración, se pueden distinguir los siguientes tipos de metadatos:

- Metadatos técnicos: Describen información relativa a mecanismos de acceso técnico de componentes, como pueden ser, protocolos, velocidad de conexión, coste de consultas, capacidades de consultas, etc. Se utilizan para conectar la heterogeneidad técnica y la de interfaz.
- Metadatos lógicos: Son los relativos a los esquemas y a sus relaciones lógicas, por ejemplo los diagramas de clases o esquemas relacionales. Las relaciones y dependencias entre varios esquemas de un modelo de datos es un importante tipo de metadato lógico en los sistemas de integración.
- Meta-modelos: Son metadatos que posibilitan la interoperabilidad de esquemas en diferentes modelos de datos.
- Metadatos semánticos: Representación de información que ayuda a describir conceptos semánticos. En particular, ontologías y tesauros se usan para este propósito. Todas las descripciones específicas de dominio pertenecen a esta clase.
- Metadatos relacionados con la calidad: Describen propiedades específicas de la fuente de datos de los sistemas de información relativos a su calidad, rendimiento, etc. Se usan principalmente para optimización.
- Metadatos de infraestructura: Posibilita a los usuarios encontrar datos relevantes. Esto incluye ayuda contextual a los usuarios, estructura de tesauros, etc. Los datos de infraestructura son usados por los usuarios o por la capa de presentación del sistema de integración en los sistemas de información.
- Metadatos relacionados con los usuarios: Describen responsabilidades y preferencias de usuarios en los sistemas de información (perfiles).

Un sistema de integración heterogéneo normalmente usa metadatos en algunos aspectos para ganar flexibilidad. La descripción de componentes de un sistema de información por un conjunto predefinido de metadatos, permite desarrollar métodos genéricos para acceder a estos componentes. En todas las capas de un sistema de integración se hacen necesarios estos metadatos que suelen estar almacenados en un repositorio común.

2.3 Criterios de clasificación para sistema de integración

En el punto anterior se han identificado características comunes de los componentes de los sistemas de integración. Dichos componente suelen seguir estructuras de desarrollo heterogéneas. Es tarea de la capa de integración hacer frente a estos tipos de heterogeneidad.

En este punto se expondrán los diferentes criterios que distinguen las diferentes aproximaciones que puede seguir un sistema de integración. Por ejemplo, se puede distinguir un sistema de integración por el tipo de componentes que puede integrar, el nivel de trasparencia que tiene de cara al usuario, el grado de integración semántica o por la metodología de desarrollo. Aunque es deseable, es imposible proporcionar un criterio ortogonal. Por lo tanto, algunos criterios de clasificación que usan una dimensión podrían requerir o inducir alguna dimensión adicional. Por ejemplo, la transparencia del esquema requiere la existencia de un sistema integrado, las integraciones más flexibles requieren de una construcción top-down y los métodos de consulta basados en recuperación de información previenen de una integración semántica muy cerrada.

2.3.1 Tipos de componentes

Los sistemas de integración pueden permitir o no la integración de datos estructurados, semi-estructurados y no-estructurados. Las fuentes estructuradas tienen un esquema predefinido. Todos los datos están definidos formalmente por un elemento del esquema al que pertenecen. Además, el esquema indica el formato de todos los datos. Los datos que no encajan en un esquema no pueden ser integrados en un conjunto de datos.

Una fuente de datos semi-estructurada tiene una estructura, pero esta no está predefinida en forma de un esquema estricto (Buneman, 1997). Esto es, cada dato tiene su propia definición semántica, normalmente dada en forma de etiqueta. En este sentido, la suma de todas las etiquetas de una fuente de datos semi-estructurada puede ser considerada como un esquema. Esto también implica que el esquema puede cambiar cada vez que se añaden nuevos datos, mientras que en las fuentes de datos estructuradas los cambios de esquema ocurren con menor frecuencia.

Las fuentes de datos no estructuradas no tienen ningún tipo de estructura o esquema, por ejemplo los documentos de texto.

2.3.2 Integración abierta / cerrada

Una de las maneras de clasificar la capa integración podría centrarse en si es una integración cerrada (*tight*) o una integración abierta (*loose*). Una integración cerrada tiene un esquema global unificado, que además es el esquema de acceso de todos los usuarios. Por el contrario, una integración abierta no tiene ese esquema global. En lugar de esto ofrece un lenguaje de consulta unificado para los datos de los componentes. Por un lado, la integración cerrada ofrece un esquema, un lenguaje y un interfaz transparente, mientras que una integración abierta ofrece sólo este último.

- Integración cerrada: Una integración cerrada ofrece un esquema unificado (integrado en el esquema global) como acceso del interfaz a la capa de integración (Figura 5). Este esquema puede:
  - Ser construido a través de un sistema de integración automático o semiautomático o ser creado ad-hoc.
  - Cubrir semánticamente los componentes total o parcialmente.

  En cualquier caso, la "esencia semántica" (conceptos de la vida real que cubre el esquema) de un esquema global puede ser un subconjunto de las "esencias semánticas" de los esquemas de los componentes. Esto implica que un esquema global que está cubierto de manera ad-hoc considera el contenido de los componentes para asegurarse de esto, mientras que en un proceso de integración de esquema, este no está definido ni formalizado.

  La principal tarea cuando se usa un sistema de integración es la resolución de la heterogeneidad lógica de los esquemas fuente. Debe ser considerado tanto durante la integración del esquema como durante el procesado de las consultas. Para asegurar la equivalencia semántica, el sistema de integración debe conocer las correspondencias (metadatos lógicos) entre las consultas y los esquemas globales y de las fuentes. Estas correspondencias pueden ser expresadas, por ejemplo, a través de ontologías o reglas. Estas

pueden ser definidas por humanos, por expresiones del lenguaje o incluso inferidas automáticamente.

Las integraciones cerradas son cómodas para los usuarios, ya que no necesitan saber nada de los esquemas de los componentes, únicamente deben conocer el esquema de integración. Por otro lado, los usuarios dependen de mecanismos de traducción y por lo tanto de correspondencias, que deben ser definidas por un experto en el dominio.

El uso de esquemas globales es esencial si se van a integrar muchas fuentes de datos, donde es imposible que los usuarios conozcan todas ellas en profundidad, o si las fuentes modifican su esquema con cierta frecuencia, de manera que el usuario no pueda seguir los cambios. Por otro lado, en algunas situaciones es imprescindible el uso de esquemas globales, especialmente cuando se usan esquemas estándar. En ese caso los componentes deben ajustarse al esquema global.

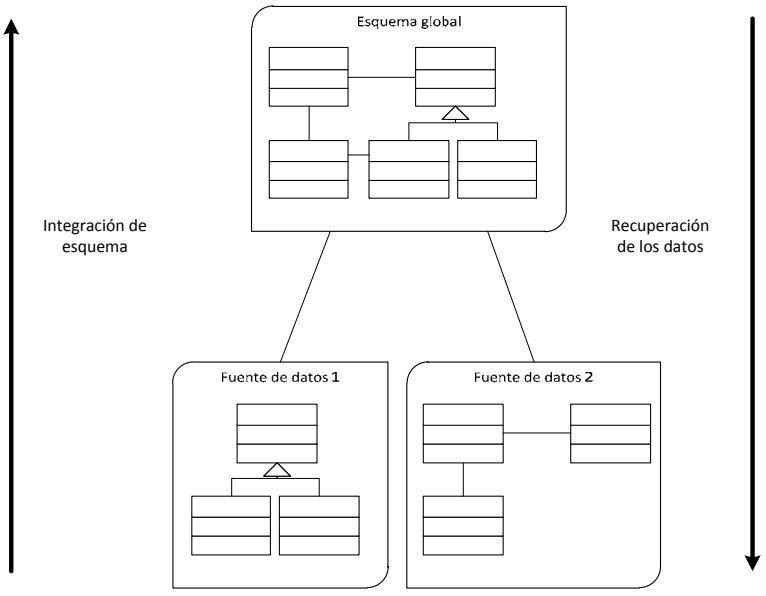


Figura 5: Integración de esquemas cerrada

- Integración abierta

La integración abierta no ofrece un esquema uniforme para las consultas a la capa de integración. En lugar de eso ofrece un sistema de consultas uniforme que abstrae la consulta de los lenguajes de consulta de

cada componente y oculta la heterogeneidad técnica y de lenguaje. Por este motivo, el usuario se hace responsable de manejar la heterogeneidad lógica en los componentes. Este tipo de integración sólo se puede construir si cada una de las fuentes que la componen pueden ser accedidas a través de un lenguaje de consultas. En una integración cerrada la propia capa de integración puede intentar compensar las limitaciones de las fuentes, en cambio en una integración abierta esto es mucho más complicado, debido a que las consultas son ejecutadas directamente por el usuario.

Para que el usuario no tenga todo el peso en el proceso de integración, algunos sistemas utilizan vistas de integración, que permiten al usuario definir vistas en los componentes, de manera que se usan como si fuesen relaciones globales (Figura 6).

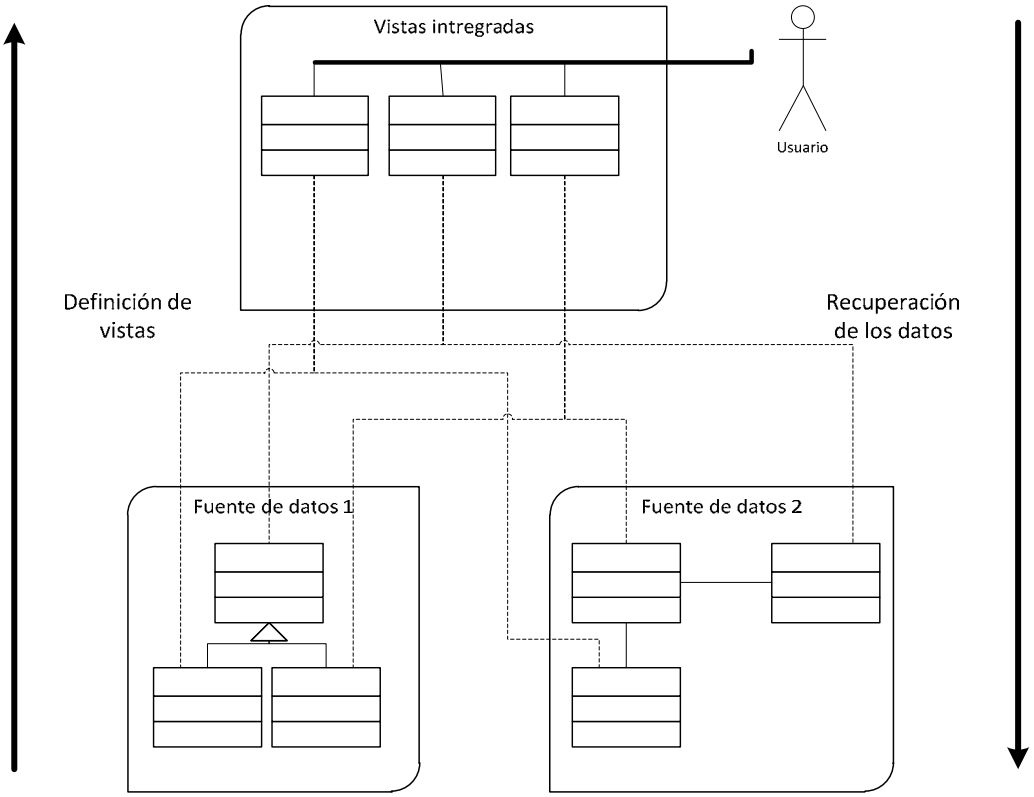


Figura 6: Integración de esquemas abierta

Los sistemas de integración abiertos que utilizan vistas de integración, posiblemente definidas por expertos de dominio, se pueden ver como sistemas de integración cerrados. El criterio para diferenciar un sistema de integración abierto o cerrado sería entonces si los esquemas de los componentes son

visibles o no al usuario. En un sistema cerrado, los esquemas nunca son visibles, en uno abierto, los esquemas son visibles, aunque se pueden utilizar vistas de integración que faciliten su manejo.

2.3.3 Modelo de datos en sistemas de integración

La capa de integración debe basarse en un modelo de datos específico, llamado modelo de datos canónico o modelo de datos común. El esquema de un sistema de integración cerrado es el esquema de este modelo. En un sistema abierto, el sistema de consultas con el que se accede a las fuentes de datos se basa parcialmente en este modelo de datos.

El modelo de datos restringe los tipos de componentes que pueden integrarse en el sistema de integración, debido a las posibles pérdidas de traducción entre algunos tipos de modelo. Por ejemplo, la integración de modelos objetuales en esquemas relacionales sólo es posible asumiendo una pérdida de conocimiento semántico. Si se hace al contrario (de relacional a objetual), se requiere un proceso de enriquecimiento semántico para obtener una correcta representación.

2.3.4 Tipos de integración semántica

La capa de integración encapsula fuentes de datos para el usuario. En contraste con un sistema de integración abierto, donde los datos se recogen sin cambios de cada fuente, los servicios de integración de un sistema de integración cerrado deben mantener una mejor integración semántica. A nivel de datos se pueden distinguir los siguientes tipos de integración semántica:

- Colección: Los datos de los componentes se obtienen sin cambios, sin buscar equivalencias entre objetos de diferentes fuentes.
- Fusión: La integración de los datos de un componente se realiza por medio de una extracción (expresada en forma de consulta contra el esquema del componente), en contraste con la aproximación de colección, la fusión de objetos se realiza para identificar entidades semánticamente equivalentes provenientes de diferentes fuentes. El sistema de integración intenta determinar una representación coherente (por ejemplo, si el sistema detecta valores contradictorios para el mismo objeto, intenta eliminar el conflicto

usando reglas). Este tipo de integración es muy compleja, frecuentemente es imposible identificar objetos o decidir qué valor de dato es el correcto.

- Abstracción: Utilizando la abstracción, se obtienen los datos integrados basándose en los datos obtenidos de los componentes, pero posteriormente se deben aplicar métodos para saber qué dato debe pasar desde los componentes a la capa de integración. La necesidad de abstracción es causada generalmente por conflictos semánticos. Esto acompaña funciones para agregar datos, reclasificar entidades o incluso aplicar procesos complejos de razonamiento.
- Suplementación: Los datos no sólo se derivan de los datos de componentes, sino que se añaden otros datos que describen el contenido o el contexto semántico de los datos (metadatos semánticos). Este tipo de integración se usa para manejar la semántica implícita de los componentes. Es necesaria, por ejemplo, cuando las fuentes de datos no proporcionan esquemas, pero la capa de integración se basa en esquemas de metadatos.

2.3.5 Transparencia

La transparencia para el usuario es el principal objeto de la integración. Un sistema de integración perfecto debería de proporcionar al usuario la sensación de que interactúa con un solo sistema central, homogéneo y consistente. Se pueden distinguir los siguientes tipos de transparencia:

- Transparencia de localización: Los usuarios no necesitan saber cuál es la localización física de la información.
- Transparencia del esquema: Los usuarios no necesitan saber las diferentes denominaciones que tienen las entidades o atributos en diferentes fuentes de datos. En un escenario relacional, los usuarios no deben preocuparse de los diferentes nombres de las relaciones y los atributos. En otras palabras, la transparencia de esquema enmascara los conflictos lógicos.
- Transparencia de lenguaje: Los usuarios no necesitan tratar con diferentes mecanismos, ni lenguajes de consulta. Esto implica tanto el lenguaje de consulta, el modelo de datos como los mecanismos de acceso.

Existe una clara relación entre el tratamiento de la heterogeneidad y el nivel de trasparencia que ofrece un sistema de integración. La transparencia de esquema se reduce a ocultar la heterogeneidad lógica, mientras que la transparencia del lenguaje se consigue ocultando la heterogeneidad de interfaz. La transparencia de localización está relacionada con la heterogeneidad técnica.

2.3.6 Paradigma de consultas

Se podrían dividir los sistemas de integración por el tipo de consultas que permiten: consultas estructuradas o consultas IR (recuperación de información). Mientras que las primeras asumen alguna estructura en la información que se usa para especificar los datos de la consulta, la segunda realiza búsquedas de similaridad en documentos. Podría existir un tercer tipo de paradigma de consultas que se usa para metadatos específicos que describen objetos, que no son datos en sí mismos. Por ejemplo, la búsqueda de documentos puede permitir criterios de búsqueda como el tamaño del documento o su fecha, que no están relacionados con el contenido del documento.

2.3.7 Desarrollo Botom-up vs Top-down

En un sistema de integración cerrado, se pueden distinguir dos tipos de arquitecturas: *Top-down*, por ejemplo, comenzando por la información global necesaria y, posteriormente, añadiendo fuentes que pueden contribuir con las necesidades del problema, o *bottom-up*, que comienza por los requisitos de integración de una serie de fuentes.

2.3.7.1 Estrategia top-down

Las aproximaciones *top-down* se construyen de acuerdo a las necesidades globales de información. Por ejemplo, una compañía quiere ofrecer un servicio que busque los precios de libros más bajos de diferentes tiendas en internet, o un sistema de apoyo a la toma de decisiones que quiere integrar cierta información sobre clientes que está distribuida sobre las bases de datos de diferentes departamentos. En estos casos el esquema de componentes no es importante en el diseño del esquema global. Partiendo de los cuatro requisitos para la integración de un esquema (integridad, exactitud, comprensibilidad y minimalidad) (Batini et al., 1986), dos de ellos no se

aplican. En primer lugar no es necesario incluir todos los esquemas completamente, si lo único necesario son los datos de los clientes (integridad). En segundo lugar, no es necesario representar los datos del nivel global exactamente como aparecen en los componentes, si sólo son necesarios datos derivados o abstraídos (exactitud, el *mapping* se vuelve unidireccional). Por ejemplo, alguien podría decidir agrupar a los clientes por grupos de salario y no necesita almacenar el salario concreto.

El esquema global en las aproximaciones *top-down* puede ser generado tanto ad-hoc, como ser el resultado de un proceso más formal de análisis, comenzando por las necesidades de casos de uso y terminando por las técnicas de integración de vistas. El esquema global debe seguir un estándar. En cualquier caso, el esquema de los componentes se considera únicamente en una segunda fase, cuando se establecen las correspondencias entre los esquemas globales y los de las fuentes para permitir la traducción de las consultas.

Las aproximaciones *top-down* tienen muchas ventajas en escenarios donde las fuentes cambian continuamente, si las fuentes aparecen y desaparecen a menudo, si la integración del esquema es inviable o demasiado costosa, o si los requisitos globales están cambiando continuamente. Estas son las razones por las que la integración basada en esquemas o *bottom-up* es extremadamente vulnerable a cualquier cambio. En cualquier caso, las aproximaciones *top-down* son normalmente menos cerradas que las *bottom-up*

2.3.7.2 Estrategias bottom-up

La construcción de un sistema de integración cerrado *bottom-up* significa que uno de los requisitos iniciales es la necesidad de tener acceso integrado a ciertas fuentes de datos. Un escenario típico es la necesidad de tener acceso detallado e uniforme a todas las bases de datos de una empresa para construir aplicaciones globales. El sistema por lo tanto proporcionaría acceso (integral y exacto) a todas las bases de datos, mientras que en la aproximación *top-down* se buscaba únicamente el acceso a todos los datos de clientes, (sin importar donde estuviese localizados). En este tipo de esquemas existe una gran necesidad de garantizar la integridad y la exactitud del esquema integrado, por lo tanto son más apropiadas técnicas formales

de integración. La integración *bottom-up* lleva a un sistema bien integrado semánticamente porque se asume que los componentes son conocidos antes del proceso de integración. Un cambio en la configuración requiere un nuevo proceso de integración.

Un problema particular en la construcción de un sistema de integración es la necesidad de actualizar datos en las fuentes a través del esquema global. Las actualizaciones son sólo posibles si las conexiones entre el esquema global y los esquemas de los componentes son muy cerradas.

2.3.8 Integración material vs integración virtual

El proceso de integración puede diferenciarse en si los datos de los componentes se persisten en la capa de integración. En las arquitecturas de integración virtual los resultados de una consulta no se almacenan. Esto requiere unos mecanismos que traduzcan las consultas contra el esquema de integración en una o más consultas contra los componentes que son propagadas dinámicamente a las fuentes, este esquema de integración virtual se conoce como modelo federado.

En el otro extremo están las arquitecturas que materializan todo o parte de las fuentes en el nivel de integración. Estas son conocidas como *data warehouses*.

Esta materialización tiene las siguientes ventajas:

- Alto rendimiento de las consultas contra los datos almacenados.
- Control sobre los datos almacenados, haciendo posible la curación especializada de los datos.

Y desventajas:

- Mantener al día los datos requiere un complejo proceso de actualización.
- La capa federación debe almacenar una gran cantidad de datos.

2.3.9 Acceso lectura-escritura o solo-lectura

Se puede distinguir entre un sistema de integración que permita la creación o actualización de datos en un componente a través de la capa de integración y aquellos

que no lo permiten. Normalmente en los proyectos de integración no se permite acceso de escritura por lo siguiente:

- Muchos interfaces (ej. Servicios web) no permiten escritura.
- La escritura a través de vistas de integración se topa con el problema de la actualización de datos sobre vistas (Barsalou et al., 1991).
- La escritura a través de un esquema integrado puede acarrear el problema de no saber en qué fuente escribir, si el dato está presente en más de una fuente.
- Las transacciones globales requieren protocolos complejos.

2.3.10 Métodos de acceso

En este apartado se verán los diferentes tipos de acceso a los componentes del sistema (o sus *wrappers*):

- Acceso a través de lenguajes de consulta (ej. SQL). Este acceso puede datos, a través de su interfaz nativo, de OJDB, JDCBC u otros APIs especializados.
- Acceso a través de consultas parametrizadas. Las consultas parametrizadas son consultas predefinidas en las que sólo pueden modificarse ciertos parámetros. Ejemplos de este tipo de acceso son las llamadas a procedimiento remoto o servicios web.
- Acceso a través de http. Especialmente en entornos web, los datos pueden ser expuestos en algún formato de etiquetado.

Los diferentes métodos de acceso presentan dos tipos de heterogeneidad, la heterogeneidad técnica y la heterogeneidad de acceso al lenguaje. Desde el punto de vista lógico, sólo importa la heterogeneidad de acceso al lenguaje.

2.4 Arquitectura de los sistemas de integración

2.4.1 Sistemas *warehouse*

Los sistemas *warehouse* consisten en un conjunto de sistemas, arquitecturas y algoritmos que permiten seleccionar datos de múltiples bases de datos e incorporarlas a un único repositorio denominado *data warehouse* (Widom, 1995, Connolly and Begg, 2005), sobre el que posteriormente el usuario puede realizar consultas y análisis.

En general, en integración de datos distribuidos homogéneos, existen dos enfoques. En el primero de ellos, los sistemas federados se basan en aceptar una consulta, determinar el conjunto de fuentes de información que pueden contener los datos deseados y generar las consultas apropiadas para cada fuente de datos. Posteriormente, una vez recuperados los datos de las fuentes de datos originales, se realiza la traducción, filtrado y unión de la información en tiempo real y se devuelve al usuario que realizó la consulta.

En la segunda aproximación, los sistemas basados en *warehouse*, la información de cada fuente se extrae con anterioridad a que estos datos sean requeridos, se traduce y se filtra apropiadamente, se une con información que ya está almacenada en el *warehouse* de manera apropiada y se almacena en un repositorio centralizado. Cuando se realiza una consulta, esta consulta se ejecuta directamente contra el repositorio, sin necesidad de volver a acceder a las fuentes de datos originales.

Esta aproximación se conoce como *data warehouse*, porque el repositorio se comporta como un almacén (en inglés *warehouse*) para guardar los datos de interés. Según la clasificación vista anteriormente, un sistema *warehouse* es un sistema de integración de fuentes heterogéneas y distribuidas, que presenta un esquema de integración cerrada.

Según Bill Inmon, conocido como el padre de los almacenes de datos, un almacén de datos "es una colección de datos clasificada por temas, integrada, variable en el tiempo y no volátil que se utiliza como ayuda en el proceso de toma de decisiones por parte de quienes dirigen una organización" (Inmon, 1992).

En esta definición los datos están clasificados por temas ya que el *warehouse* está organizado de acuerdo con los temas que más importancia tienen para la organización (ej. Clientes, productos, ventas) en lugar de organizarse por áreas de aplicación (facturación, control de almacén o pedidos). Esto se refleja en la necesidad almacenar datos de ayuda a la toma de decisiones en lugar de datos orientados a aplicaciones.

Cuando dice que los datos están integrados se refiere a la mezcla de datos procedentes de diferentes sistemas de aplicación utilizados dentro de la organización. Los datos de origen son a menudo incoherentes utilizando, por ejemplo, diferentes formatos. El *warehouse* integrado de datos debe volver a dotarse de coherencia para presentar una vista unificada de los datos a los usuarios.

Se dice que son variables en el tiempo porque los datos del *warehouse* sólo son precisos y válidos en algún instante temporal o a lo largo de un cierto intervalo de tiempo. La variación con respecto al tiempo del almacén de datos también se refleja en el gran intervalo de tiempo durante el que se almacenan los datos, en la asociación implícita o explícita de la variable temporal con todos los datos y en el hecho de que los datos representan una serie de instantáneas.

Finalmente, se dice que los datos son no volátiles, ya que los datos no se actualizan en tiempo real, sino que se refrescan de forma periódica a partir de los sistemas operacionales. Los nuevos datos se añaden siempre para aumentar la base de datos, en lugar de para sustituir la información ya existente. La base de datos absorbe continuamente estos nuevos datos, integrándolos incrementalmente con los datos anteriores.

La importancia del modelo de *data warehouse* nace de la necesidad de las empresas de poder almacenar toda la información relacionada con la organización en un lugar centralizado para poder realizar análisis y de la necesidad de poder desacoplar esos análisis de los sistemas de transacciones online. El proceso de análisis involucra consultas muy complejas y pocas o ninguna actualización y suele ser el principal uso de estos sistemas.

En esta aproximación, la información integrada está disponible para consulta y análisis de manera inmediata, por lo que es especialmente útil para clientes que requieren porciones específicas y predecibles de la información disponible, clientes que requieren gran rendimiento en la recuperación de la información, pero que no requieren necesariamente que la información esté completamente actualizada. También es útil para entornos en los que las aplicaciones requieran alto rendimiento o

en el que los clientes quieran copias privadas de la información, para poder anotarla, resumirla, etc.

2.4.1.1 Arquitectura de un sistema de data warehouse

En la Figura 7 se muestran las bases de la arquitectura del *data warehouse*. En la parte inferior de la figura se muestran las fuentes de datos. Aunque están representadas como bases de datos, estas fuentes de datos pueden ser sistemas heterogéneos, como un fichero, una página web o un servicio web. Conectado a estas fuentes de información se encuentran los *wrappers* y monitores. Los *wrappers* son los responsables de traducir la información del formato nativo de la fuente de dato al formato y modelo de datos del *warehouse*, mientras que los monitores son los componentes responsables de detectar automáticamente cambios de interés en la fuente de datos y enviárselos al integrador.

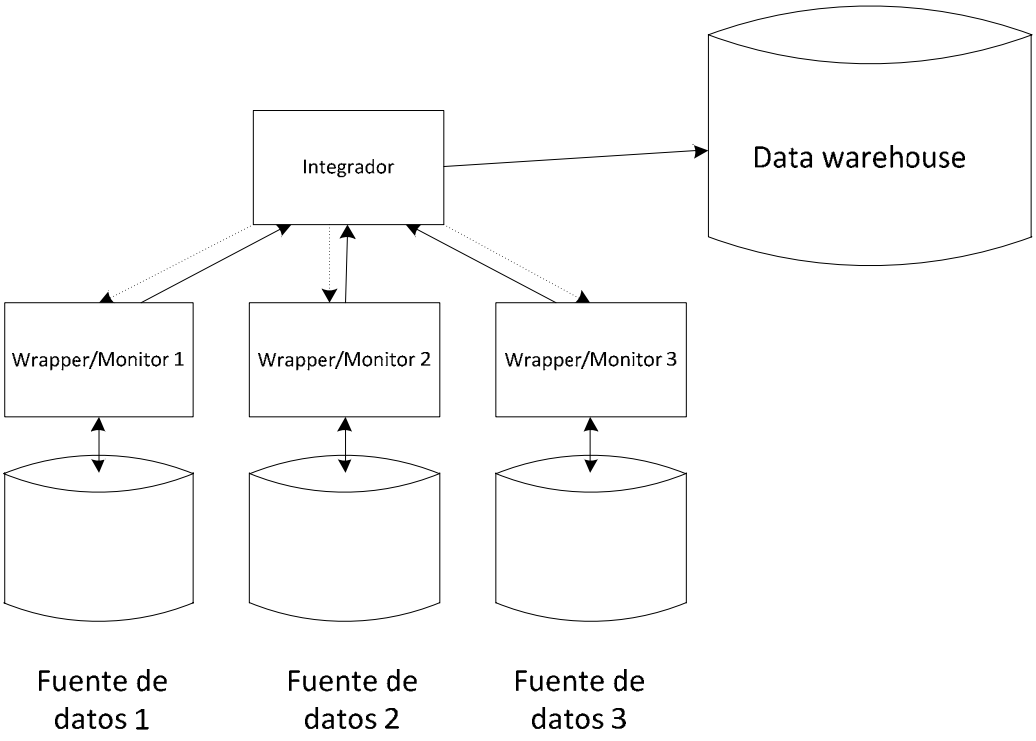


Figura 7: Esquema de data warehouse

Cuando se incluye una nueva fuente de datos al *warehouse*, o cuando una fuente de información relevante cambia, los datos nuevos o modificados llegan al integrador. El integrador es responsable de almacenar la información dentro del

warehouse, que puede incluir alguna etapa de filtrado o resumen y la unión de los datos nuevos con los que ya estaban en el *warehouse* o con datos de otras fuentes. Con el objetivo de integrar correctamente la nueva información, el integrador debe obtener información adicional de las fuentes de información. Esta consulta se muestra en la figura como la flecha punteada.

El *warehouse* puede utilizar un solo sistema gestor de bases de datos o puede usar un sistema distribuido.

La arquitectura y la funcionalidad descrita es la más general y es, por tanto, la que proporcionan la mayoría de los sistemas *warehouse* comerciales. Concretamente, los sistemas *warehouse* asumen que las fuentes y los *warehouse* utilizan un solo modelo de datos, normalmente relacional, que la propagación de la información desde las fuentes de datos al *warehouse* se hace off-line (procesado por lotes) y que las consultas desde el integrador hasta las fuentes de datos normalmente no son necesarias.

A continuación se analizarán en mayor profundidad los módulos de la arquitectura.

2.4.1.1.1 Wrapper y monitores

Los *wrappers/monitores* tienen dos responsabilidades principales que están interrelacionadas:

- Traducción: Consiste en hacer que la información que está en las fuentes de datos aparezca ante el *warehouse* como si tuviese el mismo modelo de datos. Por ejemplo, si las fuentes de información son ficheros planos, pero el modelo de datos del *warehouse* es relacional, entonces el *wrapper*/monitor debe crear un interfaz que presente los datos al *warehouse* como si estos fuesen relacionales. El problema de la traducción es inherente a cualquier sistema de integración. Normalmente el componente que traduce la información a un modelo de integración común se denomina *wrapper*.

- Detección de cambios: Consiste en monitorizar las fuentes de información, con el objetivo de detectar cambios en los datos originales que puedan ser relevantes para el *warehouse* y propagar estos cambios al modulo integrador. Esta funcionalidad depende de la traducción, debido a que los datos nuevos deben ser traducidos del formato y modelo de la fuente de información al formato y modelo de *warehouse*.

En ocasiones, para solucionar los problemas que puede causar la detección de cambios, simplemente se propagan copias completas de los datos relevantes desde las fuentes de información al *warehouse* periódicamente. El integrador puede combinar estos nuevos datos con los datos del *warehouse*. Esto solo puede ser aceptable en ciertos escenarios, en los que el *warehouse* pueda estar off-line en algún momento para hacer este tipo de cambios. En caso de que sea necesaria concurrencia, eficiencia y acceso continuo, es necesaria la detección de cambios.

Es importante resaltar que es necesario un *wrapper* distinto para cada tipo de fuente de información, ya que la funcionalidad de este *wrapper* depende del tipo de fuente (base de datos, fichero plano, servicio web, etc.), tanto como de los datos de la fuente. A pesar de que sería preferible no tener que codificar un *wrapper* para cada fuente de datos que participa en el *warehouse*, suele ser necesario. Existen diversas líneas de investigación para proporcionar *wrappers* automáticos o semiautomáticos.

2.4.1.1.2 Integrador

Asumiendo que los datos originales del *warehouse* han sido cargados desde las fuentes de datos originales, el trabajo del integrador es recibir las notificaciones de cambio de las fuentes de datos por parte del monitor y reflejar esos cambios en el *warehouse*.

A cierto nivel abstracto, los datos en el *warehouse* pueden ser vistos como un conjunto de vistas o vista materializada, donde los datos base residen en las fuentes de información. Si se enfoca el *warehouse* de esta manera, el trabajo del integrador es básicamente desarrollar un mantenimiento de estas vistas materializadas. Sin embargo, existen ciertas razones por las que las técnicas de mantenimiento de vistas convencionales no pueden usarse:

- En la mayoría de los escenarios donde se aplica un data *warehouse*, las vistas tienden a ser más complejas que las vistas convencionales. Por ejemplo, incluso si el *warehouse* y las fuentes de información son relacionales, las vistas almacenadas en el *warehouse* pueden no ser expresables usando un lenguaje relacional como SQL sobre los datos base. Normalmente, los *warehouse* suelen contener una gran cantidad de datos históricos, mientras que las fuentes originales no suelen mantener este tipo de información. Por lo tanto, las vistas del *warehouse* no deben establecerse en función de los datos base originales, sino en función de la historia de los datos.
- Las fuentes originales de datos actualizan sus bases de datos independientemente del *warehouse* donde sus vistas están almacenadas, y los datos originales suelen venir de sistemas que no pueden o no quieren participar en el mantenimiento de las vistas. La mayoría de las técnicas de mantenimiento de vistas materializadas se basan en las actualizaciones de los datos base y la modificación de las vistas aparece en la misma transacción que las actualizaciones. En un escenario de *warehouse* se produce lo siguiente:
    - El sistema de mantenimiento de vistas (integrador) está poco acoplado con los sistemas que gestionan los datos base (las fuentes de datos originales)
    - Las fuentes de datos originales no participan en el mantenimiento de vistas, sino que simplemente notifican los cambios.
    - Algunas fuentes no proporcionan capacidad de bloqueo lo que imposibilita un sistema común de transacciones.
- En un *data warehouse*, las vistas pueden no necesitar actualizarse después de cada modificación. En lugar de eso, pueden realizarse grandes actualizaciones periódicas, en cuyo caso las técnicas de mantenimiento de las vistas deben ser eficientes.
- En un entorno de *data warehouse* puede ser necesario transformar los datos originales de las fuentes de datos antes de que los datos se

integren en el *warehouse*. Estas transformaciones pueden incluir, por ejemplo agregaciones de datos, resúmenes, descartar o corregir datos, insertar valores por defecto o eliminar duplicidades e inconsistencias.

2.4.2 Sistemas de integración federados

En la Figura 8 se muestra la arquitectura en 3 capas de los sistemas de integración federados. Las aplicaciones y los usuarios acceden a un conjunto de fuentes de datos heterogéneos a través de la capa de federación, que es un componente software que ofrece una manera uniforme de acceder a los datos almacenados en las fuentes de datos. La uniformidad se alcanza con una estrategia de interoperabilidad (por ejemplo, esta capa ofrece un esquema federado, un sistema de consulta uniforme y una descripción uniforme de las fuentes de datos y los contenidos a través de conjuntos de metadatos). Las fuentes de datos normalmente se integran en la infraestructura utilizando *wrappers* que resuelven aspectos técnicos de la integración.

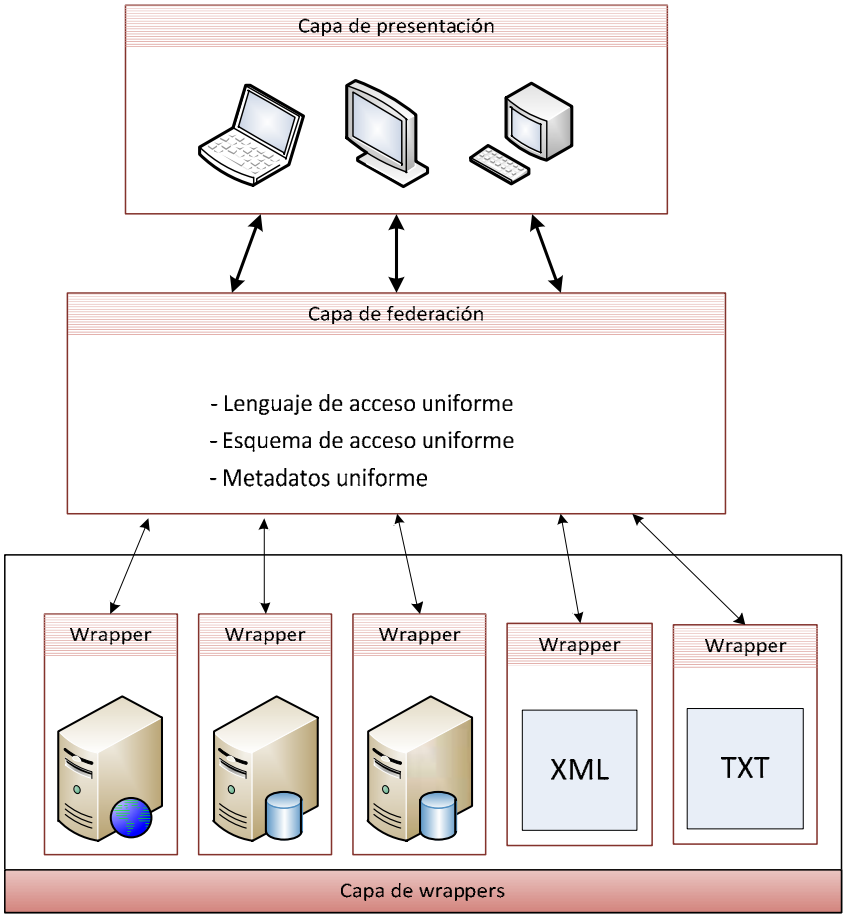


Figura 8: Tres capas de un sistema federado

Un sistema de integración federado se construye normalmente a partir de un conjunto de componentes distribuidos, heterogéneos y semi-autónomos, aunque si los componentes son muy homogéneos aún se puede hablar de Sistema de bases de datos distribuidas. En este sentido, no está claro el grado de heterogeneidad que los componentes deben alcanzar para definir una frontera entre sistema distribuido y sistema federado, de la misma manera que tampoco existe una frontera clara entre los sistemas que usan una BD y los sistemas que no la usan. Existen sistemas que utilizan colecciones de ficheros planos muy bien estructurados que presentan sistemas de consulta muy potentes que pueden ser tratados como un sistema de base de datos (por ejemplo las bases de datos NCBI). Po otro lado, algunos sistemas de bases de datos permiten datos complejos (BLOBs) que no son compatibles con los sistemas clásicos de consulta.

2.4.2.1 Sistemas de bases de datos federados

Este tipo de sistemas proporciona las funciones clásicas de un sistema gestor de bases de datos, lo que incluye acceso de lectura-escritura. Los componentes de los sistemas de bases de datos federados tienen fuentes estructuradas, a los que puede accederse a través de un lenguaje de consultas. Estos componentes suelen renunciar a cierta autonomía en cuestión de notificación de cambios, acceso a metadatos, etc.

Los sistemas de bases de datos federados están altamente acoplados. Están construidos usando la arquitectura *bottom-up* y aplicando técnicas de integración de esquema (Figura 9). Los esquemas federados deben cumplir los requisitos de integridad, corrección, minimalismo y comprensibidad (Batini et al., 1986), lo que únicamente es posible usando integración por colección o fusión.

Como cualquier otro sistema altamente acoplado los SBDF ofrecen completa trasparencia de localización y esquema a sus usuarios. Pero los SBDF normalmente tienen una arquitectura estática con problemas en la evolución del sistema debido a la dependencia del esquema en los procesos de integración, que no permite que ningún componente pueda salir de la federación.

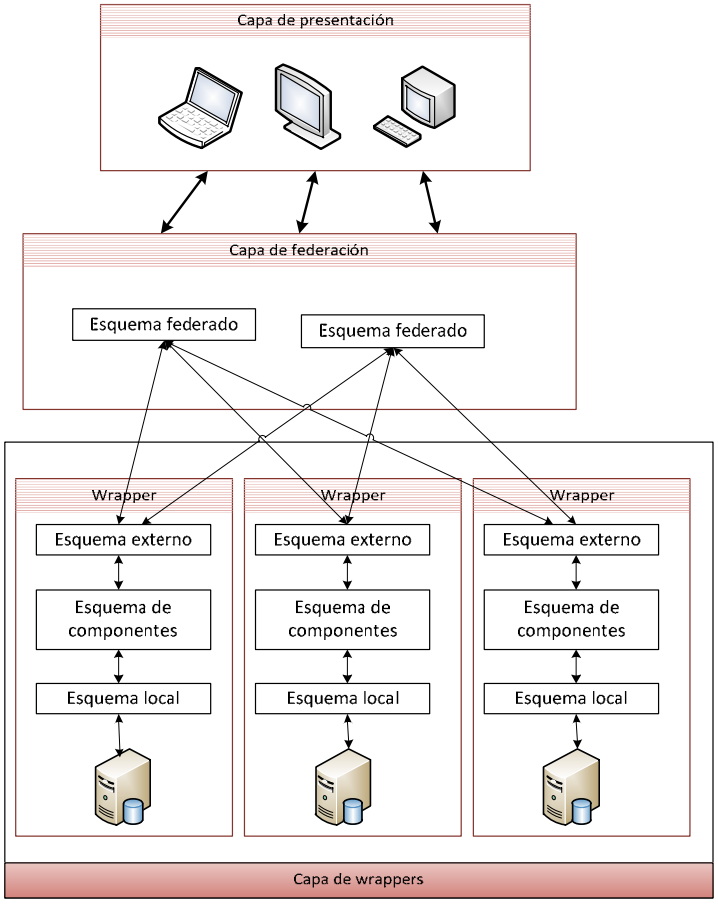


Figura 9: Arquitectura sistema de bases de datos federada

2.4.2.2 *Sistemas federados basados en mediador*

El termino mediador fue introducido por Wiederhold (Wiederhold, 1992) y se ha usado en una gran cantidad de publicaciones o proyectos sobre integración de datos. Desde sus primeros usos, no ha habido una definición de mediador ni de su relación con un sistema de base de datos federado. En generar un mediador debe ser un componente software que media entre un usuario y las fuentes de datos originales. Debe de ser un componente ligero, flexible y reusable. En particular, los mediadores pueden usarse de manera jerárquica, usando otros mediadores como fuentes de datos.

Una de las diferencias obvias entre los sistemas de bases de datos federadas y los sistemas mediador es que estos últimos presentan acceso de solo lectura. Los sistemas mediadores están altamente acoplados, por lo que se usa un esquema federado para proporcionar acceso a los datos de diferentes componentes (heterogeneidad semántica). En contraste con los sistemas de bases de datos

federados, el esquema federado con mediador usa una arquitectura *top-down* de acuerdo con los requisitos del sistema. Los sistemas basados en mediador pueden ser entendidos como servicios que se ofrecen a los consumidores. Esto implica un importante requisito de flexibilidad relacionado con la evolución del sistema. Debe ser posible eliminar o añadir componentes al sistema federado, debido a que estos componentes suelen mantener autonomía de comunicación.

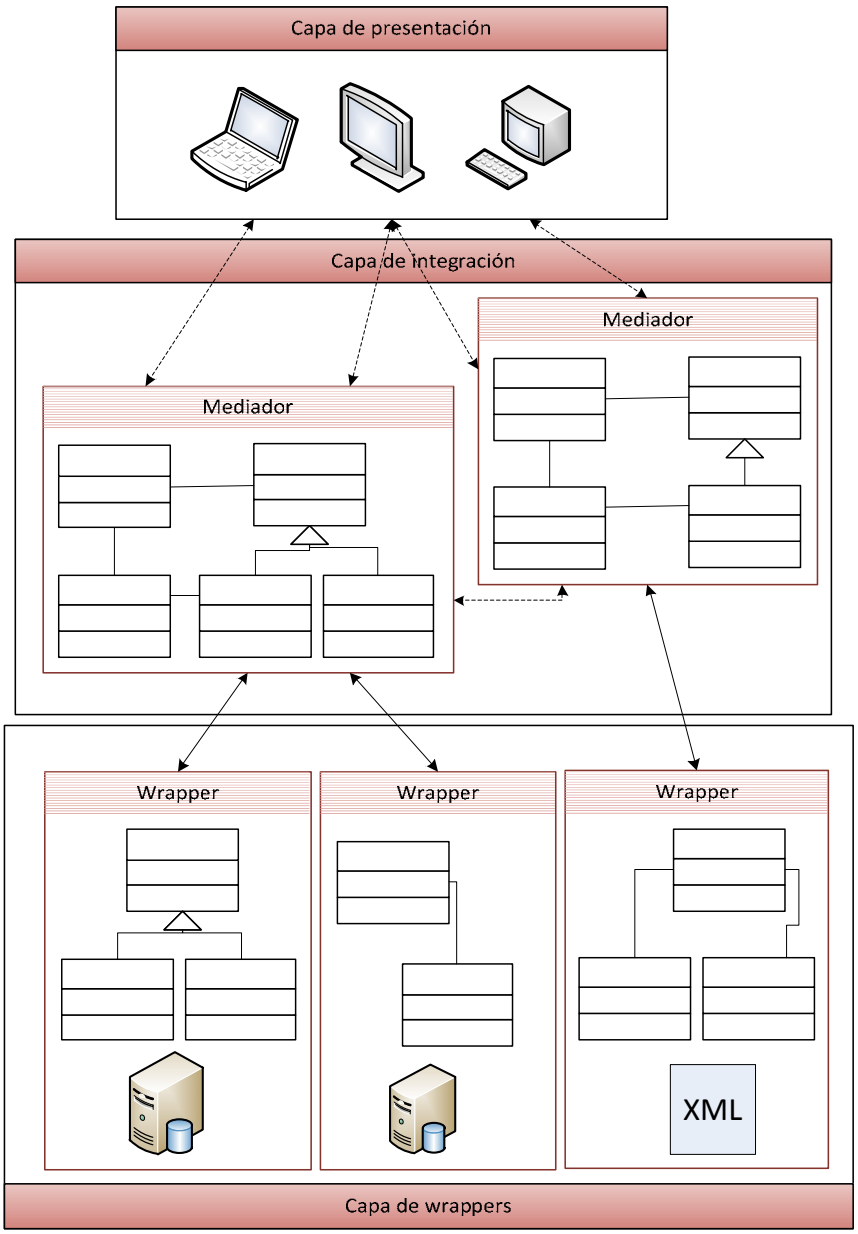


Figura 10: Arquitectura mediador

En la Figura 10 se muestra una arquitectura clásica *wrapper/mediator* adaptada de las figuras de los sistemas de integración anteriores. La capa de

integración contiene varios mediadores que proporcionan los servicios de mediación. En este modelo se denominará a la capa de integración capa mediadora. Cada mediador tiene su propio esquema federado y los mediadores pueden usar otros mediadores como fuentes de datos (redes de mediadores o mediadores jerárquicos). Los *wrappers* ocultan la heterogeneidad técnica y de modelo de datos. Cómo estas acceden a las fuentes de datos originales es transparente al mediador. Las consultas contra el modelo general (consultas de usuario se muestran con flechas discontinuas mientras que las consultas contra los *wrappers* como flechas continuas.

2.4.2.2.1 Tratamiento de consultas

En este apartado se mostrarán los distintos tipos de procesos para responder una consulta al esquema global, transformándola en consultas contra otros esquemas, en presencia de conflictos semánticos, estructurales y esquemáticos entre ambos esquemas.

En muy pocas ocasiones las consultas al esquema global pueden ser respondidas recuperando información de una única fuente. En la mayoría de los casos, el mediador debe encontrar las combinaciones de consultas correctas a las fuentes de datos originales, de manera que respondan correctamente la consulta original. Se denomina plan de consultas a la combinación de consultas que responden correctamente la consulta original. Este plan se dice que es correcto cuando obtiene respuestas semánticamente correctas. La mediación de consultas consiste en los siguientes pasos:

- Planificación de consultas es el proceso de buscar el plan de consultas correcto para una consulta dada al esquema global. La planificación de consultas debe estar basada en correspondencias predefinidas entre las consultas o conceptos entre los diferentes esquemas y debe tener en cuenta las restricciones de consulta de las diversas fuentes.
- El plan de ejecución es el proceso de ejecutar el plan. Este proceso está compuesto de los pasos de optimización, que decide qué pasos de la consulta se ejecutan en cada componente, la división en las

subconsultas y el postprocesado necesario para una futura integración de los resultados.

- La integración de los resultados finalmente homogeniza los datos obtenidos, eliminando redundancia, identificando objetos idénticos y resolviendo valores inconsistentes.

2.4.2.2.2 Planificación de consultas

El plan de consultas debe estar basado en alguna descripción del contenido de la fuente con respecto al esquema global, por ejemplo, la manera en como las vistas están definidas en SQL. Los lenguajes para expresar estas correspondencias se denominan Correspondece Specification Languages (CSL) (Lenzerini, 2002).

El plan de consultas debe encontrar el conjunto de subconsultas que explote el conocimiento semántico que está expresado en reglas en cierto CSL. Encontrar estas correspondencias es normalmente tarea de un operador humano, ya que codifica relaciones semánticas entre conceptos. Existen también trabajos de correspondencia automática de esquemas (Rahm and Bernstein, 2001) (Shvaiko and Euzenat, 2005).

Existen dos clases de CSL's. Siguiendo el paradigma GlobalAsView (GaV), el esquema global se define teniendo una o más vistas sobre los esquemas de las fuentes para cada clase. Cada regla de correspondencia tiene una sola clase global en un lado de la regla y define su equivalencia semántica a una consulta a la fuente original en el otro lado de la regla. La situación inversa es la del paradigma LocalAsView (LaV), donde las clases del esquema local se describen a partir de unas vistas en el esquema global (Hull, 1997). En este caso también cada regla tiene una clase y una consulta, pero en este caso, la clase es local y la consulta es global. También existe el llamado Both As View o Global Local As View (Friedman et al., 1999), donde las relaciones entre el esquema global y los fuentes se realizan haciendo uso tanto del LaV como de GaV. El *mapping* GLaV está formado por reglas, donde el antecedente qs es una consulta conjuntiva sobre el esquema local, mientras que el consecuente qc es una consulta conjuntiva sobre el esquema global.

Aparte de las implicaciones técnicas, la mayor diferencia es la percepción del esquema global, mientras que en la aproximación GaV se ve el esquema global como

algo artificial que debe ser rellenado accediendo a las fuentes, la aproximación LaV asume cada fuente como una parte del espacio de información global.

La traducción Global As View requiere la expansión de las clases de la consulta de usuario en una correspondencia en las consultas de fuentes. Este paso de expansión clase global -> consulta local está codificada en la definición de las clases globales (como vistas). Normalmente las reglas de fusión de información están contenidas en esta definición.

La traducción Local as View requiere un proceso más complejo, porque no está claro a priori qué partes de una determinada consulta de usuario se definen a través de la vista. Cada vista global puede contribuir potencialmente a la planificación de la consulta. Este problema, conocido como "responder consultas usando solo vistas" es un problema NP-completo (Levy et al., 1995) para consultas conjuntivas y vistas conjuntivas, que puede ser resuelto enumerando el numero de combinaciones de vistas y probando consultas para cada una de estas combinaciones.

Los CSLs son necesarios porque las relaciones del esquema global no consiguen un match uno-uno con los esquemas globales. Existen dos razones por las que esto ocurre. En primer lugar, los esquemas locales tienen diferentes niveles de detalle entre ellos y el esquema global. La segunda razón es que si diferentes esquemas modelan la misma información, tienden a dividir los atributos en las relaciones de diferentes maneras. Los modelos LaV y GaV solo resuelven estos problemas en parte. En LaV el esquema global debe contener todos los atributos compartidos por las múltiples fuentes sean o no útiles para el resultado final. En el GaV el esquema global debe tener todas las relaciones presentes en las fuentes o consultas conjuntivas sobre ellos. El modelo GLaV combina el LaV y GaV, permitiendo definiciones flexibles de esquema independientemente de las fuentes. El GLaV permite descripciones de las fuentes que contengan consultas recursivas sobre las fuentes, para recorrer todas las posibles combinaciones de las fuentes.

2.4.3 Otros modelos de federación

Además de los modelos federados comunes, es posible que en algunos contextos sea útil la adopción de unos modelos que, aún siendo teóricamente

federados, se basan en arquitecturas tecnológicas distintas a las originales de los sistemas de bases de datos distribuidos clásicos.

2.4.3.1 Federación basada en Workflows

Los sistemas de *workflows* han sido muy utilizados en los últimos años para generar procesos complejos de procesado de datos, estos sistemas tienen una gran capacidad de reutilizar módulos y permiten que una persona sin una gran formación en informática pueda realizar flujos de procesado complejos de forma visual y automatizar tareas que anteriormente le llevaban semanas. Aunque es posible que haya cierta controversia al denominar a los sistemas de *workflows* sistemas federados, en este trabajo se incluyen en esta categoría los sistemas de *workflows* que, además de presentar las características básicas vistas en el capítulo III, presenten un modelo común, que actuaría como esquema común en un sistema federado normal y la posibilidad de incorporar características semánticas dentro del sistema.

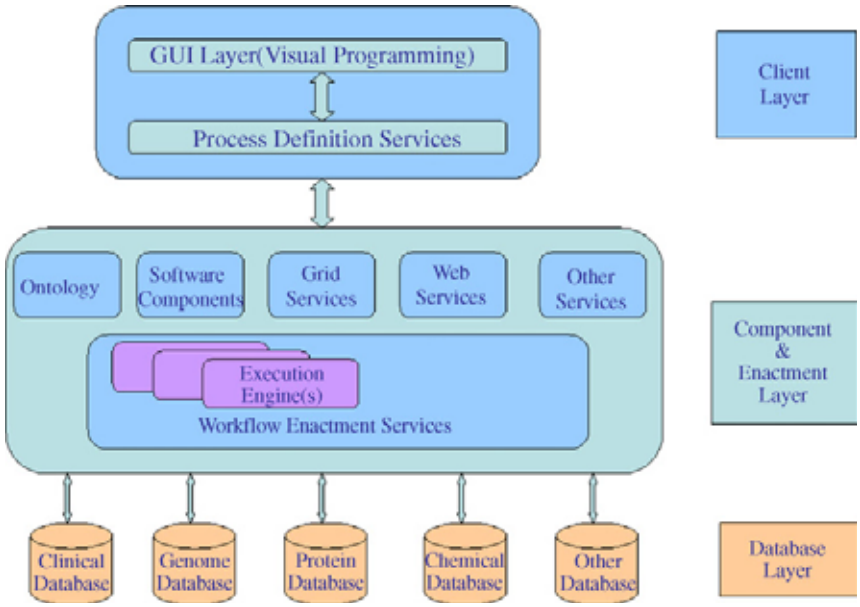


Figura 11: Arquitectura de integración con workflows de Towari et al. 2007

Se entenderá en este caso que un sistema de *workflows* federado (Figura 11) está formado por:

- Un conjunto de servicios asociados a una fuente de datos que obtienen los datos de esta y lo transforman al modelo común. Estos servicios, que actúan como *wrappers*, deben también anotar semánticamente dichos datos.
- Puede existir un servicio "mediador" que, a partir de las consultas que le envíe el cliente, subdivida dichas consultas en subconsultas con las que llama a los servicios *wrapper*. Posteriormente recoge la información obtenida sobre el modelo común y la presenta al cliente.
- En términos de usabilidad, la característica más interesante de este sistema es que el servicio mediador puede ser sustituido directamente por un usuario sin muchos conocimientos de informática que, de manera visual, cree el *workflow* seleccionando los datos que pretende recuperar de cada fuente e interrelacionándolos entre ellos. Obviamente esto tiene como requisito que exista una serie de servicios *wrappers*, unidos a servicios de tipo filtro que permitan filtrar los datos obtenidos a partir de características definidas por el usuario y un interfaz visual, como Taverna (Figura 12) (Hull et al., 2006), para que el usuario pueda crear el *workflow*, ejecutarlo y recuperar el resultado.

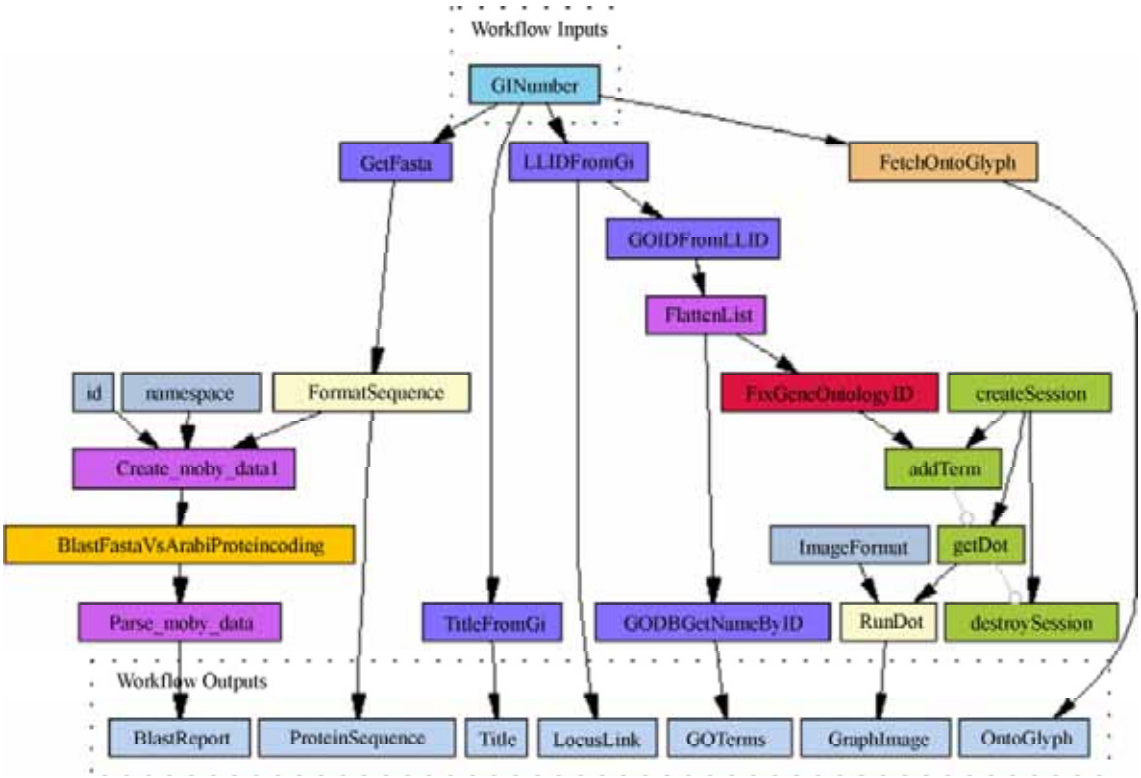


Figura 12: Ejemplo de workflow de Taverna

2.4.3.2 Federación basada en sistemas multi-agente

El modelo de integración de información basado en sistemas multi-agente extiende el modelo federado, añadiendo al sistema de integración las características propias de los sistemas de agentes.

El sistema se compone de los siguientes agentes:

- Agente usuario: Este agente es la vista del sistema. A partir de él los usuarios interactúan con el resto del sistema de integración. Su cometido es obtener las consultas del cliente y reenviarle la consulta transformada al sistema de integración por medio del "Agente mediador".
- Agente mediador: Este agente obtiene la petición del agente usuario, descompone la consulta y la envía a los agentes apropiados. Una vez obtiene las respuestas, las integra de manera correcta y las devuelve al agente usuario.
- Agente recurso: Este agente hace las funciones del *wrapper* en el sistema mediador clásico. Recibe la consulta del agente mediador y genera una consulta apropiada en el lenguaje concreto de la base de datos de la que está encargado. Una vez obtenidos los datos, los traslada al lenguaje intermedio y se los reenvía al agente mediador.
- Agente de ontología: Este agente contiene la o las ontologías que usa el sistema. Puede ser usado tanto como vocabulario común, para que todos los agentes puedan resolver potenciales heterogeneidades semánticas, como para obtener relaciones entre datos.

2.5 Integración de datos biomédicos

Existe una gran cantidad de trabajos que discuten los problemas de la integración biomédica (Stevens et al., 2000b, Stein, 2003, Cannata et al., 2005, Fisher and Henzinger, 2007, Brooksbank and Quackenbush, 2006, Philippi and Kohler, 2006, Philippi, 2008, Goble and Stevens, 2008, Antezana et al., 2009). En la era postgenómica, la investigación biomédica se ha convertido en un área basada en un estudio intensivo de grandes colecciones de datos. Por lo tanto, los investigadores en este dominio se encuentran con los problemas de tratar con grandes cantidades de datos heterogéneos y distribuidos. Este tipo de problemas se acentúan en las ciencias

de la vida, debido a una serie de factores. Estos factores se pueden dividir en dos grupos (Antezana et al., 2009). El primer grupo de factores, que podrían denominarse "naturales", reflejan las peculiaridades de los datos y el conocimiento biológico, en especial, como se ha visto antes, su complejidad. El otro grupo de factores podrían denominarse "culturales". La avalancha de datos procedentes de las técnicas de alto rendimiento ha cogido a los investigadores desprevenidos. Tradicionalmente, la investigación biomédica no se ha caracterizado por usar lenguajes formales o estándares. El resultado es que el dominio biomédico está altamente fragmentado. El número de bases de datos en biomedicina es exageradamente grande comparado con otras disciplinas que hacen uso de una gran cantidad de datos (e.j. física de partículas o astronomía) y las bases de datos suelen ser autónomas y estar desconectadas. La extrema heterogeneidad en los modelos de datos y de los formatos (Wang et al., 2005, Brazma et al., 2006, Cannata et al., 2005) hacen que la interoperabilidad y la integración de datos sea enormemente complicada (Quan, 2007). Finalmente, existen una gran cantidad de sinónimos, homónimos y polisemia en los datos biomédicos, que afectan negativamente tanto a la sensibilidad como a la especificidad de la recuperación de datos biomédicos (Blaschke et al., 2003). Como resultado de estos problemas, existe una diferencia entre la cantidad de datos obtenidos y el conocimiento extraído de ellos (Hodgson, 2001) (Szalay and Gray, 2006), incluso teniendo en cuenta que la cantidad de datos biológicos son menores que en otras disciplinas. Para solucionar esto, la integración de datos biomédicos se hace imprescindible para realizar consultas complejas, detalladas, sobre diversos recursos distribuidos, para facilitar el análisis de los datos, la generación de hipótesis y el diseño de experimentos.

En los últimos años ha aparecido la voluntad de estandarizar la información biológica, principalmente a partir de las sinergias surgidas entre las comunidades de investigadores biomédicos e informáticos, que reconocieron la necesidad de hablar el mismo lenguaje. Algunos de estos esfuerzos se basan en definir protocolos estandarizados para describir experimentos biológicos (Taylor et al., 2008), otros para desarrollar dialectos XML para intercambiar datos (Spellman et al., 2002). Estos esfuerzos ayudan a compartir, diseminar y reanalizar los datos generados.

Aparte de esto, el paso más importante ha sido la adopción de ontologías como medios para proporcionar conceptualización estandarizada del dominio en biomedicina. Las ontologías capturan entidades y relaciones en el dominio de la investigación biomédica (Bodenreider and Stevens, 2006). Actualmente, las ontologías más importantes en biomedicina está agrupadas bajo el paraguas de la fundación OBO (Open Biological Ontologies) (Smith et al., 2007b). La fundación OBO es un proyecto colaborativo que se dedica no sólo a la recolección de ontologías biomédicas, sino también a proporcionar un conjunto de principios fundacionales para estructurar futuros desarrollos de nuevas ontologías (ortogonalidad de ontologías, no solapamiento, etc.).

Inicialmente, el desarrollo de ontologías biológicas fue realizado por expertos de ese dominio, pero con pocos conocimientos de desarrollo formal de ontologías. Sin embargo, algunos desarrollos sí se llevaron a cabo por expertos multidisciplinares con conocimiento de biología, informática y filosofía (p.e. Gene Ontology).

Las ontologías están penetrando el campo de la integración de datos en la investigación biomédica. La integración de todos los datos biológicos depende de una recuperación y gestión eficiente de la información para hacer frente a cada vez más recursos distribuidos. Una de las soluciones más prometedoras es el uso de tecnologías de Web Semántica (Ruttenberg et al., 2007).

Las tecnologías de la Web Semántica (Berners-Lee et al., 2001) fueron diseñadas para afrontar el reto de la gestión de información de un mundo con recursos distribuidos. La web semántica promete una infraestructura que cuenta con contenido entendible para máquinas y, por lo tanto, una Word Wide Web hecha con contenidos relacionados semánticamente, en lugar de una mera colección de documentos HTML. Los sistemas informáticos basados en integración semántica de datos y relaciones ontológicas proporcionarán el marco necesario para interrogar y recuperar información correcta.

3 Estado de la cuestión

3.1 Introducción

El presente capítulo está dedicado a describir las diferentes aproximaciones a la integración de datos en el ámbito biomédico existentes en la bibliografía. Se describen distribuidas por la tecnología que utilizan. Se pueden encontrar más aplicaciones de integración de datos en biomedicina en estos trabajos (Williams, 1997) (Sujansky, 2001) (Wong, 2002) (Stein, 2003) (Köhler, 2004) (Goble et al., 2006, Philippi and Kohler, 2006) (Brazhnik and Jones, 2007) (Louie et al., 2007) (Ruttenberg et al., 2007) (Burgun and Bodenreider, 2008) (Cheung et al., 2008) (Goble and Stevens, 2008, Antezana et al., 2009) (Cheung et al., 2009) (Akula et al., 2009) (Zhang et al., 2009) (Ruttenberg et al., 2009) (Lambrix et al., 2009).

3.2 Sistema de navegación por links

Los sistemas de navegación por links están basados en enlaces entre fuentes de datos, con el objetivo de poder navegar desde un tipo de dato hasta otro. Este tipo de sistemas permiten acceder de manera muy simple a fuentes de datos diferentes.

Uno de los sistemas más importantes de navegación por enlaces es el EBI-SRS (Sequence Retrieval System), en la Figura 13 (Etzold et al., 1996, Zdobnov et al., 2002). Este sistema permite la conexión de cientos de bases de datos y permite la navegación a través de ellas. El SRS fue desarrollado inicialmente para facilitar el acceso a bases de datos de secuencias biológicas que, en un principio, se almacenaban como ficheros de texto. Posteriormente se incluyó un modelo orientado a objetos con el objetivo de poder representar clases biológicas más complejas. Una de las principales ventajas de este sistema de integración es la rapidez. El SRS recupera datos directamente de ficheros de texto plano sin pasar por los procesos de integración de los sistemas gestores de bases de datos. El problema del SRS es que no permite añadir nuevas bases de datos que no hayan sido descritas previamente como parte del sistema.

Figura 13: Portal SRS del EBI

Entrez (Benson et al., 1990) es otra importante y popular herramienta en investigación biomédica, que proporciona acceso a más de 40 bases de datos. Este sistema de navegación por links se desarrolló en el National Center for Biotechnology Information (NCBI), y consiste en establecer enlaces entre las entradas individuales de sus bases de datos, como Pubmed y otras populares bases de datos de biomedicina.

El BioMolquest (Bukhman and Skolnick, 2001) es un motor de búsqueda relacionado con bases de datos relacionadas con proteínas como PDB, Swiss-PROT, ENZYME o CATH. Se basa en almacenar bases de datos completas en un repositorio central y proporcionar enlaces cruzados para los datos de estas bases de datos. Esta aproximación no puede ser considerada como un warehouse, debido a que almacena las bases de datos con su formato nativo y únicamente proporciona un front-end de acceso y un sistema de navegación por enlaces a través de ellas.

Otro enfoque distinto de sistema de navegación por links es el portal Diseasecard (Oliveira et al., 2004, Dias et al., 2006). Este portal proporciona un punto de acceso a información médica y genética de enfermedades raras. Este sistema integra diversas bases de datos, relacionando conceptos y construyendo “tarjetas” (*cards*) para cada enfermedad (Figura 14).

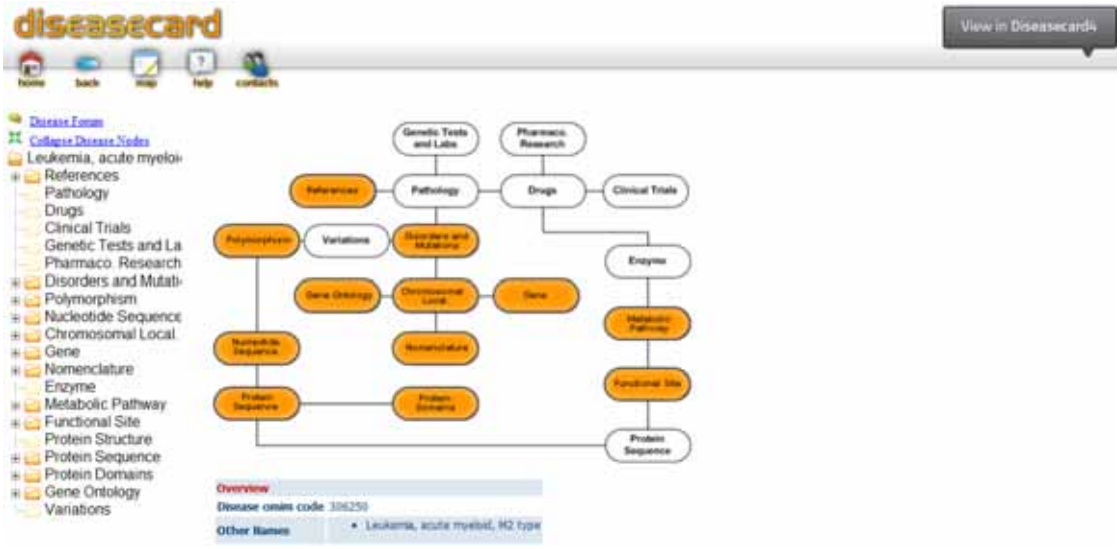


Figura 14: Ejemplo del card "Leukemia" en DiseaseCard

A partir de estas tarjetas se accede a la información más amplia sobre la enfermedad a través de las fuentes de datos originales (Figura 15).

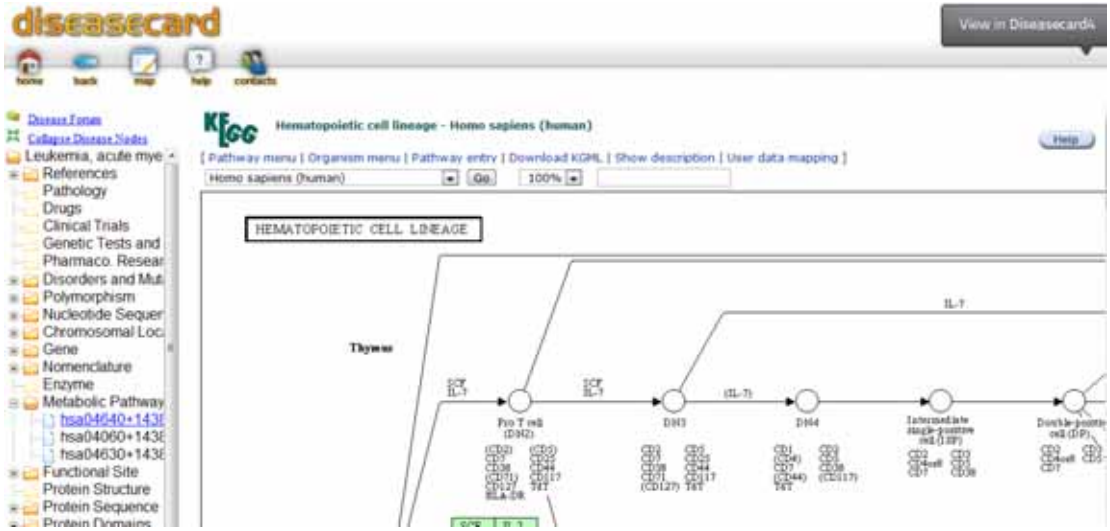


Figura 15: Información KEGG sobre el card "Leukemia"

BioDBNet (Mudunuri et al., 2009) es un portal web que permite la integración de más de 100 bases de datos diferentes usando referencias cruzadas para desarrollar el proceso de integración. Los autores consideran este sistema como un warehouse, pero la ausencia de un esquema global identifica este sistema más como un sistema de navegación por links. Una de sus características más útiles es la posibilidad de convertir identificadores de una entidad en una base de datos en un identificador de otra base de datos.

3.3 Warehouse

La arquitectura *warehouse* ha sido muy utilizada para resolver problemas de integración de datos en biomedicina. Uno de los pioneros en el campo de los sistemas de integración de información genómica es el IGD (Integrated Genomic Database) (Ritter et al., 1994). Este innovador sistema utilizaba data *warehouses* locales para almacenar los datos de diversas bases de datos genómicas en un esquema especifico.

Otro original sistema de *data warehouse* es la aproximación de Nadkarni et al. (Nadkarni et al., 1998) del Center of Medical Informatics de Yale, que consiste en la integración y almacenamiento homogéneo de estudios clínicos obtenidos en formatos heterogéneos. Esta aproximación consiste en almacenar los datos de estudios clínicos en formato entidad-atributo-valor. Esto permite consultar los sistemas obteniendo los resultados en diversos estándares de informes o en texto plano.

La aproximación de *data warehouse* del DataFoundry (Critchlow et al., 2000) introduce el concepto de mediador (Figura 16) para crear, poblar y mantener el *warehouse*.

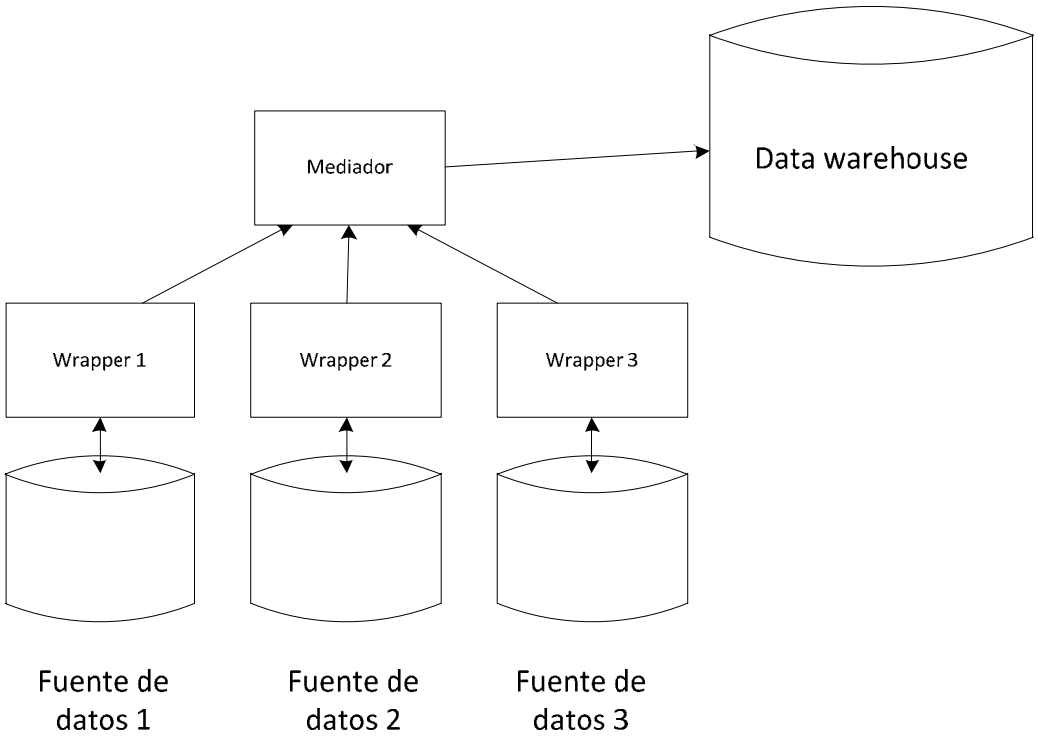


Figura 16: Arquitectura warehouse con mediador del DataFoundry

El proceso de integración se realiza conectando los diversos datos relacionados de las distintas bases de datos, al contrario de lo que sucede en otras aproximaciones que integran todos los datos en una vista simple. Esta aproximación permite obtener vistas consistentes de las instancias de los datos mientras que mantienen la heterogeneidad entre las instancias. El sistema trabaja con varias bases de datos como pueden ser PDB, SWissProt, SCoP o dbEST.

El Genome Information Management System (GIMS) (Cornell et al., 2001) integra datos de genoma, transcriptoma, mutaciones, fenotipo e interacciones de proteínas en un entorno de *warehouse*. Este sistema almacena los datos provenientes de las bases de datos en una base de datos objetual y utiliza ese mismo esquema como modelo global. El objetivo principal de este sistema es realizar tareas de análisis de los datos sobre las secuencias.

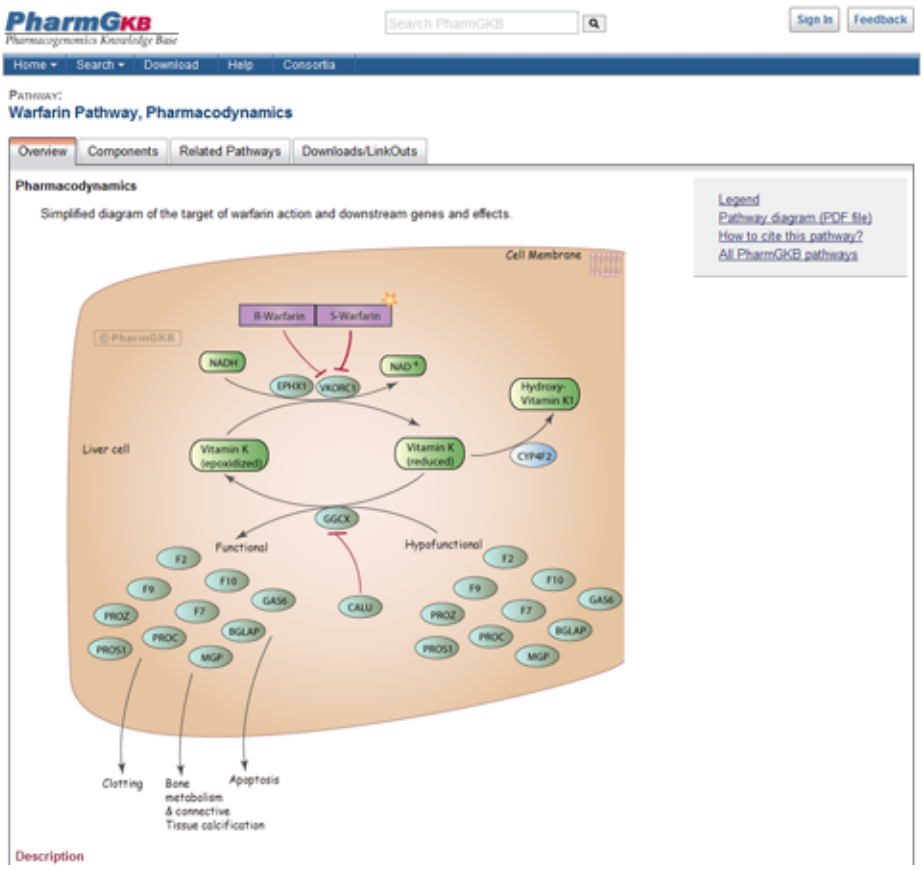


Figura 17: Análisis del pathway de la Warfarina en PharmGKB

El proyecto PharmGKB (Klein et al., 2001, Gong et al., 2008, Thorn et al., 2010, McDonagh et al., 2011) comenzó como un repositorio de datos que relacionaba genes

y fármacos, pero pronto se convirtió en un sistema de integración genérico que comprende la integración de una gran cantidad de fuentes de datos, principalmente en farmacogenética (Figura 17). PharmGKB está basado en un warehouse, que incluye datos de ciertas entidades biológicas principales, como Genes, Fármacos, Enfermedades, Literatura, Rutas, Variantes, Estructuras y otras, que son anotadas con datos provenientes de bases de datos externas.

El Ensembl (Hubbard et al., 2002) (Flicek et al., 2011) nació como una base de datos de secuencias desarrollado por el EBI, pero pronto fue integrando datos provenientes de otras bases de datos. Una vez que los genes son incluidos en el sistema, se anotan funcionalmente con información proveniente de InterPro, OMIM, SAGE expression, etc. Debido a que todas las bases de datos se almacenan dentro del propio sistema y con un esquema propio, el Ensembl puede ser considerado como un *data warehouse*.

Un caso muy parecido al anterior es el UCSC Genome Browser (Karolchik et al., 2003), que partiendo de una base de datos de secuencias, ha ido incorporando múltiples bases de datos hasta convertirse en un visor de múltiples tipos datos procedentes de múltiples bases de datos, centrándose todo en la secuencia. Además, este sistema permite el acceso a bases de datos de Ensembl.

En 2002 la Comisión Europea financió el proyecto TEMBLOR, coordinado con el EBI, con el objetivo de proporcionar recursos para datos de interacciones proteína-proteína, estructurales y de microarrays. Este consorcio desarrolló una capa de integración, el Integr8 (Kersey et al., 2003, Kersey et al., 2005). Esta aproximación usa *data warehouse*, para integrar datos procedentes de diversos tipos de entidades biológicas (genes, proteínas, cromosomas, etc.). Posteriormente, a través de su portal web permite realizar análisis de los datos.

EnsMart (Kasprzyk et al., 2004) está basado en una aproximación de *warehouse*, recopilando y transformándolos datos de fuentes de datos originales y almacenándolos en bases de datos locales denominadas *data marts*. Este sistema tiene dos tipos de *front-end* para proporcionar la interacción, uno basada en web y otro *standalone*, que además proporciona un API para interactuar con otras aplicaciones.

Este sistema fue originalmente desarrollado para interactuar con las bases de datos de Ensembl principalmente de anotación genómica, anotación funcional y expresión, aunque actualmente puede incorporar otro tipo de fuentes de datos externas.

La aproximación seguida por LIMBO (Philippi, 2004) está basada en data *warehouse*, almacenando datos de bases de datos heterogéneas y distribuidas en un repositorio centralizado con una estructura común. La característica más destacable de este sistema es que el esquema del *warehouse* está implementado sobre una estructura generalizada de sólo tres tablas, DATA, RELATION y METADATA. La ventaja de usar este tipo de estructura es que no es necesario desarrollar un modelo determinado para poder proporcionar acceso a datos complejos y que los cambios en la estructura de las fuentes no afectan demasiado a la estructura de datos.

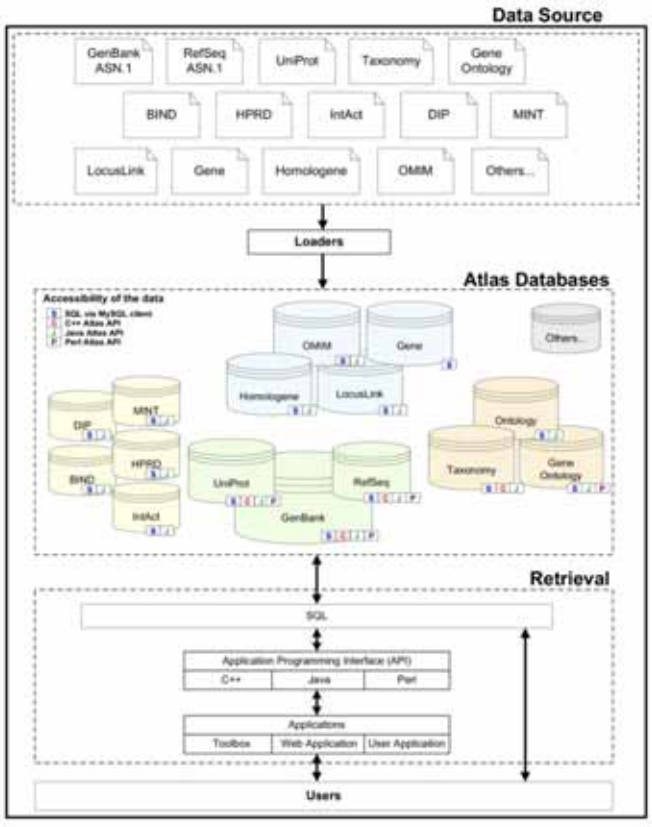


Figura 18: Arquitectura del Atlas

ATLAS (Shah et al., 2005) es un *data warehouse* extensible que permite usar modelos relacionales comunes como modelos de integración (Figura 18). Cada uno de estos modelos de datos comunes almacena información sobre una entidad (p.e. interacciones moleculares, genes, secuencias, etc.) que puede estar relacionada con

otras entidades usando el modelo relacional. Los autores desarrollaron un API para poblar el *warehouse* con datos de nuevas bases de datos. Este sistema puede consultarse usando APIs escritos en diferentes lenguajes como Java, Perl o C++.

El sistema *warehouse* COLUMBA (Trissl et al., 2005) permite la integración de datos relacionados con proteínas. En sus orígenes Columba integraba 12 bases de datos estructurales, entre las que incluye PDB, KEGG, Swiss-PROT, SCOP, Gene Ontology y ENZYME. Una de las características diferenciadoras del COLUMBA es que nunca mezcla datos de diferentes fuentes en una misma tabla. En lugar de eso utilizan una aproximación multidimensional, en la que cada fuente de datos se considera como una dimensión.

Biowarehouse (Lee et al., 2006) es un toolkit de código abierto que permite la construcción de *warehouses* usando sistemas gestores de bases de datos comunes. El esquema de este sistema se diseñó estudiando el esquema de cada una de las bases de datos que lo forman, y está basado en un modelo relacional. Este nuevo *warehouse* debe de ser compatible con los esquemas de las fuentes de datos originales porque es rígido.

Biozon (Birkland and Yona, 2006) es un portal web que sirve como interfaz de un *data warehouse* desarrollado en un principio para biología estructural, pero que también incluye información genómica. Está basado en las relaciones entre objetos biológicos fundamentales, donde cada base de datos de origen está mapeada en un esquema jerárquico altamente integrado. Biozon incorpora herramientas de análisis como pueden ser el Blast, análisis de EST, etc.

El modelo hibrido del Cancer Research Databases (Bichutskiy et al., 2007) está combina un modelo un *warehouse* con una aproximación federada a través de un diseño con un mediador. En lugar de usar los dos enfoques secuencialmente, esta arquitectura los utiliza en paralelo, dependiendo de los tipos de datos o de las fuentes. Por un lado, el enfoque mediador se usa para tratar con los tipos de datos de gran tamaño, que cambian a menudo y con fuentes de datos de alta disponibilidad y tiempos de procesado de dato flexibles, mientras que el *warehouse* se usa para tipos de datos más pequeños, que cambian periódicamente, fuentes que pueden estar off-

line con cierta frecuencia o con unos requisitos de tiempo de ejecución importantes. El diseño del esquema se basa en dos entidades principales, DockingResult y AssayResults, que están relacionadas con las tablas Experiments, Mutants y Molecules.

A continuación se presentan dos modelos de integración de datos de hongos a partir de *warehouse*. En primer lugar E-Fungi (Hedeler et al., 2007) integra más de 30 genomas de hongos, además de otras bases de datos, como puede ser Pfam, KEGG, Reactome o Metacyc. El modelo de datos se ha implementado utilizando Java Data Objects, que permite la persistencia de estos datos en el *warehouse*.

En la línea del anterior E-Fungi, el Comparative Fungal Genomic Platform (Park et al., 2008) es otra plataforma de comparación de datos de hongos. En este caso, integra 65 especies, e integra herramientas de análisis genómico comparativo como el Blast, ClustalW, InterProScan, SignalP, PSORT, etc. También incorpora 4 especies de Oomicota y 27 organismos que no son hongos, así como bases de datos externas como InterPro y GO.

BioDWH (Topel et al., 2008) es otro *toolkit* para el desarrollo de *data warehouses* que contengan datos biomédicos. Una de las características más importantes de este sistema es que mantiene los datos siempre actualizados, monitorizando las fuentes de datos originales y actualizando los datos del *warehouse* cuando sea necesario, siguiendo algunas estrategias de actualización. El sistema tiene una serie de *wrappers* para las fuentes de datos más comunes, como UniProt, KEGG, OMIM, GO, Enzyme, BRENDA, PDB, MINT, SCOP, EMBL-Bank o PubChem. Este sistema utiliza un modelo de datos objetual, que persiste el modelo relacional por medio del mapeador objeto-relacional Hibernate.

Anwar y Hunt (Anwar and Hunt, 2009) describen un sistema de integración semántico aplicado a la bacteria Francisella Tularensis Novicida. En primer lugar transforman las fuentes de datos a formato RDF, que posteriormente almacenan en un repositorio común, en este caso el Sesame.

El GeNS (Arrais et al., 2009) es un sistema *warehouse* que permite la integración de los recursos biomédicos más utilizados, como EMBL, Uniprot (SwissProt

y TrEMBL), ExPASy (PROSITE y ENZYME), NCBI (Entrez, Taxonomy, Pubmed, GenBank y OMIM), Biomart, ArrayExpress, InterPro, GO, KEGG y PharmGKB. El esquema del GeNS está desarrollado en dos niveles. En un nivel general prioriza la flexibilidad, escalabilidad y la integración de diversos tipos de datos. Por otro lado, un esquema especializado integra únicamente un pequeño número de conjunto de datos. Este esquema permite la integración, a partir de un modelo genérico, de una gran cantidad de datos heterogéneos, pero especializa esta integración en una capa superior con un meta-modelo concreto.

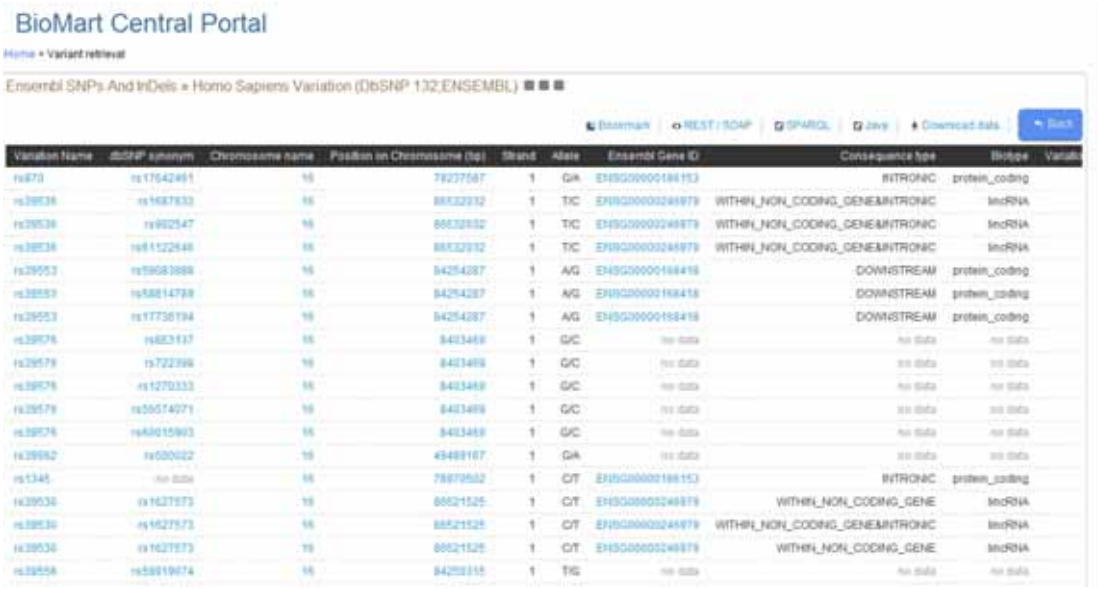


Figura 19: Portal BioMart

MartView es un portal web (Figura 19) que permite el acceso a diversas fuentes heterogéneas a partir del *warehouse* BioMart (Smedley et al., 2009). Este sistema implementa una capa de servicio (MartService) que proporciona acceso, mediante programación, a todos los recursos de Biomart a través de servicios web. La flexibilidad de la arquitectura desarrollada en Biomart, convierte esta arquitectura en una de las soluciones más utilizadas, implementando también interfaces para que pueda usar paquetes externos de acceso o procesado de datos como Bioconductor, DAS, Galaxy, Cytoscape o Taverna.

3.4 Sistemas federados

Una de las primeras propuestas para integración de datos en sistemas federados fue The Stanford-IBM Manager of Multiple Information Sources (TSIMMIS)

(Chawathe et al., 1994) (Garcia-Molina et al., 1997) y el SIMS (Arens et al., 1998), que no estaban orientados a datos biomédicos, sino que eran sistemas generales. Otras aproximaciones genéricas, que utilizan sistemas multi-agente son la de Bayardo et al. (R. J. Bayardo et al., 1997) y el KRAFT (Preece et al., 2000). A continuación se detallan las aproximaciones de integración con sistemas federados orientadas a sistemas biomédicos.

Biokleisli (Davidson et al., 1997) (Kolatkar et al., 1998) (Chung and Wong, 1999) es uno de los primeros intentos de desarrollar una aproximación federada de integración de datos en biomedicina. Consiste en un *framework* que proporciona acceso de lectura a diversos conjuntos de datos en una estructura de datos compleja. Está basado en el Kleisli (Wong, 1995), un lenguaje de procesado de colecciones, que permite exportar los datos de cada fuente de datos en tiempo de consulta a un formato de texto llamado CPL y presentarlo de manera homogénea. Chung et al. (Chung and Wong, 1999) desarrollaron una solución similar usando también Kleisli.

Desde el comienzo de la federación de datos biomédicos se ha visto la posibilidad de la aplicación de sistemas multiagente. En el trabajo de Imai et al. (Imai et al., 1997), se utiliza una aproximación multi-agente para implementar un modelo federado, donde hay un agente *wrapper* para cada fuente de datos y un agente-usuario para la presentación de los resultados, que actúa como mediador. Los autores comparan su sistema con el TSIMMIS, ya que es el primer desarrollo que integra datos biomédicos utilizando el patrón *wrapper*/mediador.

El primer modelo federado en incorporar información semántica fue el TAMBIS (Transparent Access to Multiple Bioinformatics Information Sources) (Baker et al., 1998) (Stevens et al., 2000a), que proporciona filtrado y recuperación transparente de información biológica usando un mediador y varios *wrappers* para crear la ilusión de una sola fuente de datos. El mediador usa una ontología como modelo conceptual, que en sus inicios se implemento en GAIL, pero que actualmente está implementada en DAML + OIL. Este sistema fue el primero en permitir realizar consultas sobre el modelo conceptual utilizando ontologías para posteriormente traducir la consulta al original.

La experiencia de fabricantes comerciales en sistemas federados como IBM se aplicó a datos biomédicos en el software de acceso a datos Discoverylink (Haas et al., 2001). Este sistema está basado en Garlic, que es un prototipo de un sistema federado de IBM y DataJoiner, un sistema gestor de bases de datos federados de IBM. El primero usa *wrappers* para permitir una arquitectura modular y facilitar la integración de nuevas fuentes de datos, mientras que el segundo es una tecnología comercial para la federación de bases de datos relacionales y optimización de consultas.

El Target Informatics Net (Eckman et al., 2001) es un sistema de integración que permite la integración de bases de datos heterogéneas como la Mouse Genome Database, Gene Expression Database, Gen Bank, SwissProt, Pubmed, GeneCards, los resultados de software como el Blast o Prosites u otras bases de datos locales. Estas fuentes de datos pueden ser tanto bases de datos, ficheros planos, sitios web o aplicaciones. Este sistema está basado en una aproximación hibrida que combina el modelo federado con las copias locales de la aproximación de *warehouse*. Todas las fuentes de datos se copian y actualizan regularmente en un repositorio centralizado. Posteriormente, se usa la federación para acceder a estas copias locales. La comunicación entre las bases de datos locales y el procesador de consultas se realiza usando la arquitectura de objetos remotos CORBA.

El modelo presentado por Mork (Mork et al., 2001) está basado en el esquema mediador, representado como un grafo, donde cada nodo representa una entidad y las aristas representan las conexiones entre las entidades. Está basado en seis entidades principales, fenotipo, gen, locus, nucleótido, proteína y estructura. Este esquema mediador es muy flexible y puede ser fácilmente extendido tanto en nuevas entidades como en nuevas fuentes de datos. El modelo de datos se implementó describiendo cada entidad con un DTD y formulando las consultas utilizando XML para obtener los datos de las fuentes de datos originales a través del motor de consultas Tukwila.

La plataforma ISYS (Siepel et al., 2001) está basada en una aproximación con mediador usando componentes. En lugar de definir un esquema muy detallado, con el objetivo de maximizar la flexibilidad del sistema, el modelo general se desarrolló

utilizando interfaces Java para definir términos abstractos y generales en las entidades esenciales. Cada componente puede extender o implementar estas entidades.

El modelo de federación utilizando mediación propuesto por Ludasher (Ludascher et al., 2001) es otra aproximación semántica para resolver problemas de integración de datos. En este modelo, las vistas se definen y se ejecutan a nivel de modelos conceptuales, en lugar de modelos estructurales. Este sistema usa mapas de dominio, redes semánticas de conceptos y relaciones para mediar entre las fuentes. Los datos se traducen a XML y posteriormente se integran in en mediador a través del modelo conceptual. Para realizar el mapeo de los conceptos y la definición de las consultas se utiliza también XML.

En el mismo sentido que Imai, Karasavvas (Karasavvas et al., 2002) (Karasavvas et al., 2004) propone un sistema multi-agente de integración de fuentes bioinformáticas. En contraste con la propuesta de Imai, el sistema multi-agente de Karasavvas usa agentes no sólo para implementar la estructura de *wrapper*/mediator, sino también para implementar el proceso de toma de decisiones y de construcción y ejecución de la consulta mediante los agentes de decisión.

Después del desarrollo de DiscoveryLink, IBM propuso el Information Integrator. Este paquete software comercial de IBM fue desarrollado como una extensión de la base de datos DB2 de IBM, para proporcionar acceso federado a diferentes fuentes de datos. En la publicación de Arenson (Arenson, 2003), se describen los beneficios de la aplicación del Information Integrator en un contexto de integración de datos biomédicos. La ventaja principal del Information Integration es la inclusión de un módulo llamado Life Sciences Data Connect, que consiste en una serie de *wrappers* para acceder a fuentes de datos biomédicas no relacionales.

El uso de sistemas de gestión de bases de datos relacionales en el proceso de integración de información hace que surjan una serie de problemas relacionados con la semántica. En el SEMEDA (Semantic Meta Database) (Köhler et al., 2003), se desarrolló una arquitectura de 3 capas. Consiste en una base de datos relacional donde se almacenan ontologías, metadatos de la base de datos y definiciones semánticas. Utilizan JDBC para acceder a las fuentes de datos biomédicas en el caso de que tengan

un interfaz relacional, en otro caso utilizan BioDataServer. En el proceso de integración SEMEDA utiliza dos tipos de ontologías; una general como por ejemplo GO, para unificar los registros de las bases de datos y una propia SEMEDA Custom Ontology con la que define las bases de datos a nivel esquema y que actúa como modelo de datos.

El sistema Biomediator (Donelson et al., 2004) es una aproximación federada que utiliza un grafo para definir el modelo, donde los nodos representan las instancias y las aristas representan las relaciones que conectan las entidades en una o más de las fuentes de datos del sistema. El sistema utiliza un esquema mediador anotado para describir las entidades biomédicas. Cada usuario puede crear un esquema mediador propio donde describa sus necesidades de integración. El usuario anota el esquema añadiendo información sobre fuentes de datos y sus relaciones. Los *wrappers* de las fuentes de datos en Biomediator están generalizados, exponiendo todos los datos y sin mapearlos a un esquema particular.

El esquema federado que presentan Robinson y Rahayu (Robinson and Rahayu, 2004) está basado en mediación y tiene como peculiaridad que usa el Bioinformatic Sequence Markup Language (BSML) para intercambiar datos entre las diferentes capas del sistema.

Basándose en la arquitectura propuesta en TAMBIS, y añadiendo algunos nuevos, como puede ser la definición de ontologías partiendo de las bases de datos de origen, el Query Integrator System (Marenco et al., 2004, Lam et al., 2007, Marenco et al., 2009) es una aproximación de mediación semántica federada. Las consultas se describen en el mediador utilizando un lenguaje común y se traducen a la sintaxis específica de la fuente de datos original. El sistema se divide en tres servidores: El Integrator Server (IS), el DataSource Server (DSS) y el Ontology Server (OS). El DSS captura el esquema de la fuente de datos, creando un repositorio con los metadatos del esquema y los mapea en una ontología creada ad-hoc para esa fuente. El IS crea una ontología específica relacionada con el dominio del problema. Dicha ontología se mapea con las ontologías específicas de las fuentes de datos del DSS, que representan los metadatos de estas. Finalmente el OS mantiene un esquema de integración y uno o más vocabularios controlados que se usan para la federación.

Uno de los desarrollos basados en agentes más citados es Ontofusion (Alonso-Calvo et al., 2007), que consiste en una aproximación a la integración de bases de datos utilizando mediación usando arquitectura de sistemas multi-agente y ontologías. La arquitectura del sistema es la más utilizada en problemas de integración con MAS, que consiste en un agente actuando como mediador y un agente *wrapper* por cada fuente de datos. Para la implementación utiliza la plataforma de sistemas multi-agente JADE. Ontofusion usa una traducción de consultas híbrida, en la que cada fuente de datos presenta su propio esquema individual (esquema virtual). Estos esquemas virtuales se generan por medio de un proceso de *mapping*, en el que un usuario asigna cada elemento del origen de datos a los conceptos de la ontología de dominio, pudiendo usar el sistema varias ontologías (Figura 20). Todos los esquemas virtuales se unifican en un esquema global automáticamente.

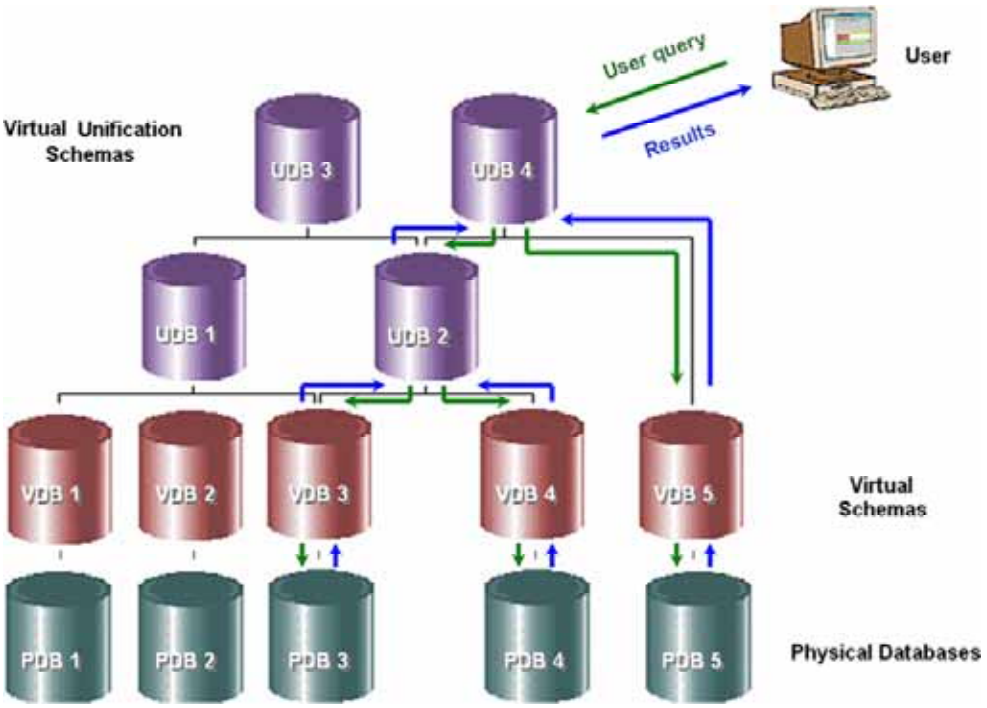


Figura 20: Arquitectura usando esquemas virtuales del Ontofusion

El Semantic Web-Enabled Data Integration (SWEDI) (Post et al., 2007) es una aproximación de integración utilizando modelos ontológicos, en particular OWL, para enlazar pequeños modelos de fuentes de datos con grandes modelos semánticos, usando una aproximación *bottom-up*. En primer lugar, todas las fuentes de datos se transforman a RDF, posteriormente se enlazan con los modelos de conocimiento.

Después se escogen los dominios de conocimiento comunes y finalmente el sistema construye y ejecuta la consulta semántica.

Linkhub (Smith et al., 2007a) es una arquitectura semántica para integrar datos usando un modelo de integración federado jerárquico de "hub de hubs" (ver Figura 21). Las fuentes de datos se conectan utilizando pequeños hubs locales, que cubren entidades similares. Estos hubs locales se conectan además en un hub más grande siguiendo una arquitectura federada jerárquica. Se puede acceder a las fuentes de datos tanto utilizando SQL como RBF. El sistema puede construir consultas complejas utilizando un grafo semántico RDF de identificadores biomédicos y sus relaciones. Este sistema se ha probado usando Uniprot y el North East Structural Genomics Consortium.

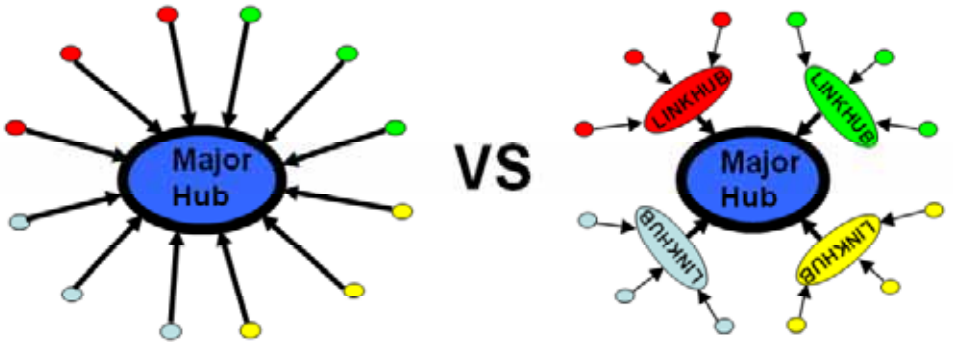


Figura 21: Modelo de hubs jerárquicos del Linkhub

DebugIT (Lovis et al., 2008) es un proyecto del 7º Programa Marco de la UE cuya meta es mejorar la detección y eliminación de bacterias usando las tecnologías de la información. Uno de los objetivos del proyecto es integrar fuentes heterogéneas, tanto clínicas como biológicas, usando una infraestructura semántica. Se utiliza SPARQL para acceder a las fuentes de datos y los resultados se presentan en RDF. El sistema utiliza *wrappers* para extraer los datos de las fuentes de datos locales, transformarlos al modelo global y almacenarlos localmente. El núcleo del sistema es una base de datos federada que enlaza todos los datos normalizados almacenados localmente y permite acceder a ellos utilizando un único punto de entrada.

La arquitectura del Prostate Cancer Information System (Min et al., 2009) es una federación con dos bases de datos principales, la base de datos de cáncer de próstata y la base de datos del registro de tumores. Un servidor de mapeo, que actúa

como mediador, almacena los *mappings* entre las ontologías y las bases de datos, que han sido descritos previamente en el lenguaje D2RQ. Finalmente, el último módulo de la arquitectura es un sistema de consulta de datos, que consiste en un interfaz en SPARQL que permite a los usuarios enviar consultas a través de un interfaz web.

El framework mediador basado en ontologías Khaos (Roldan-Garcia Mdel et al., 2009) es una arquitectura de integración federada que usa un directorio semántico para registrar y gestionar las ontologías. Este directorio semántico se usa para construir un modelo común.

BioXM es un modelo de integración semántica orientada a objetos para construir sistemas de gestión de conocimiento. Usa redes semánticas complejas para enlazar los conceptos "elemento", "relación", "anotación" y "contexto" para construir el modelo o usar modelos creados previamente como ontologías. Estas redes de relaciones semánticas complejas permiten detectar conexiones, extraer patrones y responder consultas complejas. Para optimizar el rendimiento, los datos se pueden gestionar usando alternativamente un sistema gestor de bases de datos relacional o integrando fuentes de datos externas, combinando las ventajas del modelo federado y del modelo *warehouse*.

3.5 Discusión

A lo largo de este capítulo se han visto una serie de soluciones que, desde mediados de los 90, han servido para realizar la integración de datos biomédicos en distintos ámbitos.

Muchas de las aproximaciones analizadas están muy cerca de resolver el problema de la integración de datos (Alonso-Calvo et al., 2007), o resuelven correctamente alguna de las etapas que comprenden el proceso de integración (Marenco et al., 2009), sin embargo, ninguna de las soluciones analizadas resuelven completamente el problema de la integración de datos biomédicos en un contexto genérico.

Las aproximaciones basadas en enlaces, a pesar de ser muy usadas en la comunidad científica biomédica, no proporcionan suficiente funcionalidad para

suponer una solución efectiva, no pueden tratar con consultas complejas y no trabajan bien con grandes cantidades de datos de naturaleza cambiante.

En el caso de las aproximaciones federadas y *warehouse*, se puede observar que existe cierta tendencia al cambio de un modelo centralizado *warehouse* a modelos federados semánticos (Post et al., 2007) (Smith et al., 2007a) (Maier et al., 2011), consecuencia en parte de las mejoras en las comunicaciones, en los sistemas gestores de bases de datos y en la incorporación de las herramientas semánticas como ontologías y el desarrollo de lenguajes de representación semántica como OWL o RDF. Otra posible evolución que se aprecia es el uso de modelos híbridos federados *warehouse* (Bichutskiy et al., 2007) (Arrais et al., 2009), que incorporan las ventajas de ambos.

En muchas ocasiones los sistemas desarrollados no han escogido la opción más ventajosa a nivel de arquitectura o a nivel de integración, no han seleccionado las fuentes adecuadas o han diseñado sistemas en los que añadir una nueva fuente, una vez diseñado el sistema, dispara los costes de desarrollo. Un ejemplo de estos errores es el ATLAS (Shah et al., 2005), en el que todos los datos del sistema se anotan con su fuente de datos de origen con el objetivo de no perder información, pero lo que consigue es que aparezcan inconsistencias semánticas y datos replicados. Un ejemplo de error al escoger el modelo con vistas a una posible ampliación es el GIMS o el ISYS, en los que la inclusión de una nueva fuente de datos implicaría un rediseño total, no solo del modelo de datos, sino del sistema completo, debido a la alta acoplación entre el modelo de datos y las fuentes de datos.

Para afrontar correctamente una tarea de integración de datos biomédicos, es necesario entender la complejidad de los datos biomédicos y analizar todas las posibles soluciones, evaluar cada una de las características del proceso de integración y de las fuentes de datos que deben ser integradas. Para esto se requiere el desarrollo de algún tipo de metodología de integración de datos biomédicos.

Sin embargo, la mayoría de los trabajos describen sistemas concretos que sólo pueden aplicarse a un escenario concreto. En algunos trabajos, como las revisiones de Sujansky (Sujansky, 2001), Philippi (Philippi and Kohler, 2006), Brazhnik (Brazhnik and

Jones, 2007) o Goble (Goble and Stevens, 2008) se analizan factores que son muy importantes de cara a la integración, que podrían considerarse una parte de una metodología. Otros trabajos proponen modelos que se adaptan a ciertas soluciones, como el caso de Zapletal (Zapletal et al., 2010) para warehouse, Hwang (Hwang et al., 2005) para agregación de datos o Reddy (Reddy et al., 1994) para la integración de esquemas. Sin embargo, no se ha encontrado en la bibliografía la descripción de ninguna metodología que intente abarcar todas las fases del proceso de integración de datos.

4 Hipótesis

Como se ha comentado en los capítulos I y II, existen una serie de problemas que atañen a la integración de datos biomédicos.

En la era post-genómica cada vez es mayor la generación de nuevos datos biomédicos. La gestión de estos datos implica su almacenamiento en bases de datos distribuidas de laboratorios y organizaciones. Esta gran cantidad de datos no se refiere únicamente al número, sino también a los distintos tipos de datos generados.

Esta gran cantidad de datos, la distribución de ellos en distintas fuentes y la complejidad inherente de los datos biomédicos hace muy complejo poder disponer de una visión conjunta del problema biomédico.

En los últimos años, la medicina genómica comenzando a ser utilizada en la práctica clínica. Los avances en la aplicación de las nuevas tecnologías ómicas a tareas, tanto de diagnóstico, como para la generación de fármacos dirigidos a un tipo de población concreta, indican que en no demasiado tiempo se podrá tener disponible realmente una medicina personalizada. Para esto será necesario que haya una comunicación entre los distintos niveles de información en salud, que van desde los niveles más bajos (nivel molecular o incluso menor, gracias a los avances de la nanotecnología), hasta los niveles poblacionales (epidemiología).

Tanto para poder interpretar de manera conjunta los datos ómicos generados por las nuevas técnicas de alto rendimiento, como para poder aplicar esta información a la práctica clínica, es necesario realizar un esfuerzo en la integración de los datos. El análisis de estos datos de manera manual no es posible. Es por ello que durante los últimos años han surgido diversas soluciones con el objetivo de integrar datos biomédicos.

Sin embargo, todas estas soluciones sirvieron en su momento para problemas concretos en escenarios concretos, no existen todavía modelos claros de sistemas de información integrados que permitan abordar problemas de integración genéricos. Esto es debido a la importancia de diversos factores dependientes del escenario en el sistema de integración.

4.1 Planteamiento

Basándose en la problemática que se ha descrito anteriormente en esta tesis se plantea la siguiente hipótesis:

El desarrollo de un marco metodológico para la integración de datos biomédicos entre los distintos niveles de información en salud permitirá el desarrollo de sistemas de información aplicables a la medicina personalizada.

4.2 Objetivos

En base a la descripción de la problemática, se define como objetivo principal de la tesis el desarrollo de un marco metodológico para el desarrollo de un modelo de integración de datos que defina las necesidades de intercambio de información de los diferentes tipos de niveles de información en salud. Este modelo debe permitir utilizar el mayor número de soluciones de las tecnologías de la información y las comunicaciones posible.

4.2.1 Objetivos específicos

- Análisis de los distintos tipos de datos biomédicos que será necesario integrar, en función no sólo del tipo de dato, sino también de aspectos técnicos como la localización, estructura e interfaz de las bases de datos donde se encuentran.
- Diseño de un marco metodológico de integración de datos biomédicos que permita utilizar las soluciones tecnológicas actuales, dependiendo del problema de integración que se presente. Este modelo debe permitir la obtención de nueva información biomédica relacionando los datos en los distintos niveles de información en salud.
- Validación del marco metodológico por medio del desarrollo de varios prototipos de integración de datos en escenarios representativos. Este paso permitirá evaluar el marco metodológico propuesto y probar hipótesis planteada en el ámbito de aplicación biomédico.

5 Análisis conceptual

En este capítulo se realizará un análisis de los factores que hacen de los datos biológicos un caso especial de integración de información. Posteriormente, se propondrán una serie de soluciones para este tipo de problemas.

5.1 Análisis

Como se ha comentado en la introducción, la biomedicina del siglo XXI se ha convertido en una tarea en la que se genera una gran cantidad de datos, con una gran variabilidad y complejidad, inherente al modelo biológico. Tanto en la investigación básica como en la clínica estos datos deben confirmar o rebatir las teorías y modelos biológicos que construyen los investigadores. Además de estas tareas, la nueva investigación biomédica tiene el reto de permitir el intercambio de datos entre diversos laboratorios y sistemas de información.

El problema de la gestión (que incluye el modelado, almacenado y consulta) de estos datos no se ha resuelto satisfactoriamente, y no sólo tiene que ver con el gran tamaño de los datos que se tratan, como recuerda Achard (Achard et al., 2001) sino, sobre todo, con los siguientes factores conceptuales y técnicos.

5.1.1 Factores conceptuales

5.1.1.1 Número de bases de datos

Anualmente se publica un número especial en la revista Nucleid Acids Research que estudia las bases de datos de diferentes campos de la biología molecular. Si en 2001 había 96, en el número de 2011 (Galperin and Cochrane, 2011) se listan un total de 1330 bases de datos sólo de temas relacionadas con la biología molecular. Un factor fundamental en la integración de datos es cuestionarse la necesidad de semejante cantidad de bases de datos. Las razones son que, en primer lugar, existe un número relativamente pequeño de bases de datos con colecciones de datos primarios, que luego son utilizados por otras colecciones de datos secundarios o terciarios. Por otro lado, existen muchos tipos de datos, cada uno adecuado a su propia comunidad y tipo de dato. Cada vez que aparece una nueva disciplina se crea un cierto número de bases de datos que imposibilita la creación de repositorios centralizados de datos. Consecuentemente, cada tipo de dato hace que se multipliquen las fuentes de datos,

muchas de ellas replicadas y parcialmente solapadas, que presentan un gran número de tipos de datos en común.

5.1.1.2 Complejidad de los datos

Como comenta Goble (Goble and Stevens, 2008), el número de bases de datos es cada vez mayor y el número de datos almacenados en ellas se incrementa exponencialmente. Pero la cantidad de los datos no debería de ser un problema, teniendo en cuenta las mejoras en cuanto al rendimiento de los equipos informáticos y los algoritmos de procesado de esos datos.

Un factor mucho más decisivo de cara a la integración es el de la variabilidad y complejidad de los datos. Los datos biomédicos, además de su tamaño, se caracterizan por su gran complejidad. En lo referente al tamaño del conjunto de datos, por ejemplo, en un estudio epidemiológico, se almacenan un gran conjunto de datos, acompañados de los metadatos, y estos, por cada paciente. Pero es necesario también almacenar datos temporales de los pacientes, ya que sus características pueden cambiar a lo largo del tratamiento (p.e. estudios de expresión, proteómica, etc.).

En lo referente a la complejidad de los datos biológicos, los sistemas de integración de información tradicionales han sido desarrollados para modelar sistemas creados por el hombre, como pueden ser organizaciones empresariales, datos económicos, etc. y no funcionan correctamente con datos complejos como los datos biológicos. Dicha complejidad se debe, entre otras cosas a la dificultad de definir algunos conceptos a través de condiciones necesarias y suficientes. Esto es debido, según Bernstam (Bernstam et al., 2010) a que los sistemas biológicos son sistemas que han evolucionado naturalmente, en contraposición con los sistemas que han sido creados por el hombre. Este proceso evolutivo implica un conjunto cíclico de selecciones, copias y modificaciones evaluadas por medio de la selección natural. Los sistemas que evolucionan tienden a expresar propiedades que son complejas de expresar matemáticamente y por lo tanto de computar. Otra característica es que no son naturalmente descomponibles. Bernstam pone el ejemplo de un avión y un pájaro. En el primero tiene dos sistemas, las alas para crear sustentación y los motores para generar la fuerza. En el segundo, las alas del pájaro no son tan claramente

descomponibles, sin embargo son las que proporcionan sustentación y fuerza, además de otras funciones como conservar la temperatura, protección, etc. De esta manera, no es posible modelar las funciones del ala de un pájaro de manera aislada de otras. En los sistemas evolucionados no se pueden definir fronteras claras entre los distintos componentes, excepto por aproximación.

La evolución tiende a satisfacer condiciones, pero no a optimizar hasta la suficiencia de esas condiciones. Esto implica que habrá cierta variabilidad dentro de una población. Esto hace que dos individuos no tengan por qué ser iguales, sin embargo dos elementos de ingeniería tienden a serlo en las características más importantes. Esto implica que dos partes de ingeniería reaccionarán igual ante las mismas condiciones, sin embargo dos sistemas evolucionados no tendrán por qué hacerlo. Esto no implica que un sistema biológico sea necesariamente más complejo cuantitativamente que uno artificial, sin embargo la modelización de un sistema biológico sí que es mucho más compleja que la de un sistema artificial.

Por otro lado, en muchas ocasiones no se conoce exactamente la relación biológica entre los datos y, aún conociéndola, probablemente esté incompleta u obsoleta en poco tiempo, como se verá en el siguiente punto. Otros aspectos que intervienen en la complejidad de los datos biomédicos son la descripción del ejemplo, la calidad del ejemplo, etc.

5.1.1.3 Alta variabilidad de los datos

Diariamente se realizan nuevos descubrimientos que incrementan el número de tipos de datos que requieren ser modelados, por ejemplo, el campo de las variaciones de ADN ha pasado por microsatélites, SNPs, haplotipos y CNVs. Este tipo de inclusiones no sólo provocan cambios en el modelado, sino que también cambian la percepción que se tiene de los viejos datos en el modelo. Aparecen nuevas relaciones en conceptos que no tenían ninguna relación previa o, si la había, modifican dicha relación. Esto significa que la aparición de nuevos datos provoca una modificación del modelo, que debe ser actualizado.

5.1.1.4 Datos sobre el análisis de datos

Los datos por si solos no son suficientes para generar nuevo conocimiento. Una vez generados, estos deben ser analizados. La generación de estos datos provenientes del análisis de datos, deben ser incluidos también en el modelo e integrados con el resto. Es recomendable también almacenar el modo en el que estos datos han sido analizados, para así poder replicarlos.

5.1.1.5 Almacenamiento de los datos crudos

El hecho de transformar los datos para poder acceder a ellos de una manera más simple, u obtener resultado de análisis, no significa que las fuentes originales de los datos deban descartarse o eliminarse. Estas fuentes deberán permanecer almacenadas para poder replicar experimentos o confirmar los resultados obtenidos con modelos computacionales. Por ejemplo, en los análisis de microarrays, la información de diferencia de expresión es la que finalmente se usa, pero todos los datos intermedios deben ser almacenados (imágenes, datos de intensidad de fondo y de intensidad de señal para cada canal, etc.).

5.1.1.6 Necesidad de datos actualizados

Los datos de cada fuente de datos son actualizados cada día, y lo que es más importante, son accedidos e intercambiados por investigadores a través de internet muy frecuentemente. Es necesario que los investigadores dispongan de versiones actualizadas de los datos.

5.1.1.7 Sistemas de acceso simples

Los sistemas de integración deberían de ser suficientemente simples como para que los distintos tipos de usuarios (informáticos, pero también investigadores biomédicos) de sistemas de integración pudiesen manejarlos, construir consultas complejas, etc.

5.1.1.8 Sensibilidad de los datos

Independientemente de la sensibilidad de los datos clínicos, que está reconocida en la legislación de la mayoría de los países, es también importante considerar la sensibilidad de los datos de otros niveles. Una serie de variaciones en una secuencia de un individuo particular puede aportar suficiente información para

identificar al individuo o deducir posibles variaciones en su fenotipo (Godard et al., 2003, Homer et al., 2008). Esto establece ciertos problemas éticos (Knoppers et al., 2006, Van Hoyweghen and Horstman, 2008), ya que aunque los datos del genoma sean anonimizados, existe el riesgo de re-identificar los individuos basándose en su perfil de variaciones. Actualmente, la mayoría de las bases de datos intentan mostrar la información suficiente para que un individuo no pueda ser identificado, sin embargo esto no es una tarea simple.

5.1.2 Factores técnicos

5.1.2.1 Gran cantidad de datos

Como se ha comentado anteriormente, ha habido un incremento exponencial de datos en cada una de las bases de datos. Esto implica un gran problema técnico en términos de gestión de datos y hace necesarias soluciones escalables.

5.1.2.2 Almacenamiento distribuido

Los datos se encuentran en bases de datos diseminadas a través de diferentes localizaciones geográficas, muchos de ellos duplicados en varios repositorios.

5.1.2.3 Heterogeneidad de nombrado

Como se comentó en el punto anterior, un mismo elemento puede aparecer en distintas bases de datos, y cada base de datos puede nombrarlo de una manera distinta. Esto supone un problema importante a la hora de identificar un mismo objeto o entidad en varias bases de datos distintas.

5.1.2.4 Heterogeneidad de formato y acceso

Cada base de datos puede tener un formato heterogéneo, muchas bases de datos parten de ficheros de texto plano sin ningún tipo de estructura, formatos de asociación par/valor, formatos semi-estructurados como XML, servicios web o bases de datos estructuradas accesibles por SQL. Esto es debido a que los grupos de investigación generan bases de datos de manera autónoma. Por otro lado, muchos sistemas de integración cambian el formato de salida cada poco tiempo. Esto provoca que los clientes dejen de funcionar.

5.1.2.5 Modelo de datos deficiente

Esta autonomía provoca que cuando un grupo decide crear una nueva base de datos, el encargado de crearla no siempre posee los conocimientos técnicos de un diseñador de bases de datos, con lo cual los modelos a menudo no son del todo correctos.

5.1.2.6 Volatilidad de los datos

Anualmente se generan nuevas bases de datos, pero según un estudio (Merali and Giles, 2005), sólo el 18% de ellas sobrevive, el resto desaparecen o son sustituidas por bases de datos más completas que incluyen los mismos datos, en la mayoría de los casos por falta de financiación.

5.1.3 Soluciones

En primer lugar es importante dejar claro que más que ningún otro factor, la complejidad de un sistema de integración de información biomédica depende en gran medida del número de bases de datos y de la complejidad de los modelos que se pretendan integrar. Posteriormente se deben tener en cuenta el resto de los factores, pero estos dos factores son claves a la hora de determinar el modelo de integración. No tiene sentido crear un modelo muy complejo de integración de datos cuando el problema de integración es simple o implica solo la integración de una parte de dos bases de datos.

A la hora de tratar con el problema del número de bases de datos, la complejidad de estos, la variabilidad de sus esquemas, etc. parece claro que cuanto mayor sea el problema o cuanto mayor tenga que ser su capacidad de modificación a lo largo del tiempo, más flexible tendrá que ser el sistema de integración. Conforme aumenta la complejidad de los modelos de la base de datos, el sistema de integración debe de ser dinámico, capaz de adaptarse a los nuevos modelos de datos, realizando pequeños cambios en su modelo o esquema, sin que la modificación de una base de datos afecte a la representación del resto.

Esta flexibilidad es necesaria si se pretende incluir en el modelo los datos provenientes del análisis de una parte de los datos, ya que un modelo rígido debería de modificarse profundamente cada vez que se quisiese incluir un nuevo tipo de

análisis. Del mismo modo, en ciertos modelos puede ser útil, en lugar de ir hacia adelante, incluyendo en el modelo datos del análisis, ir hacia atrás e incluir datos anteriores a los datos originales (datos previos al preprocesado o análisis, datos crudos, etc.).

Una aproximación, en lugar de incluir los resultados del análisis, es permitir que los investigadores repliquen el análisis sobre los datos. Existen diversas alternativas, como las que exponen Dudley y Butte (Dudley and Butte, 2010), entre las que están el almacenamiento de un conjunto de *workflows* estandarizados, imágenes virtuales de sistemas completos que cuenten tanto con el software como con los datos, de manera que se pueda repetir el análisis exactamente en las mismas condiciones o directamente análisis de los datos en la nube.

Dependiendo del tipo de problema de integración, la necesidad de tener unos datos actualizados puede ser más ineludible o menos. En este sentido, si el sistema debe disponer de los últimos datos generados (por ejemplo, un sistema que integre información clínica), debe de estar conectado obligatoriamente a las bases de datos originales, mientras que si se sabe que las bases de datos no cambian en mucho tiempo, se puede almacenar en un repositorio local una copia de esas bases de datos. Existen también soluciones intermedias, como la de almacenar una copia local de la base de datos que se está usando, pero actualizándola periódicamente. Todos estos factores dependen bastante de los recursos técnicos, principalmente de la capacidad de almacenamiento y ancho de banda de la red.

La facilidad de uso del sistema de integración forma parte de un requisito del desarrollo de la capa vista del sistema de integración, pero los mecanismos de acceso de esta capa vista son los que realmente permiten que el desarrollo de interfaces simples sea sencillo. El sistema de integración debe de proveer de interfaces completos y complejos que permitan realizar consultas simples sobre el modelo o consultas complejas. Esto permitirá que el desarrollador de la vista pueda crear distintos interfaces dependiendo del tipo de usuario para el que esté desarrollado el sistema. También es necesario que los interfaces permitan comunicarse con otros sistemas, no simplemente con la capa vista del sistema de integración. Una vez

desarrollado el sistema de integración, este debe poder pasar a formar parte de otro sistema distribuido como una fuente de datos más.

La heterogeneidad de los datos es un problema técnico ya solventado con el uso de *wrappers*, como se vio en el capítulo II. Ya sea en la etapa de centralización (*warehouse*) como en la etapa de consulta (federación), los *wrappers* solucionan el problema de la heterogeneidad transformado los datos del formato de sus fuentes originales a un formato homogéneo. El uso de *wrappers* también puede solucionar en cierta manera la utilización de un modelo de datos deficiente, rellenando las carencias del modelo con información "nula", de manera que se pueda mantener la consistencia del modelo, siempre que sea posible.

El problema de las fuentes de datos distribuidas geográficamente puede solucionarse centralizando el almacenamiento de los datos, con el inconveniente de que la gestión de los datos se complica aún más. Por lo tanto, es necesario encontrar un compromiso, entre el esfuerzo técnico que conlleva un sistema descentralizado, con las dificultades de gestión de un repositorio centralizado, teniendo en cuenta los requisitos de rendimiento del sistema. El uso de un modelo centralizado también solucionaría en parte los problemas de la volatilidad de datos, ya que tanto si los datos están en un repositorio centralizado, como si se tiene una copia de la fuente de datos original, la desaparición de la fuente original no conlleva más problemas que la falta de actualizaciones. Se profundizará sobre las ventajas / inconvenientes de un modelo centralizado / descentralizado más adelante.

Desde el punto de vista del cliente, el problema de la heterogeneidad de formato debe solucionarse definiendo desde el principio un interfaz estable, con formatos, mensajes y protocolos comunes y conocidos.

Desde el punto de vista de la legalidad y la seguridad, en el caso de que se trabaje con datos clínicos, deben tomarse todas las medidas de securización necesarias para asegurar la confidencialidad de los datos. Dependiendo de la legislación y la política de protección de datos de cada fuente de datos, será necesario anonimizar los datos que lleguen al sistema de integración.

Tal y como apunta Goble (Goble and Stevens, 2008), existe una necesidad clara de la utilización de identidades y nombres comunes. No importa qué método de integración o qué esquema se esté usando, todas las aproximaciones necesitan almacenar registros que se corresponden con el mismo objeto. Por ejemplo, la proteína WS-1 tiene más de 10 nombres y más de 21 números de acceso distintos. Stein (Stein, 2003) proponía en 2003 la noción de entidades de nombrado con distintos *namespaces* de facto, este tema es bastante confuso y controvertido. Dos entradas de distintas bases de datos pueden ser claramente diferentes, pero que la entidad o entidades que representan sean idénticas o equivalentes, puede no ser tan claro.

Una vez está cubierta la necesidad de un sistema de nombrado homogéneo, el siguiente paso es definir una semántica común. Esta semántica incluye el paso anterior, definiendo no solo los conceptos sino también las relaciones entre ellos. La integración de datos que tienen conceptualizaciones y representaciones diferentes provoca que los análisis de datos biomédicos sean muy complejos. La comunidad reconoce la necesidad de estándares tanto para los esquemas como para los valores de los datos. Existen recomendaciones sobre "información mínima" en varios contextos, como el MIAME (Minimum Information About a Microarray Experiment) (Brazma et al., 2001) o MIAPE (Minimum Information about a Proteomics Experiment) (Taylor et al., 2007).

Por otro lado, el elemento clave en la capacidad de un sistema de integración de trabajar correctamente con múltiples bases de datos son las ontologías. La existencia de una o varias ontologías compartidas permite al sistema de integración combinar múltiples bases de datos con la garantía de que esos términos son equivalentes, a pesar de ser nombrados de manera distinta. Además, las ontologías no son solo vocabularios controlados, sino que definen conceptos describiendo las relaciones con otros conceptos proporcionando las relaciones semánticas mencionadas anteriormente. Esto permite llevar los sistemas de integración a otro nivel semántico. El uso de las anotaciones con ontologías permite afrontar el problema de la complejidad de los datos desde otro punto de vista, permitiendo definir los datos y las relaciones entre ellos de manera que este tipo de relaciones puedan ser mapeadas también en el sistema de integración.

5.1.3.1 Problemas de las ontologías biomédicas

En este punto es necesario hacer un inciso con respecto a las ontologías biomédicas. En un principio, pueden parecer la solución a los problemas más importantes de la integración de datos biomédicos, la heterogeneidad del nombrado y la diversidad y complejidad de los modelos de datos a integrar. Pero para usar las ontologías en todo su potencial, los conceptos, relaciones y axiomas que las componen deben ser compartidos y permanecer accesibles para todo el mundo. Desafortunadamente, cada ontología biomédica parece haber sido desarrollada como una porción de información independiente en la que cada parte del conocimiento está completamente definida. Este aislamiento de una parte importante de las ontologías biomédicas imposibilita la reutilización del conocimiento y complica la integración de los datos (Pasquier, 2008) (Soldatova and King, 2005).

5.2 Análisis de la arquitectura

Se partirá de la clasificación de Hernández (Hernandez and Kambhampati, 2004) para hacer el análisis, aunque posteriormente se verán otros modelos que se podrían incluir en alguna de las categorías de esta clasificación.

5.2.1 Modelos de navegación por links

El concepto de navegación por links, aunque aparece en gran parte de la bibliografía como una de las tres principales formas de integración de datos biomédicos, no parece una alternativa real a las otras dos aproximaciones. Esta aproximación es muy usada por la comunidad biomédica, como se vio en el capítulo de estado de la cuestión, pero realmente no se considera un tipo de arquitectura de integración (Davidson et al., 1995). Esto es debido a que no ofrece suficiente funcionalidad en las consultas que puede realizar y, por otro lado, porque no se adapta bien a la gran cantidad de datos y a la naturaleza cambiante, tanto de los datos, como de los interfaces que almacenan estas fuentes de datos. Para que funcione bien debe de haber una gran implicación entre todas las bases de datos que se integran. Es muy vulnerable a los conflictos de nombre y a las ambigüedades. Realmente se trata más de una unión de enlaces que de integración. La integración y la interpretación se realizan por medio de otro mecanismo.

Generalmente este modelo de integración, aceptando que lo sea, sólo puede funcionar bien cuando se trata del desarrollo de un sistema de navegación entre las bases de datos de una misma organización. Así, los ejemplos más notables de este tipo de integración son de grandes organizaciones (SRS e Integ8 del EBI y Entrez del NCBI) que gestionan multitud de bases de datos y no de laboratorios pequeños.

5.2.2 *Data Warehouse* vs Modelo Federado

A continuación se estudiarán los principales pros y contras de los modelos federados y los modelos basados en *warehouse* en lo que respecta a la integración de datos biomédicos. El modelo centralizado que representa el *warehouse* implica que hay una gran cantidad de información que se almacena en una o un conjunto de bases de datos centralizadas o almacén (*warehouse*) tal y como se vio en el capítulo de fundamentos. Desde ese almacén central es donde toda esta información debe de ser gestionada y es donde se deben proporcionar las herramientas de búsqueda y representación de la información contenida en el *warehouse*. Por el contrario, en el modelo federado simple, todos los datos se almacenan en un conjunto de bases de datos dispersas geográficamente, sin existir un intercambio de información regular entre ellas. Las búsquedas se llevan a cabo a través de sistemas que consultan todos los contenidos disponibles y cada una de las bases de datos presenta su propio modelo de búsqueda y representación de datos.

La aproximación federada presenta varias ventajas con respecto a *warehouse*, como puede ser que no se requiere la inversión en hardware necesaria para poder alojar un *warehouse* que permita almacenar todos los datos que se requieren. Por otro lado, el modelo federado proporciona siempre acceso a la última versión de los datos debido a que los datos provienen siempre de la fuente original, mientras que los *warehouses* deben de actualizarse regularmente para poder proporcionar los últimos datos. Esto aplicado a grandes bases de datos implica un gran consumo de ancho de banda, salvo que las bases de datos públicas proporcionen en lugar de la base de datos completa, únicamente las modificaciones, situación que no se da regularmente.

Las principales ventajas de los sistemas *warehouse* radica en su centralización que permite una mayor facilidad de escalado, mejor rendimiento en las consultas, una

alta disponibilidad y, por encima de todo, mayor control sobre los datos (Wong, 2002). Por el contrario los modelos federados representan una solución más complicada, mayor latencia en recuperar los datos y una pérdida del control de estos, ya que estos pertenecen a las fuentes de datos originales. Además, la disponibilidad del conjunto del sistema depende de que la base de datos que contiene alguno de los datos que se quieren recuperar se encuentre disponible. A mayor cantidad de fuentes del sistema, mayor posibilidad de que el sistema no funcione correctamente debido a la falta de algún dato clave alojado en alguna fuente de datos que podría no estar accesible.

Sin embargo, el control total de los datos puede convertirse en una desventaja, si se mira desde el punto de vista de los propietarios de la base de datos. En un modelo federado, los datos pertenecen a los equipos de investigación que los han obtenido y publicado, por lo que el control y el reconocimiento de los datos no pertenecen al grupo que integra, sino al grupo que generó los datos originales. Esto puede ser muy importante en el caso de los datos sensibles ya que, por ejemplo, el propietario de la base de datos puede escoger qué conjunto de datos proporciona o no de acuerdo a las posibilidades de identificación de un individuo. Por otro lado, la federación de los datos distribuye la gestión y el control de calidad entre los participantes en la federación. Generalmente esto asegura que estos datos tienen una calidad mayor que los datos que pueden encontrarse en modelos centralizados, ya que el control de calidad de las bases de datos no debería de centrarse únicamente en la primera etapa de los datos, sino que estos datos deberían de curarse continuamente. Wong (Wong, 2002) opina lo contrario, es decir, que un *warehouse* permite un mayor control de los datos debido a que en los datos originales puede haber muchos errores. En este trabajo, por el contrario, se considera que esta labor es mucho más abordable dentro de la propia base de datos de una organización que en un *warehouse* que contiene datos de multitud de grupos. Por otro lado, los dueños de una base de datos en un modelo federado mantienen la autoría y el control de sus propios datos, pero proporcionan acceso a ellos desde fuera, mientras que en un modelo *warehouse*, el usuario puede perder control de quién es el responsable de los datos a los que está accediendo.

Otra ventaja de los modelos federados con respecto al warehouse en biomedicina es que en un modelo centralizado, los modelos de datos se han hecho ad-hoc, lo que generalmente implica que una parte de los datos y de las relaciones entre estos datos que estaban en las bases de datos originales se pueden haber perdido. En los modelos federados sin embargo, los mecanismos de accesos son más sofisticados, lo que permite consultar (en el caso de que el interfaz no restrinja esta ventaja) directamente a la base de datos de manera que se puedan obtener una mayor cantidad de datos e inferir las relaciones entre ellos, que en los modelos *warehouse*. Esto es particularmente importante en el campo de la biomedicina, donde como se comentó al principio de este capítulo, los datos son especialmente complejos, tanto en el "que" representan como en "cómo" lo representan. Unos mecanismos de acceso más complejos permiten obtener tanto los datos como los metadatos necesarios para contestar las preguntas complejas que surgen diariamente en la biomedicina, mientras que en un *warehouse* generalmente no se podrían obtener.

A pesar de que ambos modelos se han aplicado exitosamente en algunos problemas de integración de los que se han visto en el estado del arte del capítulo III, posiblemente ninguno de los dos modelos, en la versión más extrema, sea aplicable a un problema realista de integración compleja de datos biomédicos, debido a las limitaciones que presentan ambos modelos. Estudiando los modelos revisados en el capítulo III se aprecia una clara tendencia al cambio de un modelo centralizado a un modelo federado semántico. Los primeros sistemas se basaron en modelos centralizados, pero con la mejora de los sistemas de comunicación, de los sistemas gestores de bases de datos y de la ingeniería ontológica, la tendencia ha ido cambiado a modelos más o menos federados con componentes semánticas. Parece lógico pensar que será necesario un modelo híbrido que aúne las mejores características de cada uno de ellos.

Por lo tanto, después de analizar estos pros y contras, se puede afirmar que los sistemas federados proporcionan generalmente una mejor solución a los problemas de integración de datos biomédicos en el caso de que el problema de integración no sea una aplicación monolítica, cuyos requisitos de integración sean conocidos de antemano, sea posible el acceso a las bases de datos completas y no se requiera que

los datos estén completamente actualizados. Por el contrario, se desaconseja el modelo federado en casos en los que sea necesario un control total sobre los datos o un gran rendimiento.

Como se ha visto hasta ahora, en la mayoría de los casos un modelo federado es generalmente el modelo más adecuado. A partir de este punto se va a centrar el resto de los puntos de la arquitectura en el marco metodológico en modelos federados.

5.2.3 Otros modelos de federación

5.2.3.1 Federación basada en Workflows

Como se ha mencionado anteriormente, el sistema de federación de workflows es tremendamente útil cuando el objetivo es que una persona sin muchos conocimientos de informática pueda realizar sus propias consultas a partir de encadenar una serie de servicios visualmente. Sin embargo, la integración que se realiza de esta manera o es muy limitada, o necesita el desarrollo de servicios de integración específicos (mediadores intermedios) que realicen tareas de integración más complejas.

Por otro lado, es necesario que estén implementados servicios que trasladen los datos de la fuente de datos al modelo común, en otro caso habría que implementarlos.

Otra ventaja importante de este sistema es que a los resultados intermedios o finales pueden aplicárseles diferentes tipos de procesados que estén implementados como servicios. Esta ventaja no la tienen el resto de los modelos de integración analizados.

Existen *frameworks* de *workflows*, como Biomoby, que podrían considerarse sistemas de *workflows* federados, debido a que incorporan un modelo de datos global, que se amplía o extiende cada vez que algún usuario crea un nuevo tipo de dato (debido a que tanto los tipos de datos como los servicios permanecen públicos y a disposición libre de cualquier investigador). Los diferentes grupos de investigación que usan Biomoby han creado también un conjunto importante de servicios de procesado,

lo cual hace de este sistema una gran opción si lo que se busca es una integración sencilla, visual y con capacidades de procesado.

5.2.3.2 Federación basada en sistemas multi-agente

Este sistema de integración es útil en los casos de integración de información donde sea necesario que el sistema deba de tomar decisiones de manera inteligente, autónoma, reactiva y proactiva. Por ejemplo, el agente mediador debe decidir, dependiendo de la consulta, qué recursos son más adecuados para la consulta que le llega, tanto a nivel del origen de datos, como del tráfico de la red en un momento concreto. Para esto, periódicamente debe consultar los esquemas de cada una de las bases de datos que forman cada concepto para saber a cuál debe consultar. También debe ser capaz de reaccionar ante la caída de una de las bases de datos recurso, modificando sus planes de consulta.

Otro de los beneficios del uso de sistemas multi-agente para la integración de datos biomédicos es que existen varios *frameworks* (Tobias and Hofmann, 2004, Nikolai and Madey, 2009) de desarrollo de agentes que incorporan las características básicas de comunicación, paso de mensajes, así como sistemas de anotación semántica propios. Esto simplifica enormemente el desarrollo de este tipo de sistemas.

El mayor beneficio del uso de sistemas multi-agente en sistemas de integración de datos biomédicos es el dinamismo que aportan. Al contrario que los sistemas basados en *workflows*, un sistema basado en agentes puede buscar y utilizar datos y servicios on-line que pueden aparecer a través de internet. La metodología de este tipo de sistemas a un entorno biomédico incluye el diseño de una ontología de dominio como modelo general y la anotación semántica de los recursos de los que vaya a hacer uso el sistema.

5.2.3.3 Federación p2p

Desde los años 90, han emergido nuevos modelos de intercambio de datos descentralizados, denominados *peer to peer* (P2P) (Schollmeier, 2001). Estos modelos distribuidos han sido usados para integrar historias clínicas electrónicas (Kilic et al., 2010) así como tener sistemas PACS distribuidos (Costa et al., 2009, Costa et al., 2011) y pueden también ser usados a modo de sistema de integración federado de datos.

Tal y como apunta Louie (Louie et al., 2007), una de las mayores desventajas en los sistemas de integración de datos que necesitan un modelo global es la generación de este esquema. A medida que se incluyen más y más fuentes de datos, el esquema global es más complejo y más difícil de mantener. El esquema federado P2P se basa en que cada uno de los *peers* puede ser una fuente de datos, o un mediador que proporciona una parte del esquema global de los de otros *peers* que hacen las veces de fuentes de datos. Esto crea una red de recursos semánticos que forman un esquema global general en el cual todos los datos están representados. Estos *peers* se pueden organizar de manera jerárquica de manera que existen uno o varios nodos *super-peers* sobre los cuales los clientes pueden hacer consultas. Estos nodos desvían sus consultas a otros *peers* mediadores y así sucesivamente hasta que la petición de información llega hasta los *peers* fuentes de datos.

Este tipo de sistema es imposible sin la existencia de una completa anotación semántica de los *peers* y un *mapping* entre las distintas redes semánticas.

Si bien este modelo parece muy novedoso, puede verse como un tipo de modelo federado de sistemas multi-agente en el que el modelo global está distribuido, de una manera jerárquica o similar, por todos los *peers*-agentes mediador.

Por lo tanto, este sistema de federación es aconsejable cuando lo sea también la aproximación de federación de sistemas multiagente y además el esquema global sea tan grande que sea imposible poder gestionarlo de un mediador simple.

6 Marco metodológico propuesto

En este capítulo se describirá el marco metodológico propuesto para la integración de datos biomédicos.

Debido a la diversidad de los problemas de integración que puede plantear cualquier sistema que trabaje con datos biomédicos, este marco metodológico no pretende definir una serie de pasos rígidos para aplicar directamente al problema de integración, sino que debe de ser tomada como una guía de recomendaciones y consideraciones cuando se afronta un problema de integración de datos biomédicos.

6.1 Análisis de requisitos

El primer paso consiste en realizar un análisis de requisitos del sistema de integración. Para esto se deben utilizar las metodologías de análisis de requisitos básicas del desarrollo de software (Jackson, 1995, Pressman, 2010), pero es importante tener en cuenta los factores que se han mencionado anteriormente.

6.2 Análisis de casos de uso

El segundo paso de este marco metodológico consiste en realizar un análisis de casos de uso (Bittner and Spence, 2003) que permita identificar las necesidades concretas del usuario, centrándose especialmente en las indicadas anteriormente.

6.3 Análisis de los datos a integrar

En este tercer paso, es importante realizar un profundo análisis de los datos que se pretenden integrar. Este análisis debe consistir en una búsqueda exhaustiva de todas las fuentes de datos que tengan los datos que se van a necesitar. De cada una de las fuentes de datos será necesario analizar:

6.3.1 Contenido de los datos

Se debe analizar si la fuente de datos dispone de los datos que se necesitan para el proceso de integración. Es importante buscar en la bibliografía por el número de referencias de que dispone la base de datos y cotejar sus usos en la bibliografía con los usos que se le dará durante el proceso de integración. Este paso también permite descartar fuentes de datos volátiles, ya que si se seleccionan fuentes de datos muy

utilizadas por la comunidad científica, es menos probable que estas bases de datos desaparezcan.

6.3.2 Interfaz

Dependiendo del tipo de modelo que se vaya a utilizar para realizar el análisis de datos, es importante analizar de qué manera se puede acceder a los datos. Es necesario comprobar si la fuente de datos está accesible para descarga, o si está accesible utilizando algún tipo de API, servicios web o conexión directa con su base de datos o si será necesario obtener los datos de una web. Este tipo de acceso determinará también el rendimiento en términos de tiempo de respuesta si finalmente se escoge un modelo descentralizado.

Una vez estudiado el tipo de acceso, hay que estudiar cómo se recuperan estos datos, si se obtendrá un fichero plano, un XML semi-estructurado o incluso extrayendo información de una página web de manera automática. Esta fase determinará la complejidad del *wrapper* que será necesario implementar.

6.3.3 Actualizaciones

El número de referencias de esa base de datos en la bibliografía dará una idea de la utilización de esta fuente y, muy probablemente, de su nivel de actualización. La utilización de bases de datos que cambian a menudo puede ser un factor determinante para escoger entre un modelo centralizado y descentralizado.

6.3.4 Modelo de datos

Dependiendo de la complejidad del modelo de integración que se vaya a utilizar y de la complejidad del modelo biológico que se vaya a representar, el hecho de que la base de datos de origen parta de un modelo de datos correcto puede facilitar mucho las tareas de integración.

6.3.5 Anotación semántica

El punto clave, una vez más, para poder realizar una integración de datos compleja es la anotación semántica de estos datos dentro de la base de datos original. Una correcta anotación es imprescindible para el posterior proceso de integración. Actualmente las principales bases de datos biológicas no solo disponen de una

anotación uniforme sino que, cada vez más, se están anotando con relaciones semánticas que facilitan la labor de integración.

6.3.6 Autonomía, heterogeneidad y distribución

De manera formal, se propone para realizar el análisis de cada uno de estos puntos, utilizar la dimensión ortogonal propuesta por Busse, adaptada a datos biomédicos:

- Autonomía: Cada base de datos biomédica presenta un cierto nivel de autonomía que debe ser estudiado. Se distinguen:
    - Autonomía de diseño: Cada base de datos es propia del laboratorio u organización que la creó, por lo que tiene total autonomía para modificar su modelo de datos y nombrado de conceptos. Es importante tener en cuenta este punto de cara a la inclusión en un sistema descentralizado, debido a la posibilidad de posibles modificaciones
    - Autonomía de comunicaciones: Cada base de datos es libre de comunicarse con los sistemas externos. Esto es importante en el sentido de la disponibilidad de la base de datos y a la volatilidad de los mismos.
    - Autonomía de ejecución: La base de datos biomédica ejecutará las consultas según su propia planificación. Este punto es importante de cara al rendimiento del sistema, debido a que algunas bases de datos biomédicas penalizan la ejecución de consultas al aumentar la frecuencia de estas. En este punto también se incluyen las actualizaciones de cada base de datos.
- Heterogeneidad: Generalmente las bases de datos biomédicas se crean para un propósito específico, con lo cual son heterogéneas con respecto a los siguientes aspectos:
    - Heterogeneidad sintáctica:
        - Heterogeneidad tecnológica y de interfaz: Se refiere a la tecnología utilizada, los protocolos de acceso y la interfaz de acceso (SQL, servicios web, etc). Como se ha comentado antes, esta característica determinará la complejidad del *wrapper* que se use. Dentro de estas se podrían distinguir también:

        - Heterogeneidad del lenguaje que se use para realizar la consulta.
        - Heterogeneidad de consulta: Restricciones de la consulta, que posibilitan o no el acceso a la totalidad del modelo.
    - Heterogeneidad del modelo de datos: Cada base de datos presenta su propio modelo de datos, en el sentido de que tipo de modelos de datos (semántico, objetual, relacional, etc.).
    - Heterogeneidad lógica:
        - Semántica: Heterogeneidad semántica referida al esquema del modelo. Es decir, como la base de datos representa los datos que contiene y sus relaciones entre ellos.
        - Esquemática: Cada base de datos puede tener un esquema distinto aunque se refiera a los mismos datos.
- Distribución: Las bases de datos pueden estar localizadas en diferentes lugares geográficos o pueden ser propias del sistema de integración.

Una vez analizados cada uno de estos puntos para todas las bases de datos se descartarán aquellas de las anteriores que peores características presenten y cuyos datos ya se encuentren en otras fuentes de datos aceptadas.

6.4 Selección de la arquitectura del sistema de integración

Tal y como se vio en el capítulo II de esta memoria, existen diferentes aproximaciones para la integración de datos, desde las más simples como la navegación por links a las más complejas como los modelos federados y los *data warehouse*.

La selección de uno u otro depende en gran medida de la complejidad del modelo de integración que se esté tratando y de los requisitos específicos del sistema de integración a implementar.

6.4.1 Modelos de Navegación por links

Este tipo de modelo de integración debe ser tenido en cuenta en entornos en los que el objetivo es proporcionar, más que un sistema de integración, un sistema de acceso homogéneo a distintas bases de datos, preferiblemente si estas están dentro de la misma organización.

6.4.2 Data warehouse

El uso de modelos de integración de *data warehouse* está recomendado principalmente a sistemas que presenten alguno de los siguientes requisitos:

- Rendimiento: Los sistemas *warehouse* proporcionan mejor rendimiento en las consultas al tener todos los datos centralizados en una sola base de datos y no tener que procesar más que la consulta principal.
- Capacidad de escalado: Si el sistema tiene que ser escalado en función de los datos, no en función del modelo.
- Control sobre los datos: Si es necesario validar o curar los datos a los que se va a acceder.
- Invariabilidad del modelo: El modelo que se va a utilizar para el data warehouse no va a variar, las relaciones se mantendrán y no se incluirán nuevas fuentes de datos. No se espera que los requisitos de integración cambien a lo largo del tiempo.
- Disponibilidad de las fuentes: Cuando las fuentes de datos no presenten una alta disponibilidad, se recomienda que los datos estén almacenados en el *warehouse*.

6.4.3 Modelo federado

El uso de modelos de integración federados se recomienda principalmente para aplicaciones que presenten alguna de estas necesidades:

- Necesidad de datos actualizados: Los modelos federados traducen las consultas, no los datos, por lo que siempre se estará accediendo a la última versión de los datos.
- Acceso a los metadatos de las fuentes de datos originales en tiempo de consulta.

- Aplicaciones que puedan tener requisitos cambiantes en función de las fuentes de datos o en las que el modelo de las fuentes de datos pueda variar afectando al esquema global.
- Necesidad de garantizar a las bases de datos originales la propiedad, gestión y calidad de los datos.

6.5 Tipo de modelo de integración

Una vez escogida el tipo de arquitectura, existen diversas aproximaciones de integración dependiendo del problema a tratar. Tal y como se vio en el capítulo II, existen varios parámetros para dividir el modelo de integración. A continuación se estudiará la adecuación de un sistema de integración según el tipo de integración, abierta o cerrada, arquitectura top/down o bottom/up.

6.5.1 Integración abierta vs cerrada

Aunque muchos autores consideren que la integración abierta no es un tipo de integración válido, es necesario considerarla para el desarrollo de sistemas de integración biomédicos. Generalmente la arquitectura de los sistemas warehouse implica una integración cerrada, sin embargo hay excepciones como el LIMBO (Philippi, 2004), que simula una integración abierta utilizando un modelo abstracto (DATA, RELATION y METADATA) o el COLUMBA (Trissl et al., 2005). En los modelos federados también la integración cerrada es la más común, pero en este caso hay más ejemplos de integración abierta, debido a la naturaleza de la federación (Haas et al., 2001) (Ludascher et al., 2001, Min et al., 2009).

Según las características que se plantean en el capítulo II, esta aproximación tiene multitud de inconvenientes, debido principalmente a carecer de esquema global. Las responsabilidades por tanto de integración, mapeado, planificación de consultas, etc. recaen exclusivamente sobre el usuario, que es el que debe realizar las consultas sobre todas las bases de datos. Sin embargo, este modelo también puede tener sus ventajas. El desarrollo de modelos integración globales imposibilita en ocasiones el acceso a ciertas partes de las bases de datos de origen, tanto en datos como en esquema, lo que resta libertad al usuario para usar este tipo de información. Muchas

relaciones que pueden aparecer muy claras en el esquema de la base de datos original pueden perderse porque el modelo global está focalizado en otras tareas.

Por ejemplo, un sistema de integración biomédico basado en esquema global que está centrado en diseño de fármacos. Si un investigador quiere utilizar ese sistema para diseño de estudios epidemiológicos tendrá problemas porque, si bien ambos sistemas utilizan bases de datos similares (pathways, variaciones, etc.), el esquema del sistema de diseño de fármacos está centrado en esa tarea por lo que imposibilitará a otro tipo de usuario utilizar esta infraestructura. Si el usuario quiere desarrollar un portal y tiene conocimientos de informática avanzados, utilizando una arquitectura de federación abierta puede crear sobre su sistema de información las consultas necesarias sin pasar por el esquema global.

El sistema de integración cerrada es aquel que posee un esquema global, ya sea creado ad-hoc o a partir de vistas de usuario creadas en un sistema de integración abierta. Este tipo de sistema tiene como ventajas que facilita al usuario la obtención de información sin tener grandes conocimientos ni informáticos ni sobre el esquema o los datos de las fuentes de datos originales. Como defecto, este sistema resta libertad al usuario y requiere de alguien con un conocimiento importante del dominio que realice los mappings del modelo global.

Por lo tanto, como se ha visto, en el esquema de integración abierta, el usuario es responsable de realizar una parte importante del proceso de integración. Si el usuario quiere desarrollar un sistema del que tenga completo control y necesita tener acceso al esquema completo de las fuentes de datos originales, este modelo de integración es adecuado, en otro caso, se recomienda el uso de un modelo global, es decir, un sistema de integración cerrado.

6.5.2 Estrategias top-down vs bottom-up

Tal y como se explicaba en el capítulo II, las aproximaciones *top-down* consisten en el desarrollo de un esquema global que incluya únicamente los tipos de datos que se hayan identificado en la especificación de requisitos, mientras que la estrategia *bottom-up* implica la inclusión en el modelo global de todos y cada uno de los tipos de datos que forman el conjunto de las bases de datos originales.

Intuitivamente se puede apreciar que teniendo en cuenta la complejidad de los datos biomédicos, la aproximación *bottom-up* sólo es posible llevarla a cabo en el caso de que el número de bases de datos a integrar sea muy reducido o el esquema de cada una de ellas muy simple o muy similar, de otra manera este esquema es prácticamente inabordable. Este modelo es también válido para escenarios en los que una gran institución quiera acceso a todos los datos de sus propias bases de datos o aplicaciones de integración donde a pesar de existir muchas bases de datos distintas, los esquemas son similares, como por ejemplo un sistema de integración de historia clínica de distintos hospitales con distintos sistemas.

El escenario típico de integración de datos no contempla la idea de la posibilidad de lectura/escritura en lugar de sólo lectura. En el extraño caso de que la necesidad de actualización sea un requisito, necesariamente la aproximación deberá de ser *bottom-up* para conservar la integridad de los datos.

El escenario *top-down* tiene, por tanto, otras ventajas, ya que sólo necesita representar en el esquema global los tipos de datos que son necesarios y de la manera en que son necesarios, es decir, con el nivel de concreción/abstracción que sea necesario. Esto es muy importante en los sistemas de integración de datos biomédicos debido a las diferencias de granularidad que existen entre los mismos tipos de datos, pero en diferentes tipos de bases de datos.

Por otro lado, este tipo de esquemas es necesario para escenarios que evolucionan constantemente y que añaden nuevas bases de datos, por lo tanto tienen que modificar su modelo global. En un sistema *bottom-up*, siendo ya extremadamente complicado desarrollar un esquema global completo, rediseñarlo cada vez que se modifican los requisitos de la aplicación es completamente inabordable.

6.6 Tipo de integración semántica

Durante el proceso de integración se pueden realizar varios tipos de integración: agregación o colección, fusión, abstracción y suplementación. El uso de cada una de ellas depende del tipo de problema y de la diferencia entre los datos que se pretenden integrar.

El uso de la agregación por si sola está desaconsejado, debido a que al no realizar ningún tipo de comprobación es muy posible que se recuperen los mismos registros de distintas fuentes de datos y se le presenten al usuario como registros distintos.

La fusión se utiliza en entornos donde es necesario recuperar datos semánticamente similares de distintas bases de datos, pero comprobando si los datos recuperados son registros únicos. Para realizar esta tarea se pueden utilizar vocabularios controlados u ontologías siempre que los esquemas estén correctamente anotados. En muchas tareas de integración simple, este proceso es suficiente.

La abstracción es especialmente común cuando se trabaja con datos biomédicos, debido a que cada base de datos puede trabajar con un nivel distinto de granularidad. El uso de relaciones ontológicas permite implementar este tipo de integración.

En la suplementación se complementan los datos de una fuente con datos de otra. Es comúnmente conocida como integración horizontal. La complejidad de los datos biomédicos lleva a que dependiendo del tipo de tarea de integración que se esté tratando sea necesario utilizar este tipo de integración. Una vez más, es necesario un mecanismo que permita relacionar semánticamente conceptos para poder establecer las relaciones de complementariedad, por lo que generalmente se realiza utilizando ontologías.

6.7 Procesado, planificación y ejecución de consultas

Esta fase sólo se aplica en modelos federados. Con respecto a la planificación de las consultas, ésta debe de estar basada en una descripción de los esquemas de las bases de datos originales con respecto al esquema global, que permita expresar una correspondencia. Genéricamente estas correspondencias se definen con un Lenguaje de Especificación de Correspondencias. Existen principalmente tres aproximaciones, Global como vista (Global as View, GaV), Local como Vista (Local as View, LaV) y Ambos como vista (Both as View, BaV).

Como se vio en el capítulo II, el modelo GaV expresa las relaciones del esquema global como vistas sobre las fuentes, es decir, el mediador se convierte en un conjunto de vistas virtuales sobre las fuentes, mientras que el esquema LaV, expresa las relaciones de las fuentes como vistas sobre las relaciones del mediador.

El modelo GaV tiene el problema de no ser modular, ya que el añadir nuevas fuentes de datos modifican el esquema del mediador, mientras que en el modelo LaV, el añadir nuevas fuentes de datos es sencillo y sólo hay que modificar o añadir una serie de relaciones

En el modelo GaV puede darse el caso de que exista lo que se denomina una "mediación con pérdida", debida a que al integrar fuentes inconsistentes entre ellas, con atributos que no puedan interrelacionarse, se pierde una parte de la integración. En el modelo LaV no puede darse este caso, ya que la aproximación es a la inversa.

Sin embargo, el modelo GaV tiene ventajas con respecto al LaV, como puede ser la programación, que en el LaV es mucho más complejo, y el uso de esquemas mediadores anidados.

El modelo LaV oculta el esquema global a través de las vistas que representan las fuentes, y estas pueden presentar únicamente información parcial sobre los datos. Ya que el mapping asocia cada fuente a una vista sobre el esquema global, no es inmediato inferir cómo se usan las fuentes para responder consultas formuladas sobre el esquema global.

Por lo tanto, el modelo GaV parece ser mejor cuando se presente un sistema de integración en el que las fuentes sean limitadas, estables y conocidas, mientras que el modelo LaV está indicado para muchas fuentes y la posibilidad de que estas cambien o desaparezcan y aparezcan otras nuevas.

6.8 Métrica y costes

Basándose en los apartados anteriores, se describirán dos métricas que permitirán evaluar el coste en tiempo y recursos del sistema de integración que se está diseñando. Dichas métricas permitirán evaluar de una manera objetiva las principales decisiones de diseño del sistema de integración.

Estas dos métricas corresponden a dos dimensiones en la evaluación del problema, la primera de ellas medirá la importancia de cierto factor de decisión clave con respecto a un escenario, permitiendo evaluar el alcance de una u otra decisión. La segunda de ellas se refiere al coste del desarrollo de cada uno de estos factores, independientemente de la importancia que tengan. La combinación de estas dos medidas permite obtener una estimación del coste/beneficio de cada decisión.

Se identifican como factores de decisión la elección de arquitectura, del tipo de integración, de estrategia de integración, de tipo de integración semántica, y en el caso de ser un sistema federado, la estrategia de consultas.

Además de estas decisiones, es importante también la selección de las fuentes, sin embargo, estas decisiones son tan dependientes del problema y del tipo de fuente necesaria, que no es posible definir una métrica.

Uno de los aspectos más importantes de cara al diseño de integración son los costes relativos al número de fuentes que se desea integrar y los costes de evolución del sistema, es decir, lo que costará añadir nuevas fuentes de datos una vez que el sistema haya sido implementado.

Por lo tanto, para definir formalmente la métrica se usaran tres valores, dos que corresponderán a las dos dimensiones explicadas anteriormente y una medida combinada de ellas:

- Importancia: Define la importancia de cada uno de los factores de decisión con respecto al escenario de integración que se está planteando.
- Valor: Define el coste de determinada decisión con respecto a los factores de decisión en un escenario de integración concreto. Estos aspectos son principalmente dependientes de las ventajas y desventajas de un factor de decisión con respecto a otra y tienen poca dependencia del escenario.
- Impacto: Es una medida combinada en función de la importancia del factor de decisión y el coste de tomar cierto valor, que mide el impacto

que cierta decisión tendrá en la implementación del sistema de integración.

6.8.1 Factores de decisión

Para cada una de las decisiones que se van a tomar, es necesario definir una serie de factores que condicionarán la decisión.

- Arquitectura: Para la decisión de la arquitectura se tendrá en cuenta:
  - Inversión inicial: Costes en equipos y desarrollo necesarios en un principio.
  - Actualización: Necesidad de tener siempre la última versión de los datos.
  - Escalado de los datos: Necesidad de manejar grandes cantidades de datos.
  - Rendimiento: Rendimiento en procesado y recuperación de información.
  - Disponibilidad: Necesidad de tener una alta disponibilidad de la información, independientemente de la disponibilidad de las fuentes.
  - Control de datos: Importancia de poseer el control de los datos.
  - Evolución del modelo: Necesidad de que el modelo pueda cambiar a lo largo del tiempo, ya sea añadiendo nuevos tipos de datos o modificando los tipos o las relaciones existentes.
- Tipo de integración:
  - Usuario no experto: Importancia de que el sistema pueda ser utilizado por un usuario no experto que no tenga que responsabilizarse de ninguna parte del proceso de integración.
  - Acceso a fuentes: Necesidad de tener, desde la vista del sistema, acceso directo a las fuentes de datos originales.
- Estrategia de integración:
  - Número de bases de datos: Necesidad de un número importante de bases de datos.

- Acceso escritura: Necesidad de poder disponer, desde la vista, de acceso de escritura a las fuentes.
- Evolución del modelo: Necesidad de que el modelo pueda cambiar a lo largo del tiempo, ya sea añadiendo nuevos tipos de datos o modificando los tipos o las relaciones existentes.

- Integración semántica:
  - Registros repetidos: Necesidad de que no aparezcan en los resultados de integración registros repetidos.
  - Granularidad: Necesidad de que el sistema sea sensible a la granularidad de los datos.
  - Integración horizontal: Necesidad de que se realice integración horizontal de los datos.
- Planificación de consultas:
  - Evolución del modelo: Necesidad de que el modelo pueda cambiar a lo largo del tiempo, ya sea añadiendo nuevos tipos de datos o modificando los tipos o las relaciones existentes.
  - Facilidad de implementación: Necesidad de que la implementación del modelo sea sencilla.

6.8.2 Costes por fuente

Como se comentó anteriormente, un aspecto fundamental es evaluar los costes por fuente de integración que se planifique en un principio, así como los costes de la evolución del sistema en función de costes por fuente añadida una vez ya se ha implementado el sistema final.

Para cada una de las decisiones clave se deberá analizar el coste por fuente que se añada en diseño y el coste por fuente que se añada posteriormente, utilizando los factores descritos de importancia, valor e impacto.

6.8.3 Aplicación de la métrica

El primer paso para aplicar la métrica es definir aquellos factores de decisión que afectan al desarrollo del sistema de integración, que pueden ser los que se

proponen en este marco metodológico u otros factores que sean específicos de cierto escenario de integración.

En un segundo paso se debe indicar, para cada uno de los factores de decisión, la importancia que este tiene en función del escenario. Para cada uno de estos factores se asigna un valor de importancia cualitativo, del tipo “alto” o “bajo”. Por ejemplo, la importancia de la actualización de los datos en un sistema que requiera un solo acceso a los datos es baja, mientras que si es necesaria la última versión de estos datos es alta.

En un tercer paso, independientemente de la importancia que tenga cada factor de decisión, se asigna un valor cuantitativo que indique el peso de las diferentes opciones que existan para cada factor de decisión. Por ejemplo, para cierto escenario, el peso del coste de la inversión inicial puede ser medio para una arquitectura federada pero alta para la arquitectura warehouse.

Finalmente, se calcula el impacto de cada una de las decisiones teniendo en cuenta la importancia de cada decisión, el valor de esta, teniendo como resultado un conjunto de valores positivos, negativos o medios para cada uno de los factores de decisión. La decisión más aconsejable es aquella que tenga mayor número de puntos de impacto positivo y menor número de puntos de impacto negativo.

6.9 Aspectos tecnológicos

Los aspectos tecnológicos dependen fundamentalmente del tipo de implementación que se decida utilizar en el desarrollo del sistema de integración. La elección de un determinado *framework* de integración o de una determinada tecnología viene determinada por los pasos vistos anteriormente.

Sin embargo, existen aspectos tecnológicos comunes que es necesario tener en cuenta en todo caso en cualquier sistema de integración de datos biomédicos, como son aquellos referidos a la seguridad y privacidad de los datos biomédicos que se tratan.

6.9.1 Seguridad y privacidad de los datos

Debido a la importancia de los datos sensibles que puede contener o gestionar un sistema de integración de datos biomédicos, es necesario adoptar una serie de medidas para poder reaccionar contra algún tipo de intrusión. Estas medidas dependen de las características del contexto y del tipo de intrusión.

En estos trabajos (Cios and Moore, 2002) (Malin, 2005) se proponen distintas medidas para aumentar la privacidad y la seguridad en diferentes contextos. Para filtrar las peticiones no autorizadas y para que sólo puedan acceder a los datos aquellos usuarios que deban acceder se usan medidas como el control de acceso, la autenticación y el establecer políticas de seguridad (Emil, 1999) (Ahn and Sandhu, 2000) (Sloman and Lupu, 2002).

Cuando se aumentan los niveles de seguridad, la capacidad de compartir los datos decrece. Obviamente, los datos dejan de estar públicamente disponibles, siendo solo accesibles para aquellos que disponen de las credenciales de seguridad adecuadas. Es necesaria una correcta gestión de estos niveles de acceso para poder satisfacer tanto los requisitos de seguridad como las necesidades de compartir los datos.

Por otra parte, la gestión de la privacidad en los datos biomédicos no es tan simple como parece. En las bases de datos genómicas puede que no sea posible anonimizar a un individuo. Y no sólo la gestión de la privacidad de un individuo es compleja, sino que además también es necesario tener en cuenta la privacidad de la familia de ese individuo. Por ejemplo, si un familiar es diagnosticado con la enfermedad de Huntington, un desorden autosómico fatal, entonces hay un mínimo de un 25% de posibilidades de que alguno de sus hijos tenga esa enfermedad, pero si un hijo es diagnosticado de Huntington, existe un 100% de posibilidades de que alguno de los padres la tenga.

El uso de datos biomédicos en investigación es un problema constante. Las soluciones actuales, como el consentimiento informado o la anonimización son en ocasiones insuficientes (Sweeney, 1997) (Kohane, 2000).

6.10 Validación

La tarea de validar alguna forma de metodología es compleja, debido a que no existe ninguna métrica de validación. Una posible solución sería comparar el marco metodológico desarrollado con otras metodologías en el campo de la integración de datos biomédicos o en el campo de la integración de datos. Sin embargo, si bien existen algunos trabajos que proponen modelos de sistemas que se adaptan a ciertos escenarios, como (Zapletal et al., 2010), específica para soluciones *warehouse*, (Hwang et al., 2005), específica para la fase de agregación de datos o (Reddy et al., 1994) para la integración de esquemas; no se ha encontrado ninguna otra metodología de integración de datos biomédicos o genérica con la que comparar el marco metodológico presentada.

Debido a esto es necesario comprobar la validez del marco metodológico mostrando su resultado para generar sistemas de integración. Para ello es necesario aplicarlo a escenarios posibles de integración biomédica. Sin embargo, no es posible aplicarla a todos los escenarios debido a la diversidad de éstos.

Por ello se ha intentado acotar los escenarios de aplicación a tres líneas complementarias que requieren de la integración de datos en diversos niveles en salud, necesarias para la medina personalizada. Éstas son:

- Apoyo de toma de decisiones en el punto de atención al paciente: En las últimas décadas la práctica médica ha incorporado a su rutina el uso de conocimiento científico. El gran incremento de conocimiento científico y de innovación tecnológica requiere el desarrollo de soluciones que permitan el uso de estas grandes cantidades de información en el proceso de toma de decisiones clínicas. En este campo, la informática biomédica ha desarrollado nuevos métodos y estándares para la integración y análisis de datos clínicos y moleculares. Concretamente, el desarrollo de nuevas aproximaciones de sistemas de ayuda al diagnóstico en el punto de atención al paciente (*Point of Care*).
- Asociación gen-enfermedad: El estudio de las causas moleculares de una determinada enfermedad y las variaciones genéticas de un

individuo permiten profundizar en la medicina personalizada desarrollando soluciones seguras y más eficientes para prevención, diagnóstico y tratamiento. La comunidad científica necesita recursos cada vez más avanzados, acceso a información genómica comparativa y predicción de efectos de mutaciones individuales (SNPs) en las rutas metabólicas y macromoléculas complejas con la consecuente implicación en las enfermedades asociadas.

- Farma-informática: El descubrimiento y desarrollo de nuevos fármacos es un área con gran importancia para la salud humana, y al mismo tiempo es un área de un gran impacto socio-económico. La investigación y desarrollo de fármacos genera grandes cantidades de información que requieren sofisticadas herramientas computacionales para gestionar y analizar todos estos datos. Los laboratorios farmacéuticos fueron pioneros en identificar la necesidad y utilidad del uso de herramientas informáticas para la gestión y explotación de los datos generados en investigación pre-clínica y clínica.

7 Resultados

Hasta ahora se ha descrito un marco metodológico que permite tener en cuenta los aspectos más importantes a la hora de diseñar un sistema de integración. Como se ha especificado en el apartado 6.10 es necesario, para probar el funcionamiento del marco metodológico, aplicarlo a una serie de escenarios representativos de tres líneas de aplicación de integración de datos biomédicos para medicina personalizada.

A lo largo de este capítulo se describirán tres escenarios reales de sistemas de información que requieren la integración de datos biomédicos heterogéneos, correspondientes a cada una de las líneas, la aplicación del marco metodológico descrito en el capítulo anterior y los resultados obtenidos.

7.1 Sistema de visualización y anotación de historia clínica electrónica

7.1.1 Presentación del caso

En las últimas décadas, la práctica médica busca una mayor incorporación a su rutina de trabajo de conocimientos que se ajusten a las necesidades clínicas. El objetivo es asegurar que los pacientes reciben una mejor atención médica, más rápida y más personalizada. Para ello, es muy importante para el equipo médico poder acceder a la información de manera sencilla y que ésta sea fácil de entender en el momento en que se necesite. Por lo tanto, esta información debe estar accesible en el punto de atención al paciente (*Point of Care*, POC). Los sistemas POC ofrecen diferentes servicios a los clínicos, incluyendo sistemas de ayuda a la toma de decisión o pruebas diagnósticas, utilizando fuentes de información de manera autónoma.

Pero para conseguir poder presentar toda esta información al clínico de manera apropiada, es necesario resolver una serie de problemas. En primer lugar debe tenerse en cuenta que la cantidad de datos biomédicos potencialmente útiles para el clínico disponibles en internet, ha aumentado dramáticamente. Por otro lado, los progresos en las "-omics" (genómica, proteómica, variómica, etc.) llevan a un mejor conocimiento de las enfermedades con algún tipo de base genética. El uso de estas nuevas tecnologías biomédicas implica el poder acceder, gestionar y procesar grandes

cantidades de datos almacenados en distintos tipos de bases de datos y sistemas de información que están normalmente disponibles públicamente para la comunidad científica. Por otro lado y no menos importante en la ayuda al clínico, están las bases de datos que este posee en su entorno de trabajo y a las que tiene que conectarse para extraer los datos que desea obtener.

Por lo tanto, el equipo médico, que es quien toma las decisiones, a menudo necesita información proveniente de distintas fuentes, pero que no puede recuperar y gestionar adecuadamente debido a las dificultades existentes cuanto intenta acceder a distintos sistemas. Por esta razón es necesario el desarrollo de métodos y modelos que permitan, no solo la recuperación de información clínica, genómica, etc., disponible en fuentes heterogéneas, sino también que proporcionen al usuario métodos uniformes de acceso.

Por otro lado, con el objetivo de facilitar información útil a los médicos, es necesario mostrar un interfaz de usuario que les permita visualizar los resultados integrados sobre el contexto que están examinando. Este interfaz debe ser intuitivo y debe mostrar los resultados apropiados al problema, válidos y aplicables al paciente concreto que se está tratando.

Para conseguir esto es importante un adecuado diseño de la interfaz y el uso de tecnologías que le puedan aportar todavía más valor para incrementar la interactividad, velocidad y usabilidad.

Finalmente, es muy común que los clínicos necesiten comunicarse con otros médicos para debatir alternativas entre determinados casos clínicos, comparar opiniones, etc. En situaciones como esta, el uso de un sistema de integración de datos sumado al potencial de las tecnologías 2.0 puede resultar muy beneficioso, al aprovechar la inteligencia colectiva.

El objetivo de este caso de estudio es el desarrollo de un visor de historia clínica electrónica que automáticamente anote los datos del paciente con información obtenida a partir de fuentes de datos heterogéneas. Esto permitirá que el clínico

pueda tener toda la información disponible sobre el paciente para una más rápida toma de decisiones.

7.1.2 Análisis

7.1.2.1 Análisis de requisitos

El objetivo de esta herramienta es desarrollar un visor de historia clínica, que permita visualizar y editar el contenido de las bases de datos de historia clínica electrónica de un hospital.

Para poder proporcionar unos resultados que abarquen completamente la historia del paciente que se quiere anotar, el sistema deberá ampliar las búsquedas, detectando sinónimos, traducciones o conceptos. Esto es importante, ya que los datos que aparecen en la historia pueden referirse a un concepto, pero los datos recuperados pueden utilizar otra palabra sinónima para expresar dicho concepto, o estar en un idioma diferente.

El sistema debe anotar las fuentes de manera rápida, teniendo en cuenta las consultas más frecuentes, almacenando los resultados parciales más comunes. De esta manera, la recuperación de los resultados será inmediata mientras estos no hayan cambiado de la fuente original.

Existen multitud de páginas web o bases de datos que permiten obtener información que es útil para el clínico cuando está tratando al paciente, ya sea en la consulta o en el punto de atención al paciente. Por lo tanto el sistema debe de obtener información de las distintas fuentes disponibles e integrarla para presentársela al clínico sobre la propia historia.

Del mismo modo que el clínico debe aprovechar las fuentes de información heterogéneas para poder tratar lo mejor posible al paciente, también existen otras fuentes que, por medio de la web 2.0, pueden ser accesibles: la inteligencia colectiva de sus compañeros. Por lo tanto el sistema debe proporcionar acceso a herramientas colaborativas de manera que el clínico pueda pedir consejo y ayuda a sus compañeros, como pueden ser una wiki centrada en una enfermedad en concreto, foro o un chat.

La herramienta debe de ser sencilla, de manera que el clínico pueda familiarizarse rápidamente con ella.

Basándose en los requisitos del sistema de integración, se ha evaluado la importancia de los factores de decisión ordenados por decisiones clave.

En el caso de la arquitectura, representada en la Tabla 1, los factores de decisión que mayor importancia tienen son los de actualización, control de datos y evolución del modelo. Teniendo en cuenta que parte de los datos se recuperan de la historia clínica, que presenta una variación constante, la importancia de actualización es alta. Del mismo modo, teniendo en cuenta la sensibilidad de los valores de historia clínica, la importancia del control de los datos es alta. En un sistema de anotación como el presentado en este caso de uso, pueden incorporarse en cualquier momento nuevas fuentes de datos y el sistema puede variar su modelo con frecuencia, por lo que la importancia de la evolución del modelo es alta. La importancia del resto de los factores es media. El coste de inversión inicial no es muy importante con respecto a la inversión en el sistema de historia clínica. Por otro lado, el sistema no manejará grandes cantidades de datos que incrementen la importancia del rendimiento del sistema y escalado de los datos. Las fuentes de datos externas necesarias son bien conocidas y presentan una disponibilidad alta, por otro lado, pueden ser sustituidas unas por otras, por lo que la disponibilidad no es un factor de importancia alta. La importancia del coste por fuente o coste por fuente añadida tampoco es alta debido a la simplicidad del modelo.

Tabla 1: Valores de importancia de los factores de decisión en arquitectura escenario 1

Arquitectura	
	Importancia
Inversión inicial	Media
Actualización	Alta
Escalado datos	Media
Rendimiento	Media
Disponibilidad	Media
Control datos	Alta
Evolución modelo	Alta
Coste por fuente	Media
Coste por fuente añadida	Media

En el caso del modelo de integración, que se resume en la Tabla 2, debido a que es una herramienta desarrollada para el uso de clínicos, es vital que el sistema sea usable por un usuario no experto. Por otro lado, no existe una necesidad justificada de tener acceso directo a las fuentes de datos.

Tabla 2: Valores de importancia de los factores de decisión en modelo de integración escenario 1

Integración	
	Importancia
Usuario no experto	Alta
Acceso fuentes	Baja

La importancia de los factores de decisión de la estrategia de integración se resume en la Tabla 2. Tienen una importancia alta el poder disponer de un número alto de bases de datos y la posibilidad de evolución del modelo de datos que se mencionó anteriormente, mientras que, como el sistema es de consulta, el acceso a escritura de las fuentes no tiene importancia.

Tabla 3: Valores de importancia de los factores de decisión en estrategia de integración escenario 1

Estrategia	
	Importancia
Nº Bases datos	Alta
Acceso escritura	Baja
Evolución Modelo	Alta

En cuanto a la integración semántica (Tabla 4) es importante evitar los registros repetidos y tener en cuenta la granularidad de los datos a la hora de anotar la historia clínica. Por otro lado, no se hace necesaria la integración horizontal, debido a que los datos a recuperar son siempre del mismo nivel.

Tabla 4: Valores de importancia de los factores de decisión en tipo de integración semántica escenario 1

Integración semántica	
	Importancia
Registros repetidos	Alta
Granularidad	Alta
Integración horizontal	Baja

En cuanto a la gestión de las consultas (Tabla 5), como se vio anteriormente, la evolución del modelo tiene una importancia alta, y la facilidad de implementación, teniendo en cuenta especialmente la ausencia de integración horizontal una importancia media.

Tabla 5: Valores de importancia de los factores de decisión en gestión de consultas escenario 1

Consultas	
	Importancia
Evolución modelo	Alta
Facilidad implementación	Media

7.1.2.2 Análisis de casos de uso

En este escenario aparece un actor principal, que es el clínico que desencadena las búsquedas al consultar el sistema.

- Acceso a la historia clínica: Un clínico autenticado podrá visualizar y editar las historias clínicas de los pacientes. Además de esto, una vez visualizada la historia, el sistema automáticamente realizará una búsqueda tanto en las herramientas colaborativas en las que el clínico esté autorizado, así como en diversas fuentes de datos como pueden ser páginas web, bases de datos biomédicas o servicios web. La información encontrada relativa a los términos de la historia clínica será resaltada con un hiperenlace con objetivo de no entorpecer la consulta. Si el médico quiere ampliar la información sobre alguno de los términos resaltados únicamente tendrá que pinchar en él y accederá a la información obtenida.
- Funcionalidades Web 2.0:
    - Incorporación de un foro: El usuario podrá crear hilos, categorías, publicar entradas en un foro, que serán accesibles en varios niveles, desde públicos hasta limitados a los clínicos del mismo servicio de un hospital.

- Incorporación de una wiki: El clínico podrá crear o modificar wikis públicas o en las que esté autorizado. Dichas wikis pueden ser compartidas por un servicio de un hospital o por especialistas de distintos hospitales.
- Acceso al chat: El sistema incorpora un chat que permite que un usuario se comunique con otros especialistas del servicio o de otros hospitales.
- Gestión del portal: El usuario podrá crear portales personalizados añadiendo *widgets*. Dichos portales estarán asociados a la cuenta del clínico y darán acceso directo a la aplicación de historia y a las funcionalidades 2.0.

7.1.2.3 Análisis de datos a integrar

A partir de los requisitos que se han recogido en los apartados anteriores, se ha buscado una serie de fuentes de datos que puedan ser interesantes para el clínico con el objetivo de anotar la historia, tanto las propias bases de datos que contengan la historia clínica como bases de datos relativas a enfermedades, a fármacos, etc.

7.1.2.3.1 Contenido de los datos

Para analizar los datos que se van a mostrar en la aplicación de historia clínica electrónica, en primer lugar se hará una división entre los datos base y los datos externos. Los datos base serán aquellos que contienen los datos propios de la historia clínica electrónica, es decir, los datos clínicos de un paciente en concreto. Por otro lado estarán los datos externos que son los que se utilizarán para aportar información que servirá para completar la historia clínica.

Dentro del primer grupo estarán las bases de datos de historia clínica electrónica que tenga el hospital, pero también otro tipo de datos que no tienen que estar dentro de la historia clínica, como las imágenes médicas del paciente que pueden estar en sistemas de información radiológica.

Dentro del segundo grupo podrían estar todo el conjunto de fuentes de datos, debido a que, con los avances de la medicina personalizada, el clínico cada vez tendrá que acceder a datos a más relacionados con las ómicas, y no sólo los datos puramente

clínicos. Es por eso que dependiendo del alcance de la aplicación de historia clínica que se desee realizar habrá que incluir un mayor número de fuentes de datos. Una aplicación destinada a un servicio de atención primaria no requeriría los mismos datos que una aplicación destinada a un servicio de diagnóstico genético.

Teniendo en cuenta los factores mencionados anteriormente, será necesario el acceso a fuentes de datos de literatura médica, de enfermedades, de asociación entre enfermedades y genes o variaciones y de fármacos.

Como ejemplos de bases de datos de literatura, probablemente las más importantes son Medline o herramientas más complejas que incluyan esta como pueden ser Pubmed del NCBI o CiteXplore del EBI.

Como ejemplos de bases de datos de enfermedades, MedlinePlus como fuente genérica, GIDEON para enfermedades infecciosas o TRDTargets para enfermedades tropicales. Para enfermedades de base genética OMIM, PharmGKB, Genes and Diseases o ClinVar del NCBI.

Finalmente, sobre fármacos se puede considerar desde fuentes más genéricas como Vademecum, Boletín de Fármacos o Drugs@FDA hasta fuentes más científicas como la anteriormente mencionada PharmGKB.

7.1.2.3.2 Interfaz

En este punto se analizan los interfaces o modos de acceso de las diferentes fuentes de datos seleccionadas.

El primer grupo de fuentes de datos, que está formado por los distintos tipos de acceso a la historia clínica de los pacientes, generalmente utilizan un formato de acceso que está compuesto por una serie de estándares de intercambio de información clínica. Existen distintos tipos de estándares de intercambio de mensajes en HCE, como puede ser el HL7, GEHR/openEHR o CEN EN 13606 (Dolin et al., 2001) (Blobel et al., 2006). Además también es necesario contar con el estándar de intercambio de imágenes médicas DICOM (Bidgood et al., 1997). También es posible que algunas de las bases de datos de historia clínica estén en formato propio, por lo que serán accesibles mediante SQL.

En el segundo grupo de fuentes de datos se pueden encontrar principalmente tres grupos:

- Acceso web: Algunas de las fuentes de datos son accesibles únicamente por medio de su página web, por lo que requieren el desarrollo de un parseador para poder recuperar los datos de manera homogénea.
- Acceso por servicios web: Tanto las bases de datos alojadas en el NCBI como las alojadas en el EBI permiten acceso a través de servicios web.
- Acceso directo a las bases de datos: Algunas de las bases de datos permiten acceso directo a través de servidores de bases de datos, como es el caso del EBI con sus servidores MySQL. Otros como OMIM permiten la descarga directa de la base datos.

7.1.2.3.3 Actualizaciones

La mayoría de los servicios web y bases de datos están en servidores muy fiables como pueden ser los del NCBI o los del EBI, que aseguran su mantenimiento y continua actualización. Los datos de las páginas web también son fiables y están correctamente actualizados, sin embargo, al no proporcionar un formato genérico de acceso a los datos, estos deben ser parseados del HTML lo que origina un problema, dado que si el formato en el que se presentan los datos se modifica, es necesario modificar el parseador para que recupere los datos correctamente.

7.1.2.3.4 Modelo de datos y anotación semántica

De las fuentes de datos analizadas, aquellas provenientes de páginas webs no presentan ningún tipo de modelo de datos, siendo este definido dentro del parseador que recoge los resultados.

Las fuentes de datos que son accedidas por medio de servicios web ocultan su modelo de datos detrás de este interfaz.

Las fuentes de datos que exponen su propia base de datos como las procedentes del EBI presentan una estructura relacional, mientras que otras como el OMIM tienen una estructura en formato de ficheros de texto.

Las fuentes de historia clínica electrónica presentan su propio modelo a través de los estándares como HL7 o DICOM.

En cuanto a la anotación semántica, la gran mayoría de las webs utilizadas como fuentes de datos no presentan ningún tipo de anotación semántica, sin embargo las fuentes de datos procedentes del EBI o NCBI sí proporcionan este tipo de anotación, del mismo modo, los estándares de historia clínica o de imagen también presentan cierto tipo de anotación, que va en aumento en sus diferentes versiones.

7.1.2.3.5 Clasificación Busse

Según el modelo de Busse, el análisis de las fuentes de datos para este escenario sería el siguiente:

- Autonomía: Las fuentes de datos provenientes de organismos o laboratorios externos pertenecen a estos laboratorios, por lo tanto tienen una total autonomía de diseño y ejecución. Las bases de datos donde se almacena la historia clínica pueden tener o no autonomía de diseño dependiendo de su uso de los estándares. No tienen autonomía de ejecución, sin embargo dependiendo de la organización puede no tener acceso a ella. Normalmente las bases de datos de historia clínica tienen la autonomía de comunicación restringida al uso de estándares de paso de mensajes. En los casos en los que la base de datos puede ser almacenada y accedida en local, como es el caso del OMIM, existe una total autonomía de diseño y ejecución.
- Heterogeneidad:
    - Sintáctica tecnológica y de interfaz: Existen fuentes de datos agrupadas en páginas web sin formato, servicios web, bases de datos, historia clínica electrónica. Algunas de ellas pueden ser utilizadas en local por lo que puede ser accedida con la tecnología que se quiera.
    - Sintáctica de consulta: Las fuentes de datos que son accedidas por medio de páginas web no permiten acceder al modelo, las que son accedidas por medio de servicios web permiten acceder

sólo a una parte del modelo, que suele ser suficiente para este escenario. Las que presentan interfaz relacional pueden ser accedidas completamente, así como las que puedan ser descargadas en local.

  - De modelo y lógica: La mayoría de las bases de datos presentan un modelo de datos relacional, los servicios web se presentan sin modelo de datos. Las páginas web no presentan ningún modelo. Algunas bases de datos y servicios web (principalmente los del EBI y NCBI) pueden presentar heterogeneidad lógica semántica.

- Distribución: Las fuentes de datos están distribuidas entre los diferentes portales webs, los servicios web, principalmente del NCBI y EBI y las bases de datos del EBI. Las fuentes de datos de historia clínica suelen estar en el hospital o clínica y las bases de datos que puedan ser descargadas localmente pueden estar localizadas en local.

Una vez analizados los distintos tipos de fuentes de datos que pueden aportar información al sistema, y teniendo en cuenta el amplio rango de fuentes de datos que se podrían incluir para anotar la historia clínica del paciente, se han tenido en cuenta simplemente algunas fuentes de datos heterogéneas para la implementación de este escenario. De esta manera se han escogido la página de "Boletín de Fármacos" y "MedlinePlus" como representantes de fuentes de datos web proporcionando información de fármacos y enfermedades. OMIM de NCBI y Citexplore del EBI como representantes de servicios web proporcionando información de enfermedades de origen genético y literatura. Finalmente se optó por simular en una base de datos relacional la base de datos de historia clínica electrónica.

7.1.2.4 Selección de arquitectura

El escenario actual describe un sistema de anotación de historia clínica, con lo cual no es posible un sistema de navegación por link como aproximación, ya que se anota cada una de las entradas de la historia clínica en paralelo. Pese a que la apariencia final del sistema, puede dar la impresión de que se trata un sistema de navegación por links, es necesario algún tipo de estrategia de integración más compleja.

Se han valorado los factores clave tanto para la arquitectura warehouse como para modelo federado, con los resultados que aparecen en la Tabla 6. En este caso el valor que sigue de la arquitectura federada y warehouse se corresponde con los valores presentados en los fundamentos. En lo que se refiere al coste por fuente, el coste en tiempo de desarrollo es alto para el modelo federado pero bajo para el warehouse, mientras que el coste de incluir nuevas fuentes en un modelo warehouse ya desarrollado es sensiblemente más alto que en el federado.

Tabla 6: Valoración arquitectura escenario 1

Arquitectura		**Federado**	**Warehouse**
	Importancia	Valor	Valor
Inversión inicial	Media	Medio	Alto
Actualización	Alta	Alto	Medio
Escalado datos	Media	Bajo	Muy Alto
Rendimiento	Media	Medio	Alto
Disponibilidad	Media	Bajo	Alto
Control datos	Alta	Alto	Alto
Evolución modelo	Alta	Alto	Bajo
Coste por fuente	Media	Alto	Bajo
Coste por fuente añadida	Media	Medio	Alto

7.1.2.5 Tipo de modelo de integración

En este escenario el usuario no hará más que consultar la historia clínica del paciente que, automáticamente, disparará la ejecución de una serie de consultas que pueden ser más o menos complejas, pero que serán siempre del mismo modo. La responsabilidad de integración nunca deberá recaer sobre el usuario final. Este acceso al sistema de un usuario no experto se da en el modelo de integración cerrado, por lo que tiene un valor alto y el modelo abierto un valor bajo. El coste por fuente es más alto en la integración cerrada que en la abierta, pero en el caso de querer complicar más el sistema, este se complicará añadiendo más fuentes sobre el mismo modelo de integración que definirá el modelo de datos de la historia clínica, por lo que los costes serán medios. La valoración de cada uno de estos factores se detalla en la Tabla 7.

Tabla 7: Valoración modelo integración escenario 1

Integración		Abierta	Cerrada
	Importancia	Valor	Valor
Usuario no experto	Alta	Bajo	Alto
Acceso fuentes	Baja	Alto	Bajo
Coste por fuente	Media	Bajo	Alto
Coste por fuente añadida	Media	Medio	Medio

7.1.2.6 Estrategia de integración

La valoración de las aproximaciones para estrategia de integración se muestra en la Tabla 8. Cumplen con las características definidas en el capítulo de fundamentos sobre estrategias de integración. Debido a la complejidad de tratar con nuevas bases de datos, el coste de incluir nuevas fuentes de datos, tanto en desarrollo, como posteriormente, es mucho más alto en *bottom-up* que en *top-down.*

Tabla 8 Valoración estrategia escenario 1

Estrategia		Top-down	Bottom-up
	Importancia	Valor	Valor
Nº Bases datos	Alta	Alto	Bajo
Acceso escritura	Baja	Bajo	Alto
Evolución Modelo	Alta	Alto	Bajo
Coste por fuente	Media	Medio	Alto
Coste por fuente añadida	Media	Medio	Alto

En este punto puede definirse ya el modelo de integración que, como se comentó anteriormente, viene definido por el modelo de historia clínica que debe ser anotado. La base de datos de ejemplo creada ad-hoc para la simulación de este escenario no es tan amplia como podría ser una historia clínica completa, pero se considera que el aumento de complejidad del modelo de historia clínica y el aumento de complejidad del modelo de integración sería lineal.

Las entidades del modelo de historia que se quieren anotar son las siguientes, todas ellos puntos de entrada del sistema:

- Síntomas que presenta el paciente.
- Enfermedades que han sido diagnosticadas al paciente.
- Medicación que se le está administrando.

7.1.2.7 Tipo de integración semántica

Los valores de integración cumplen generalmente con los valores estándar definidos en el capítulo de fundamentos.

La abstracción tiene un coste bastante más alto por fuente debido a que es capaz de tratar la granularidad de los datos, aunque en este caso de estudio no es necesaria.

La suplementación tiene un coste de desarrollo mucho más alto, especialmente en este caso de estudios debido a la variabilidad de las fuentes, al implementar todas las funciones semánticas y especialmente la integración horizontal.

En la Tabla 9 se muestra un resumen de los valores para cada técnica de integración semántica.

Tabla 9 Valoración técnica de integración semántica escenario 1

Integración semántica		**Agregación**	**Fusión**	**Abstracción**	**Suplementación**
	Importancia	Valor	Valor	Valor	Valor
Registros repetidos	Alta	Bajo	Alto	Alto	Alto
Granularidad	Alta	Bajo	Bajo	Medio	Alto
Integración horizontal	Baja	Bajo	Bajo	Bajo	Alto
Coste por fuente	Media	Bajo	Medio	Alto	Muy Alto
Coste por fuente añadida	Media	Bajo	Medio	Alto	Muy Alto

7.1.2.8 Procesado, planificación y ejecución de consultas

En la Tabla 10 se muestran los valores de la estrategia de gestión de consultas, que corresponden con los definidos en los fundamentos. El coste por fuente es mayor en la fase de desarrollo en el modelo LaV, mientras que en la etapa posterior al desarrollo es mucho más costoso en el modelo GaV.

Tabla 10: Valoración gestión de consultas escenario 1

Consultas		LaV	GaV
	Importancia	Valor	Valor
Evolución modelo	Alta	Alto	Bajo
Facilidad implementación	Media	Bajo	Alto
Coste por fuente	Media	Alto	Bajo
Coste por fuente añadida	Media	Bajo	Alto

7.1.2.9 Métricas y costes

En este paso se evalúa el impacto de cada decisión teniendo en cuenta tanto la importancia como el valor.

Tabla 11: Impacto arquitectura escenario 1

Arquitectura		Federado		Warehouse	
	Importancia	Valor	Impacto	Valor	Impacto
Inversión inicial	Media	Medio	Medio	Alto	Medio
Actualización	Alta	Alto	Positivo	Medio	Medio
Escalado datos	Media	Bajo	Medio	Muy Alto	Positivo
Rendimiento	Media	Medio	Medio	Alto	Medio
Disponibilidad	Media	Bajo	Medio	Alto	Positivo
Control datos	Alta	Alto	Positivo	Alto	Positivo
Evolución modelo	Alta	Alto	Positivo	Bajo	Negativo
Coste por fuente	Media	Alto	Medio	Bajo	Medio
Coste por fuente añadida	Media	Medio	Medio	Alto	Negativo

Se desaconseja la aproximación del tipo *warehouse* principalmente porque los datos que se pretende integrar sobre los datos de la historia clínica son una serie de datos externos procedentes de fuentes heterogéneas. Este tipo de fuentes puede variar a lo largo del tiempo y puede ser interesante tener varios tipos de fuentes distintas dependiendo del fin que tenga la aplicación de visualización de historia clínica. Un *warehouse* requeriría la copia de todos estos datos adaptados en un modelo centralizado y monolítico, lo que restaría mucha flexibilidad al sistema.

La aplicación más natural en este escenario es el modelo federado, ya que el sistema de anotación que se requiere puede considerarse como un buscador complejo, que recoge información de varias fuentes heterogéneas, las homogeniza y las integra sobre la historia clínica de un paciente.

El impacto positivo o medio de factores como inversión inicial, escalado de datos, rendimiento y disponibilidad del *warehouse* es importante, sin embargo, comparado con el impacto negativo del escalado del modelo y de añadir nuevas fuentes implica que el impacto positivo del modelo federado es mayor que el del *warehouse* (Tabla 11). Por ese motivo se recomienda el modelo federado teniendo en cuenta que este sistema puede requerir añadir nuevas fuentes.

Tabla 12: Impacto modelo de integración escenario 1

Integración		**Abierta**		**Cerrada**	
	Importancia	Valor	Impacto	Valor	Impacto
Usuario no experto	Alta	Bajo	Negativo	Alto	Positivo
Acceso fuentes	Baja	Alto	Medio	Bajo	Medio
Coste por fuente	Media	Bajo	Medio	Alto	Medio
Coste por fuente añadida	Media	Medio	Medio	Medio	Medio

En el caso del modelo de integración, la recomendación es la selección de un tipo de integración cerrada, ya que uno de los factores con mayor impacto es el de la creación de un sistema que no descargue la responsabilidad de integración sobre el usuario (Tabla 12).

Tabla 13: Impacto estrategia de integración escenario 1

Estrategia		**Top-down**		**Bottom-up**	
	Importancia	Valor	Impacto	Valor	Impacto
Nº Bases datos	Alta	Alto	Positivo	Bajo	Negativo
Acceso escritura	Baja	Bajo	Medio	Alto	Medio
Evolución Modelo	Alta	Alto	Positivo	Bajo	Negativo
Coste por fuente	Media	Medio	Medio	Alto	Negativo
Coste por fuente añadida	Media	Medio	Medio	Alto	Negativo

Debido a los requisitos del sistema de anotación, el modelo de datos está definido por parte de la historia clínica que se quiera anotar. Esto datos serán los que tendrán que ser recuperados de las fuentes originales y no todos los datos que pertenezcan a las bases de datos de origen. Por otro lado, el impacto de trabajar con un gran número de bases de datos y el coste de evolución del modelo es negativo en el modelo *bottom-up* (Tabla 13). Debido a esto parece claro que la única estrategia posible es *top-down*.

Tabla 14: Impacto integración semántica escenario 1

Integración semántica		**Agregación**		**Fusión**	
	Importancia	Valor	Impacto	Valor	Impacto
Registros repetidos	Alta	Bajo	Negativo	Alto	Positivo
Granularidad	Alta	Bajo	Negativo	Bajo	Negativo
Integración horizontal	Baja	Bajo	Medio	Bajo	Medio
Coste por fuente	Media	Bajo	Medio	Medio	Medio
Coste por fuente añadida	Media	Bajo	Medio	Medio	Medio

Integración semántica		**Abstracción**		**Suplementación**	
	Importancia	Valor	Impacto	Valor	Impacto
Registros repetidos	Alta	Alto	Positivo	Alto	Positivo
Granularidad	Alta	Medio	Medio	Alto	Positivo
Integración horizontal	Baja	Bajo	Medio	Alto	Medio
Coste por fuente	Media	Alto	Medio	Muy Alto	Negativo
Coste por fuente añadida	Media	Alto	Medio	Muy Alto	Negativo

En cuanto al modelo de integración semántica, debido a que el sistema de anotación, que se fija en los requisitos, requiere claramente de un sistema de integración vertical, se descarta la suplementación, por tratarse de un tipo de integración horizontal y por los costes de implementación por fuente (Tabla 14).

Las necesidades de este sistema de anotación indican que no puede utilizarse únicamente el tipo de integración por agregación, ya que puede haber el mismo dato disponible en distintas fuentes y no tiene sentido que se presente la misma información varias veces, también por ese motivo se aconseja un tipo de integración por fusión.

Los costes de una integración por abstracción no justifican su uso, debido además a que los datos que se van a recuperar no son tan complejos como para que puedan aparecer problemas de granularidad.

Tabla 15: Impacto gestión de consultas escenario 1

Consultas		**LaV**		**GaV**	
	Importancia	Valor	Impacto	Valor	Impacto
Evolución modelo	Alta	Alto	Positivo	Bajo	Negativo
Facilidad implementación	Media	Bajo	Medio	Alto	Positivo
Coste por fuente	Media	Alto	Negativo	Bajo	Positivo
Coste por fuente añadida	Media	Bajo	Positivo	Alto	Negativo

Una vez escogido el modelo top-down, en el que las relaciones entre las fuentes de datos y el esquema global se realiza desde el punto de vista del usuario y teniendo en cuenta los requisitos de este escenario, se descarta el modelo de consultas GaV principalmente debido a su coste en función de la evolución del modelo (Tabla 15).

7.1.2.10 Aspectos tecnológicos

7.1.2.10.1 Seguridad y privacidad de los datos

En este caso es de vital importancia la seguridad de los accesos a la historia clínica, ya que representa datos del nivel más alto de seguridad, al menos dentro de la legislación española, porque especifica los datos personales del paciente y todo su historial médico. Es necesario implementar medidas de seguridad y privacidad en cuanto a los accesos por parte de los clínicos y medidas de seguridad en cuanto a las protecciones de las comunicaciones entre el sistema y los repositorios de historia clínica o repositorios de imagen.

En cuanto al resto de los accesos, las bases de datos, páginas web y servicios web son de acceso público, con lo cual no es necesario implementar medidas de seguridad adicionales.

7.1.2.10.2 Framework de integración

Los requisitos de este escenario apuntan al uso de un *framework* de integración federado basado en mediador y *wrappers*. Sin embargo, no parece que sea necesario un framework de integración complejo, por lo que se han utilizado un conjunto de *wrappers/mediator* implementados en el propio laboratorio.

7.1.3 Desarrollo

Teniendo en cuenta el análisis realizado, se utilizará una arquitectura federada basada en el patrón wrapper-mediator para realizar el proceso de integración. El modelo de integración será cerrada y la estrategia de integración será Top-down. En este caso el tipo de integración semántica que se use será la integración por fusión y el modelo de consultas el LaV.

7.1.3.1 Arquitectura

La arquitectura del sistema se dividirá en tres módulos principales, el módulo interfaz, que incluye tanto la aplicación de historia clínica utilizada para mostrar la información al clínico como las herramientas de la Web 2.0. El módulo interfaz envía las consultas al módulo de búsqueda, que expande las búsquedas utilizando la ontología UMLS y un traductor y almacena los resultados frecuentes en una cache. El módulo de búsqueda envía las consultas al módulo de integración, que es donde se aplica el patrón wrapper-mediator para recuperar la información de fuentes heterogéneas.

7.1.3.1.1 Módulo interfaz

El módulo interfaz se compone de varios subsistemas, tal y como se pueden ver en la Figura 22. El subsistema más importante es el que muestra la historia clínica de cada paciente y la completa con la información obtenida de los otros módulos. El interfaz ha sido diseñado para optimizar la usabilidad. Todos los resultados relativos a los datos del paciente han sido integrados en una página web simple.

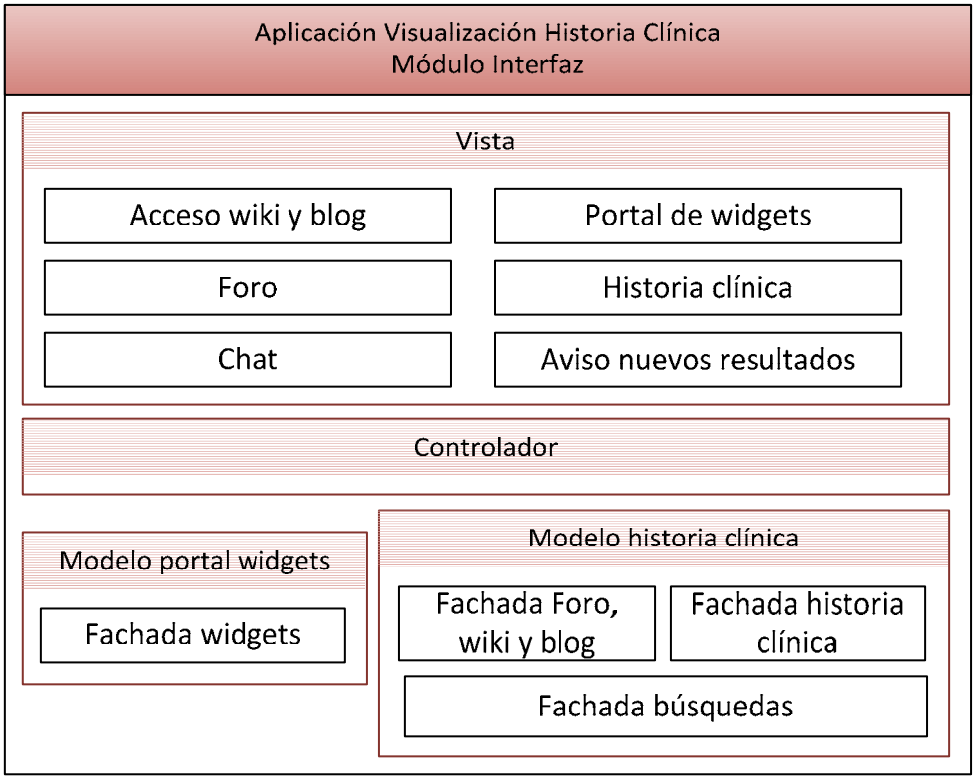


Figura 22: Arquitectura del módulo interfaz del sistema de anotación de historia clínica

Para desarrollar el módulo interfaz se utilizó el patrón arquitectónico *Model-View-Controller*. La capa modelo está compuesta, por un lado, por el subsistema de gestión del portal de *widgets,* que usa el framework Dropthings (dropThings, 2012). Por otro lado están las clases que gestionan el acceso tanto el modelo del foro, el del chat y el de la wiki, como a los datos de historia clínica y a los datos recuperados de las bases de datos externas.

El subsistema de acceso al foro y a la wiki gestiona, además del acceso a los *frameworks* propios del foro (YAF, 2012) y la wiki (XWOSW, 2012), un sistema de búsqueda que permite buscar las apariciones de términos dentro de estas dos herramientas.

El subsistema de acceso a la historia clínica trabaja directamente con la base de datos donde se almacena la historia clínica de los pacientes, mientras que el subsistema de acceso a bases de datos externos consiste en un cliente de servicios web que accede a los módulos externos. Estos servicios web se invocan de manera asíncrona, de manera que el usuario puede solicitar los datos, pero si la respuesta no

es inmediata, el sistema no se queda bloqueado, sino que continúa su ejecución normal.

El controlador se ha utilizado para separar la lógica de consultas de la vista. Para ello se ha utilizado el patrón *page controller*.

La capa vista se ha implementado utilizando páginas aspx, de manera que cuando se muestren las historias clínicas de los pacientes, además de los datos personales del paciente, se muestra información relacionada con estos, encontrada en la wiki, en el blog y en las diversas fuentes de datos externas. Para mejorar la usabilidad del sistema se ha decidido integrar todos los resultados en una sola página. En dicha página también se integran las herramientas web 2.0, para potenciar el intercambio de información entre los profesionales. Por un lado el wiki permite crear, borrar y editar páginas colaborativamente, gestionando los permisos en base a los privilegios y grupos de usuarios. El foro permite la creación de diversas categorías (p.e. Medicina General, Cardiología, Pediatría, etc.) y permite a los usuarios crear nuevos hilos dentro de las categorías y publicar comentarios. El chat permite comunicarse a los diferentes clínicos de una especialidad en tiempo real.

El portal de *widgets* se ha implementado de manera similar a iGoogle, de manera que el clínico pueda configurarlo añadiendo o eliminado *widgets* que permitan automatizar las búsquedas o suscribir a noticias o páginas a las que accedan frecuentemente (lectores RSS, traductores, etc.). El uso de estos *widgets* proporciona escalabilidad y bajo acoplamiento.

Debido a que los datos a los que se accede incorporan información relativa a pacientes, el acceso a la web de la aplicación debe ser completamente seguro. Para garantizar la seguridad se ha implementado la tecnología SSL que encripta la comunicación entre el navegador y el sistema de información de historia clínica.

7.1.3.1.2 Módulo gestor de búsquedas

El módulo gestor de búsquedas es el encargado de recibir las peticiones del interfaz y devolver los resultados que se obtengan. El módulo gestor de búsquedas se

divide en los subsistemas que se muestran en la Figura 23, el repositorio de información, el traductor y la ontología.

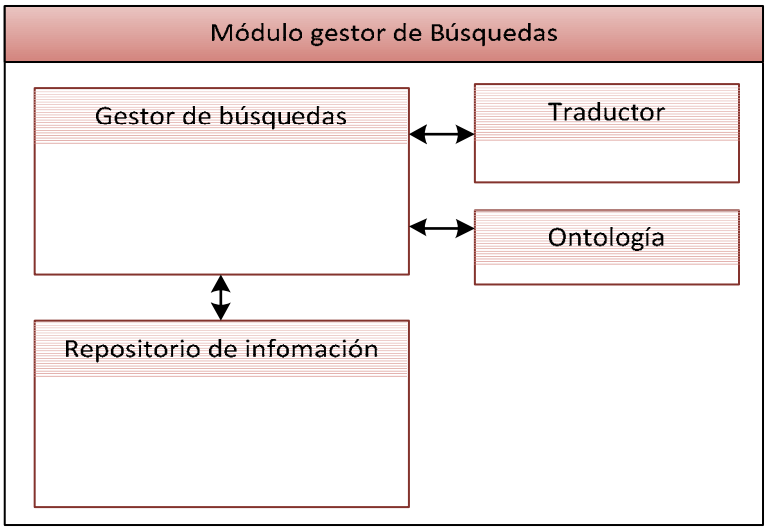


Figura 23: Arquitectura del módulo gestor de búsquedas

Con el objetivo de minimizar el tiempo de recuperación de resultados y teniendo en cuenta que habrá una gran cantidad de consultas que sean similares, se ha implementado un subsistema de cache, denominado repositorio de información. Este sistema de cache funciona de la siguiente manera: una vez que se recibe una petición se consulta al repositorio. Si éste contiene información válida, se devuelve esta información al sistema. En el caso de que el repositorio de información no tenga información válida, en primer lugar que consulta al módulo del traductor, posteriormente a la ontología y finalmente se accede al modulo de integración para obtener la información de fuentes externas. Estos resultados se almacenan en el repositorio de información o se actualizan los actuales de manera que sean válidos. Finalmente se devuelve el resultado obtenido al usuario.

El repositorio de información almacena la información durante un tiempo limitado, pasado ese tiempo la información almacenada se borra o se invalida. Para la implementación de esta cache se ha usado el principio de localización temporal, por el cual si una información ha sido buscada, es posible que estos mismos datos puedan ser usados en un breve periodo de tiempo.

El traductor permite la traducción de las palabras que están siendo buscadas en varios idiomas. La incorporación de este traductor es debido a que tanto las fuentes externas como la ontología utilizan lenguas mayoritarias, en su mayoría el inglés. Para la implementación de este subsistema de traducción se ha elegido la plataforma Open Source Apertium (Forcada et al., 2011).

El subsistema de ontología accede a la ontología UMLS para expandir las búsquedas una vez han salido del traductor. UMLS es la ontología más importante en este campo y supone un estándar en terminología médica. Una vez el termino llega, se expande utilizando la ontología buscando sinónimos, subtipos (por ejemplo, la taquicardia es un tipo de arritmia) y las palabras asociadas tanto en español como en inglés. Además, este subsistema utiliza un módulo que identifica conceptos, de manera que en caso de que exista un error tipográfico en la historia, el algoritmo permite corregirlo (por ejemplo, si en lugar de *cáncer* se ha escrito *cance*).

El módulo gestor de búsqueda presenta un interfaz de servicios web que hace que pueda ser utilizado no sólo por el módulo interfaz, sino por otro tipo de aplicaciones externas.

7.1.3.1.3 Módulo de integración

El módulo de integración (Figura 24) se basa en un enfoque federado, tal y como se decidió en la etapa de análisis. Está compuesto por un subsistema mediador con un modelo *top-down*, que recibe el conjunto de las consultas expandidas, descompone las búsquedas en una serie de consultas y redirige cada una de las consultas a los *wrappers* que contienen ese dato. Una vez recuperados los datos en un formato homogéneo, unifica los resultados y los devuelve al módulo gestor de búsquedas.

Cada *wrapper* es un traductor bidireccional que procesa la consulta del mediador, la transforma al modelo nativo de consulta de la fuente, recupera los resultados y los devuelve al modelo común del mediador.

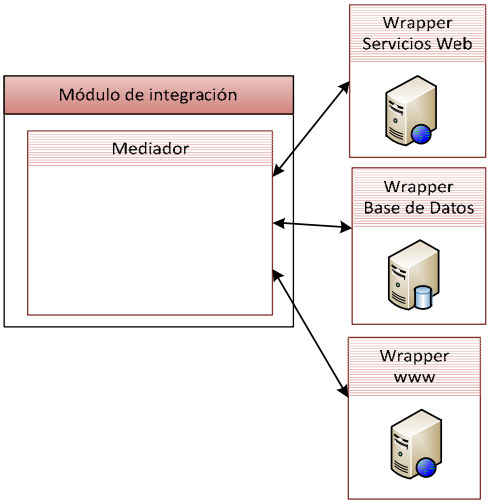


Figura 24: Arquitectura módulo de integración

Se han desarrollado tres *wrappers* para tres tipos de fuentes de datos, bases de datos, servicios web y páginas web. El *wrapper* de base de datos es genérico y se ha desarrollado de manera que el usuario únicamente tenga que definir un fichero de configuración con el nombre de la base de datos, el driver, la información de la base de datos y la consulta o consultas asociadas al modelo genérico. Se ha desarrollado un *wrapper* específico para cada una de los servicios web que se acceden, uno para el OMIM del NCBI y otro para Citexplore del EBI. Finalmente, se desarrolló un wrapper para parsear páginas html, con una instancia para "Boletín de fármacos" y otra para "Medline". Este *wrapper* es estático y se implementó utilizando la versión .Net del *parser* de java HTMLParser, implementado por NetOMatix (NetOMatix, 2012).

Todos los resultados del módulo de integración se intercambian con el módulo gestor de búsquedas utilizando servicios web. Este sistema de intercambio de información permite que el sistema de integración pueda servir de fuente de información a otros sistemas que puedan consumir servicios web.

7.1.4 Pruebas

A continuación se mostrará una prueba de ejecución del sistema. En la Figura 25 se puede ver un ejemplo del portal web, con varios widgets, como los de traducción y lectura de feeds.

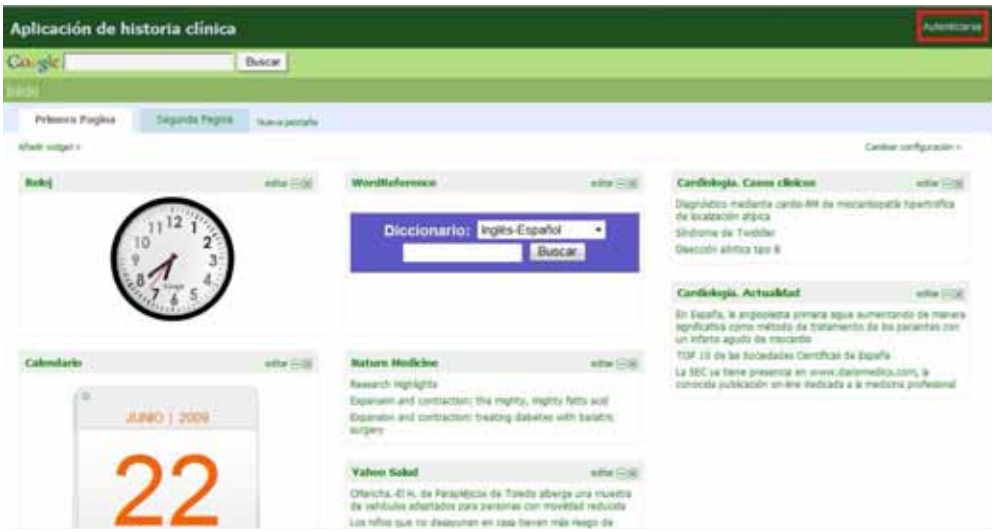


Figura 25: Portal de widgets

Una vez autenticado, se tiene acceso a los distintos servicios de consulta de historia clínica y de acceso a los servicios web 2.0 (Figura 26).

Figura 26: Opciones de usuario autenticado

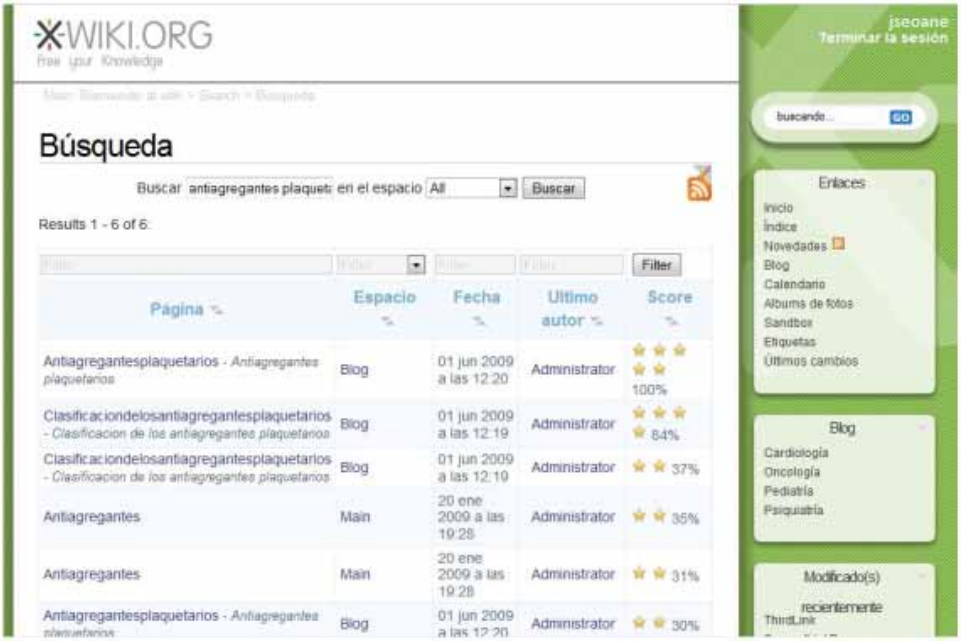
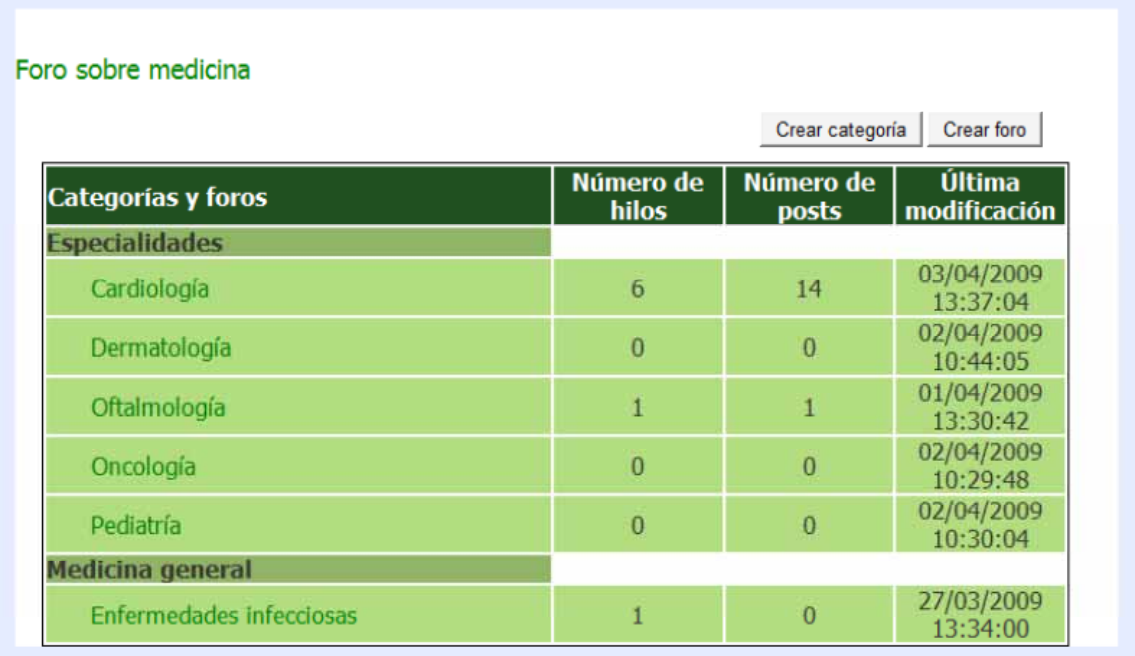


Figura 27: Wiki

La aplicación de wiki permite crear entradas y realizar búsquedas sobre las entradas ya introducidas. En la Figura 27 se muestra una captura de pantalla de la wiki.

Otra de las aplicaciones de la web 2.0 es el foro, en la Figura 28 se muestra una captura del foro, en la categoría de "especialidades", donde se muestran enfermedades médicas como "cardiología", "dermatología", etc.



Categorías y foros	Número de hilos	Número de posts	Última modificación
Especialidades			
Cardiología	6	14	03/04/2009 13:37:04
Dermatología	0	0	02/04/2009 10:44:05
Oftalmología	1	1	01/04/2009 13:30:42
Oncología	0	0	02/04/2009 10:29:48
Pediatría	0	0	02/04/2009 10:30:04
Medicina general			
Enfermedades infecciosas	1	0	27/03/2009 13:34:00

Figura 28: Foro

El prototipo de visión de historias clínicas permite buscar las historias por su identificador, como se muestra en la Figura 29.

Figura 29: Búsqueda por id. de historia clínica

Una vez encontrada la historia, se muestran una serie de datos. Al tratarse de un prototipo, únicamente se muestran el nombre, los síntomas, el diagnóstico y la medicación, un ejemplo de una historia clínica ficticia se muestra en la Figura 30.

Figura 30: Visualización de historia clínica

Una vez comienza la visualización de la historia, automáticamente el sistema genera una serie de consultas de los términos que aparecen en los campos “síntomas”, “diagnóstico” y “medicación”, enviándolos al sistema de gestión de consultas y consultándolos en el foro y en la wiki. Si se encuentran resultados, se resaltan los términos de los que se han encontrado resultados y aparece un hiperenlace para acceder a la información (Figura 31).

Figura 31: Visualización de historia clínica con información resaltada

Los términos pasan por el módulo gestor de búsquedas que, en primer lugar, los busca en el repositorio de información. En el caso de que no aparezcan, los traduce en el traductor y los introduce en la ontología, que devuelve un conjunto de términos que incluyen sinónimos y derivados. Estos conjuntos de términos se envían al módulo de integración de información. En el módulo de integración de información el

mediador reenvía las consultas a las bases de datos que pueden mantener esa información. Los términos que se han encontrado se devuelven al mediador, que los formatea de manera uniforme y los devuelve al módulo de gestión de consultas. Dicho módulo almacena estas respuestas en el repositorio de información y, posteriormente, se devuelven los datos al interfaz y se activan los nombres buscados como hiperenlaces.

Si el usuario decide ampliar la información de la historia a partir de los resultados obtenidos del sistema de búsqueda, sólo tiene pulsar sobre el enlace y se mostrarán una serie de campos. En la Figura 32 se muestran una serie de enlaces con información sobre el término “trombosis”.

Figura 32: Historia clínica con enlaces a "trombosis"

Por otro lado, se puede consultar la información recuperada, teniendo en cuenta sus fuentes originales. En la Figura 33 se muestra la información obtenida de las fuentes bibliográficas sobre “fiebre”

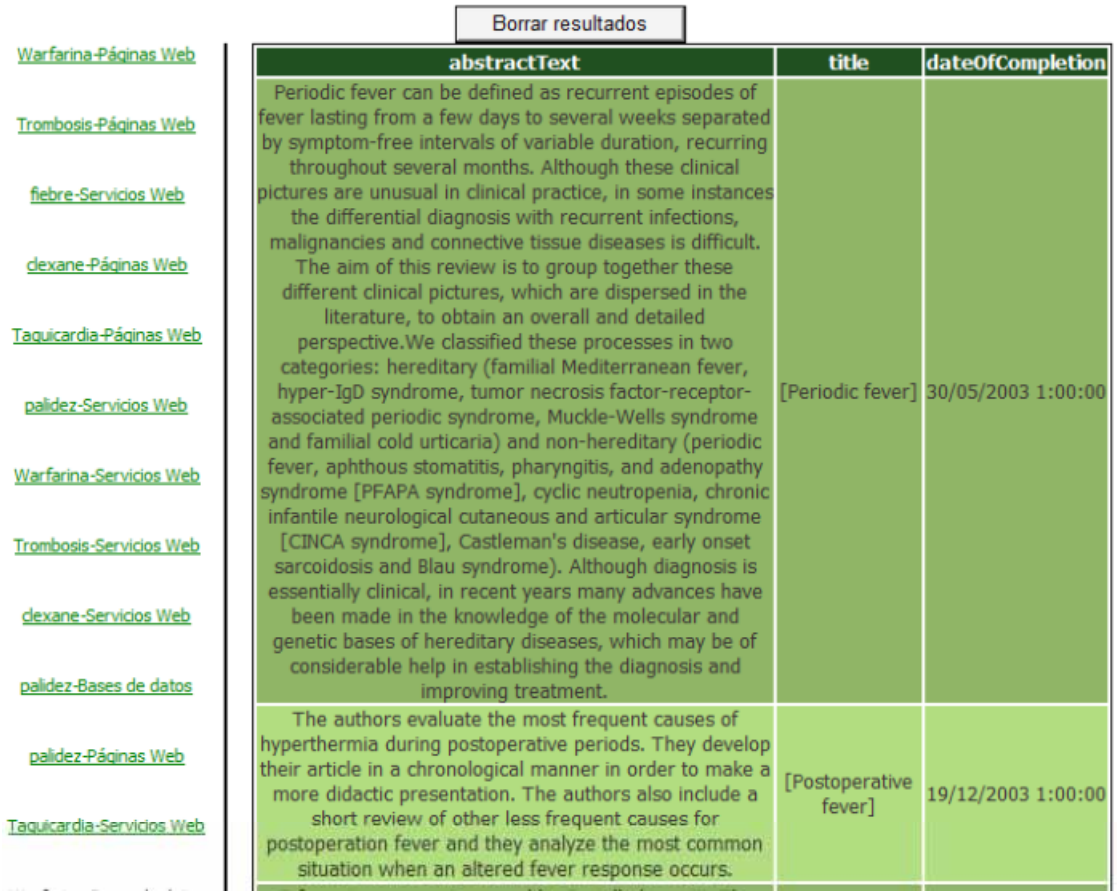


Figura 33: Fuentes bibliográficas sobre "fiebre"

El sistema desarrollado no ha podido probarse en un entorno clínico real con el objetivo de validarlo, debido a las complicaciones relacionadas con la protección de los datos de los pacientes. Sin embargo, el módulo gestor de búsquedas y el módulo de integración se han usado como parte de un sistema de gestión de historia clínica personal “Personal Health Record (PHR) 2.0” desarrollado por la empresa Aldaba en colaboración con el grupo Rnasa-IMEDIR en el marco de un proyecto del plan Avanza.

7.1.5 Discusión

Este sistema de integración demuestra que es posible la aplicación del marco metodológico desarrollado en un campo clínico, por medio de un sistema de “punto de atención al paciente”. El uso de un sistema de integración de datos para el enriquecimiento de la información de historia clínica potencia la utilidad de esta,

permitiendo al clínico disponer de toda la información posible para el diagnóstico, pronóstico y tratamiento de los pacientes.

La aproximación utilizada es la más simple de todas las estudiadas en este trabajo, utilizando un sistema federado basado en *wrapper*/*mediator* y usando sólo la ontología como vocabulario unificado, sin utilizar apenas información de relaciones semánticas, con objetivo de expandir las búsquedas.

El proceso de integración de esta aproximación se divide en dos fases, en una primera se realiza la expansión semántica y de sinónimos y, posteriormente, se realiza la expansión de todos estos términos sobre bases de datos distribuidas heterogéneas. Para este escenario concreto no hace falta que el mediador realice un proceso de integración usando ontologías, sino que simplemente se ocupe de reconducir las consultas e integrar los resultados de vuelta. Por otro lado, el modelo que presenta el mediador es un modelo muy simple, y no requiere de relación entre los datos ni de medidas de granularidad. La adopción de sistemas de integración más compleja no se habría aconsejado en este escenario, debido a que no son necesarias.

Uno de los problemas que presenta esta aproximación es el tiempo de acceso a los datos que, dependiendo de la fuente de datos a la que se acceda, no es inmediato. Para paliar en la medida de lo posible este efecto, el sistema realiza una presentación de los resultados asíncrona, de manera que los resultados se van incorporando según van llegando. A priori este defecto podría hacer pensar que en este caso hubiese sido más aconsejable utilizar una aproximación basada *warehouse*. El problema de la aplicación de esta aproximación *warehouse* en un escenario como éste de inclusión de información en historia clínica electrónica, es que la inclusión de nuevas fuentes de datos no sería tan sencilla como es en el caso de un modelo federado.

7.2 Aplicación de ayuda al diseño de estudios epidemiológicos

7.2.1 Presentación del caso

El análisis de las diferencias entre las secuencias de ADN supone una gran fuente de información para identificar genes que influyen en enfermedades y en procesos biológicos normales. En el estudio del desarrollo de enfermedades, la

información sobre variaciones genéticas es crítica para comprender cuál es la influencia que ejercen los genes y cómo se relacionan las variaciones genéticas y funcionales. Por otro lado, estas diferencias también influyen en la respuesta a terapias y fármacos.

Actualmente, de entre los distintos tipos de variaciones que se estudian a lo largo del genoma, la más utilizada son los SNPs (Single Nucleotide Polymorphisms), por ser la forma más sencilla y más frecuente de variabilidad genética.

La disponibilidad de información genética ha aumentado enormemente con bases de datos online que se actualizan regularmente y proporcionan acceso a gran cantidad de información. Para los investigadores estos servicios permiten, de una manera muy rápida, determinar ciertas características de los genes como, por ejemplo, detalles sobre la secuencia, localización o patrón de expresión. El mayor problema es que no existe un estándar para representar la información biológica de estas bases de datos, por lo que las tareas de integración de estos datos no son triviales.

Por otro lado, cuando se realiza un estudio de asociación de cierta enfermedad, existe la necesidad de priorizar una lista de SNPs con el objetivo de reducir costes, ya que el número de SNPs que podrían encontrarse con las características necesarias para el estudio pueden ser cientos o miles. Estudiar todos estos SNPs implica obtener grandes cantidades de información de diversas bases de datos. Todos estos datos deben ser gestionados adecuadamente.

Tanto el coste de genotipado por SNP, como el diseño del estudio y la gestión de todos estos datos es muy costoso en cuestión de tiempo y dinero. Por esta razón se propone un escenario que consista en una herramienta web que permita obtener una serie de SNPs candidatos para una determinada enfermedad, siguiendo una serie de criterios especificados por los investigadores. Esta herramienta debe permitir obtener datos relacionados con SNPs de diferentes bases de datos genéticas, ordenarlos y priorizarlos con respecto a algún criterio con el objetivo de obtener un conjunto reducido de SNPs ordenados para realizar el estudio. Este sistema facilitaría la tarea de realizar estudios donde estén involucrados miles de SNPs.

El objetivo de este caso de estudio es el desarrollo de un sistema de información que permita obtener una lista de SNPs priorizada a partir de una serie de datos como un conjunto de genes o enfermedades, utilizando los datos integrados de diversas bases de datos.

Las funciones principales de la arquitectura consisten en obtener información de genes a partir de una consulta sobre una enfermedad concreta. A partir de la información de los genes, obtener información sobre los SNPs de esos genes. Finalmente, a partir de ciertas restricciones o filtros impuestos por el investigador, obtener finalmente un conjunto de información de los SNPs ordenada o priorizada según los criterios que definen las restricciones o filtros.

7.2.2 Análisis

A continuación se exponen los resultados de la fase de análisis.

7.2.2.1 Análisis de requisitos

El objetivo principal impuesto a la herramienta a desarrollar fue que partiendo de la especificación de una enfermedad o de un conjunto de genes, se obtuvieran los SNPs que podrían tener relación con ella. Para ello se utilizarían los genes que tuvieran relación con esa enfermedad. Finalmente, del conjunto total de SNPs, poder organizarlos en un orden de importancia, según unos parámetros que los investigadores considerasen, para conseguir un conjunto significativo de SNPs, lo que les permitiría reducir considerablemente la cantidad de SNPs a estudiar, reduciendo de esta manera los costes de genotipado.

La entrada del sistema puede ser la enfermedad, un conjunto de genes que el investigador puede estar interesado en que entren en el sistema a pesar de no tener la evidencia de que estén relacionados con la enfermedad o incluso un conjunto de SNPs que el investigador quiera que aparezcan en el estudio, además de los criterios de priorización.

Los criterios de priorización incluyen los siguientes parámetros:

- SNP sinónimo: Si la mutación es silenciosa en el sentido de que el código mutado genera la misma proteína que el gen original.

- SNP no-sinónimo: Si la proteína que codifica el gen mutado es distinta que la original.
- SNP intrónico: Si el SNP está situado en una región del gen de manera que no codifica ninguna proteína.
- TagSNP: Existe una gran correlación entre SNPs que están localizados en la misma región, por lo que es posible usar algún SNP (tagSNP), para representar al resto de SNPs de la zona.
- Frecuencia alélica: Es la proporción con la que aparece cierto tipo de mutación.

La herramienta debe estar disponible vía web, por lo que se integrará en un portal web para que sea accesible desde cualquier equipo con acceso a internet.

La herramienta debe ser abierta a futuras inclusiones de bases de datos o filtros, para poder realizar nuevas operaciones sobre los datos obtenidos, o poder integrar esta herramienta con otras dedicadas a la recuperación de datos biológicos.

Basándose en los requisitos del sistema de integración, se ha evaluado la importancia de los factores de decisión ordenados por decisiones clave.

En el caso de la arquitectura (Tabla 16), en este caso el factor de mayor importancia es el de la actualización y la evolución del modelo, debido a que el sistema de recuperación de SNPs debe obtener esta información actualizada y es importante que el sistema pueda añadir nuevas bases de datos. Por otro lado, el control de los datos tiene poca importancia en este caso de estudio, porque se intentará que las bases de datos a integrar sean públicas y no hay información clínica.

El resto de los factores (inversión inicial, escalado, rendimiento y disponibilidad) presentan una importancia media, ya que, aunque son factores que se tienen que tener en cuenta, no son determinantes en este caso de estudio.

Finalmente, la importancia de los costes por fuente o por fuente añadida después del diseño no es muy alta debido a que se espera que el sistema no presente un modelo muy complejo.

Tabla 16: Importancia de los factores de decisión en arquitectura escenario 2

Arquitectura	
	Importancia
Inversión inicial	Media
Actualización	Alta
Escalado datos	Media
Rendimiento	Media
Disponibilidad	Media
Control datos	Baja
Evolución modelo	Alta
Coste por fuente	Media
Coste por fuente añadida	Media

Al igual que en el escenario 1, se espera que este sistema pueda ser utilizado por usuarios no expertos, por lo que la importancia será alta, mientras que la ausencia de necesidad de un acceso directo a las fuentes hace que tenga una importancia baja (Tabla 17).

Tabla 17: Importancia de los factores de decisión en tipo de integración escenario 2

Integración	
	Importancia
Usuario no experto	Alta
Acceso fuentes	Baja

En cuanto a la importancia de los factores de decisión en la estrategia de integración, que se resume en la Tabla 18, se espera que el sistema trabaje bien con un número importante de bases de datos, pero sólo en lectura. Por otro lado, la evolución del modelo a lo largo del tiempo debería de ser flexible.

Tabla 18: Importancia de los factores de decisión en estrategia de integración escenario 2

Estrategia	
	Importancia
Nº Bases datos	Alta
Acceso escritura	Baja
Evolución Modelo	Alta

Al contrario que en el escenario 1, en el que parecía que sólo sería necesario integración vertical, en el escenario 2 se podrán dar los dos tipos de integración, por lo que el valor de importancia de la integración horizontal será alto, mientras que la importancia de que aparezcan registros repetidos también será alta. En este caso también pueden aparecer problemas de granularidad, con lo que tendrá una importancia media. En la Tabla 19 se muestra un resumen de estos factores.

Tabla 19: Importancia de los factores de decisión en integración semántica escenario 2

Integración semántica	
	Importancia
Registros repetidos	Alta
Granularidad	Media
Integración horizontal	Alta

Finalmente, en lo que se refiere a los factores de decisión en gestión de consultas (Tabla 20), la evolución del modelo tiene una importancia alta como se vio anteriormente, mientras que la sencillez en la implementación tiene una importancia media.

Tabla 20: Importancia de los factores de decisión en gestión de consultas escenario 2

Consultas	
	Importancia
Evolución modelo	Alta
Facilidad implementación	Media

7.2.2.2 Análisis de casos de uso

En este escenario de aplicación de integración de datos, existe un único actor principal, el investigador que realiza el análisis, aunque también se tendrá en cuenta un actor más, que será el encargado de configurar los flujos de integración.

- Recuperar SNP de Enfermedad Ordenados: Este es el gran caso de estudio del sistema, en el que el usuario indica una enfermedad sobre la

que desea obtener los SNPs que estén relacionados con ella en cierto orden. Este caso de uso incluye una serie de casos de uso:

- Recuperar genes de enfermedad: A partir de la enfermedad indicada por el usuario, el sistema devuelve un conjunto de genes que están relacionados con la enfermedad.
- Recuperar SNPs de genes: A partir de los genes obtenidos en el paso anterior o de los indicados por el propio usuario, el sistema recupera los SNPs relacionados con esos genes. Este caso de estudio incluye a su vez otro caso de estudio sobre la información de los SNPs
  - Recuperar información sobre SNPs: A partir de la lista de SNPs obtenida del paso anterior o de los SNPs definidos por el usuario, el sistema amplía la lista con información sobre estos SNPs recogida en otras bases de datos.
- Ordenar SNPs: A partir de los SNPs y de la información relativa a ellos, y utilizando los criterios definidos por el investigador, ordena los SNPs por relevancia.

7.2.2.3 Análisis de datos a integrar

A partir de los requisitos formulados por los usuarios, es necesario encontrar una serie de bases de datos que, en primer lugar, permitan relacionar enfermedades con genes, posteriormente, encontrar los SNPs que pertenecen a dichos genes y, finalmente, encontrar las características de estos SNPs.

7.2.2.3.1 Contenido de los datos

Existen una gran cantidad de bases de datos de variaciones genómicas, como se puede ver en el sitio web que recopila una gran parte de estas bases de datos, Human Genome Variation Society (www.hgvs.org).

Sin embargo, la mayoría de estas bases de datos son específicas de algunos genes o enfermedades, por lo que se han seleccionado algunas de las bases de datos genéricas más conocidas:

- DbGaP (http://www.ncbi.nlm.nih.gov/gap): La base de datos de genotipos y fenotipos (database of Genotipes And Phenotipes, dbGAP), almacena estudios que investigan la relación entre el genotipo y el fenotipo. Pertenece al NCBI.
- GWAS Central (http://www.gwascentral.org): Anteriormente conocida como Human Genome Variation Database of Genotype to Phenotype Information) es una base de datos que almacena los resultados de estudios de asociación que buscan relacionar genotipo con fenotipo.
- KEGG (http://www.genome.jp/kegg/): La Kyoto Encyclopedia of Genes and Genomes son un conjunto de bases de datos que almacenan información de sistemas, genómica y química. De entre todas las bases de datos que presenta, concretamente para este caso de estudio, la KEGG disease es especialmente útil, ya que proporciona una colección de enfermedades y su relación con factores genéticos, ambientales, marcadores y fármacos.
- OMIM (http://www.ncbi.nlm.nih.gov/sites/entrez?db=omim): Online Mendelian Inheritance in Man, es la base de datos más antigua de todas las analizadas, ya que comenzó en los sesenta, aunque no se publicó online hasta 1985. Contiene referencias a todas las enfermedades mendelianas conocidas y sus genes asociados.
- PharmGKB (http://www.pharmgkb.org/): El portal PharmGKB (Pharmacogenomics Knowledge Base) fue creado por la Universidad de Stanford y el NIH para estudiar cómo afectan las variaciones genéticas en la reacción a los fármacos. Actualmente esta base de datos contiene información genética, genómica, molecular y clínica sobre estudios farmacogenéticos.
- HGMD (http://www.hgmd.org/): La base de datos de mutaciones genéticas (Human Gene Mutation Database), almacena las variaciones publicadas de enfermedades genéticas. Probablemente sea base de datos central de asociación entre enfermedad y una mutación.

Además de estas bases de datos de asociación entre un genotipo un fenotipo también se han estudiado otras bases de datos necesarias para el proceso de integración, como Gene Ontology (http://www.geneontology.org/) para recopilar información sobre los genes, así como dbSNP (http://www.ncbi.nlm.nih.gov/projects/SNP/) para recopilar información sobre SNPs.

7.2.2.3.2 Interfaz

En este punto se analizará el tipo de conexión que la fuente de datos ofrece a sus usuarios, así como la posibilidad o no de descargar los datos. También se analizarán las restricciones de acceso.

- dbGaP no ofrece ningún tipo de API para el acceso a los datos. Existen dos políticas de acceso, una libre para ciertos tipos de datos y otra, de acceso controlado previa solicitud, para los datos más sensibles. Es posible bajar la base de datos libre e instalarla en local.
- GWASCentral ofrece acceso a los datos a través de peticiones http y ofrece también acceso para descarga de la base de datos completa.
- KEGG ofrece acceso mediante servicios web y REST a los datos de sus bases de datos.
- OMIM ofrece acceso a sus bases de datos mediante los servicios web del NCBI. Por otro lado también es posible bajar las bases de datos.
- PharmGKB permite acceso mediante servicios web a sus datos o mediante descarga de sus bases de datos.
- HGMD requiere login para poder realizar las búsquedas sobre su propia web y no permiten ningún tipo de acceso remoto tipo servicios web, acceso a sus bases de datos, ni la posibilidad de descargarla.
- dbSNP permite acceso mediante los servicios web del NCBI, descarga de la base de datos, como acceso desde el Ensembl a través de BioMart, a través de acceso a sus servidores MySQL o descarga de sus bases de datos.
- GO permite la descarga de los datos, el acceso a través de servicios web o el acceso a través de su servidor de MySQL.

7.2.2.3.3 Actualizaciones

Cualquiera de las fuentes de datos analizadas es regularmente actualizada.

7.2.2.3.4 Modelo de datos y anotación semántica

Cada una de las bases de datos presenta un modelo de datos oculto a través de su interfaz, salvo en alguno de los casos que permiten la descarga de la base de datos, que presenta una estructura relacional en algunos casos o de ficheros de texto en otros.

En cuanto a la anotación semántica, cualquiera de las bases de datos analizada cumple con la anotación semántica necesaria para los procesos de integración necesarios en este escenario.

7.2.2.3.5 Clasificación Busse

Según el modelo de Busse, el análisis de las fuentes de datos para este escenario será el siguiente:

- Autonomía: Todas las bases de datos analizadas pertenecen a los organismos que las mantienen, por lo tanto tienen una total autonomía de diseño y de ejecución, salvo aquellas que pueden ser descargadas, que pierden su autonomía a favor del usuario.
- Heterogeneidad:
    - Sintáctica tecnológica y de interfaz: Como se ha comentado anteriormente, GWAS Central, KEGG, OMIM, PharmGKB, DbSNP y GO presentan acceso a través de servicios web o similar, mientas que dbGAP sólo mediante descarga y HGMD sólo mediante el interfaz web.
    - Sintáctica de consulta: A pesar de que al presentar interfaz web no permiten acceder a la totalidad del modelo, la mayoría de ellas pueden ser descargadas a local. En todo caso, las necesidades de integración permiten contar simplemente con la información proporcionada por el interfaz web.
    - De modelo y lógica: Los requisitos de este sistema de integración requieren un modelo que relacione enfermedad-gen-SNP. Todas

las fuentes de datos analizadas cumplen con estos requisitos de modelo.

- Distribución: El uso de DbGaP requiere que se instale en local, el resto de las bases de datos pueden ser accesibles a través de interfaz web, por lo que pueden estar localizadas en sus instituciones de origen.

Una vez analizadas todas estas bases de datos, se descartan DbGaP, ya que no proporciona interfaz web y HGMD por no proporcionar mecanismos de acceso. El resto de las fuentes de datos son válidas para el proceso de integración.

7.2.2.4 Selección de arquitectura

A partir de los requisitos de la aplicación, intuitivamente, teniendo en cuenta que el sistema tendrá que obtener una gran cantidad de SNPs, de gran cantidad de genes, se descartan los esquemas de navegación por links que supondrían una tarea tediosa. Además de la restricción impuesta por los requisitos, la heterogeneidad en las fuentes de datos que pertenecen a distintas organizaciones, no lo hace una alternativa viable.

Tabla 21: Valoración arquitectura escenario 2

Arquitectura		**Federado**	**Warehouse**
	Importancia	Valor	Valor
Inversión inicial	Media	Medio	Alto
Actualización	Alta	Alto	Bajo
Escalado datos	Media	Bajo	Alto
Rendimiento	Media	Bajo	Alto
Disponibilidad	Media	Bajo	Alto
Control datos	Baja	Alto	Alto
Evolución modelo	Alta	Alto	Bajo
Coste por fuente	Media	Alto	Bajo
Coste por fuente añadida	Media	Bajo	Alto

En lo que se refiere al uso de una arquitectura tipo *warehouse* o federada, en la Tabla 21 se analizan cada uno de los factores con respecto a las dos arquitecturas.

Cabe recordar que la mayoría de los datos se encuentran dispuestos en diversas bases de datos externas dispersas en varios sitios. Un sistema *warehouse* obligaría a recoger todos los datos de estas bases externas y adaptarlos al modelo centralizado del *warehouse,* con un coste alto de inversión inicial. Otro factor importante con respecto a las fuentes de datos es que estas cambian muy frecuentemente, añadiendo nuevas variaciones, con lo que un sistema *warehouse* tendría que actualizar constantemente sus datos y adaptar estos nuevos datos al modelo central, por lo que el *warehouse* tiene un valor de actualización y de evolución del modelo bajo.

Frente a estos factores, el modelo federado proporciona un coste de inversión medio, una buena capacidad de actualización y, como desventajas con respecto al *warehouse*, están su capacidad de escalado, rendimiento y disponibilidad.

Finalmente, es posible que se puedan añadir nuevas bases de datos de variaciones o de datos de SNPs según vayan surgiendo, aunque el modelo que requiera este escenario podría no cambiar sensiblemente. La adaptación de estas nuevas bases de datos a un modelo común en el *warehouse* podría ser costosa y requeriría muchos cambios en las estructuras de la base de datos, sin embargo, en un modelo federado, apenas implicaría modificar los tipos de consulta.

En este punto parece que el modelo de arquitectura más adecuado puede ser el federado, pero también es necesario tener en cuenta el uso que se le puede dar a este tipo de sistema. Uno de los requisitos que tiene la aplicación es que sea el propio usuario el que escoja, no sólo las fuentes de los datos que se van a integrar, sino también cómo se van a integrar. Es decir, el usuario debe de tener la capacidad de configurar su propio flujo de integración.

7.2.2.5 Tipo de modelo de integración

El objetivo del sistema desarrollado para este escenario es que, por un lado, permita la integración sencilla a partir de una aplicación web de una serie de base de datos previamente definida pero, por otro, que en caso de que el usuario tenga suficientes conocimientos y la necesidad de la construcción de un modelo de integración flexible, pueda realizar él mismo el flujo de integración, con lo que los

valores de usuario no experto tendrán valores altos tanto para abierta (alto) como para cerrada (medio).

Tabla 22: Valoración modelo de integración escenario 2

Integración		**Abierta**	**Cerrada**
	Importancia	Valor	Valor
Usuario no experto	Alta	Alto	Medio
Acceso fuentes	Baja	Alto	Bajo
Coste por fuente	Media	Bajo	Alto
Coste por fuente añadida	Media	Medio	Medio

De esta manera, no se puede encuadrar el sistema a desarrollar en las categorías de sistema de federación abierto o cerrado (Tabla 22) ya que, por un lado, la aplicación web será un sistema cerrado, donde el usuario escogerá las bases de datos donde buscar, pero no gestionará de ninguna manera la estrategia de integración, mientras que, si es el usuario el que configura su propio flujo de integración, prescinde del mediador para convertirse él mismo en el mediador, por lo que se aproximaría a un sistema abierto.

Sin embargo, tampoco este caso sería un sistema abierto al uso, ya que el usuario estaría restringido a las entradas y salidas que presenta cada una de las fuentes de datos disponibles, en lugar del acceso completo a las bases de datos.

De esta manera, tendría que existir un modelo común sobre el que el wrapper adaptase los datos del modelo especifico de cada fuente de datos, de manera que el usuario realizaría sólo una pequeña parte de la función de mediador.

En lo que al acceso a los datos originales se refiere, el valor alto del sistema cerrado viene definido por la naturaleza de la integración cerrada.

En lo que se refiere a los costes de añadir nuevas fuentes, tanto en diseño como posteriormente, en una arquitectura cerrada los costes son, de media, mayores que en una arquitectura abierta.

7.2.2.6 Estrategia de integración

Tal y como se adelantó en la descripción del marco metodológico, es complicado encontrar un escenario donde se integren diversas bases de datos biomédicas de manera que sea necesario utilizar todos los datos de origen, tal y como exige una estrategia bottom-up y este caso no lo es. En la Tabla 23 se resumen los factores que influyen en cada estrategia.

Tabla 23: Valoración estrategia de integración escenario 2

Estrategia		**Top-down**	**Bottom-up**
	Importancia	Valor	Valor
Nº Bases datos	Alta	Alto	Bajo
Acceso escritura	Baja	Bajo	Alto
Evolución Modelo	Alta	Alto	Bajo
Coste por fuente	Media	Medio	Alto
Coste por fuente añadida	Media	Medio	Alto

Por definición de los modelos *top-down* y *bottom-up*, los valores son los mismos que en el escenario 1.

En este punto, es necesario definir el modelo global. Los datos necesarios en el punto final de integración son los siguientes:

- Enfermedad: Uno de los puntos de entrada del sistema es la enfermedad o enfermedades de las que se pretenden extraer las variaciones.
- *Pathways*: Otro de los puntos de entrada puede ser la obtención de los *pathways* relacionados con cierta enfermedad.
- Genes: Otro de los puntos de entrada del sistema, pero también pueden ser obtenidos a partir de las enfermedades o de los *pathways*.
- Variaciones: El resultado del sistema de integración.

7.2.2.7 Tipo de integración

En este escenario de integración se trabaja claramente en varios niveles de integración. En la Tabla 24 se aprecia que existen requisitos de eliminar los registros repetidos en integración vertical y que se requiere integración horizontal también.

Por un lado, a través de diversas bases de datos que contienen datos similares o iguales se realiza una integración vertical, recopilando todos los datos e integrándolos en una sola salida. Es el caso de la obtención de los genes relacionados con una enfermedad o con un *pathway*, en la que se obtienen una gran cantidad de genes y puede haber resultados repetidos. Es por ello necesario contar con algún mecanismo de anotación semántica que permita diferenciar los resultados obtenidos de diversas bases de datos. Con respecto a los valores, son similares a los del escenario 1, diferenciando que, en fusión, la granularidad en este caso tiene un valor medio y la abstracción un valor alto, debido a los requisitos del escenario.

Tabla 24: Valoración tipo de integración semántica escenario 2

Integración semántica		**Agregación**	**Fusión**	**Abstracción**	**Suplementación**
	Importancia	Valor	Valor	Valor	Valor
Registros repetidos	Alta	Bajo	Alto	Alto	Alto
Granularidad	Media	Bajo	Medio	Alto	Alto
Integración horizontal	Alta	Bajo	Bajo	Bajo	Alto
Coste por fuente	Media	Bajo	Medio	Alto	Alto
Coste por fuente añadida	Media	Bajo	Medio	Alto	Alto

Por otro lado, también es necesario complementar los datos obtenidos buscando relaciones semánticas entre ellos, en este caso, es necesario obtener los genes o pathways relacionados con una enfermedad, los SNPs relacionados con esa enfermedad o con los genes obtenidos anteriormente y es necesario completar la información de los SNPs obtenidos.

7.2.2.8 Procesado, planificación y ejecución de consultas

La Tabla 25 muestra la importancia alta de la evolución del modelo, ya que el sistema a desarrollar debe de ser modular, debido a la necesidad que plantea la posibilidad de añadir nuevos servicios *wrapper* que adapten nuevas fuentes de datos. En un modelo LaV, la evolución del modelo es alta, mientras que en un GaV, es baja, debido a la complejidad de evolucionar este tipo de consultas, a pesar de que, en origen, es más sencillo de implementar.

Tabla 25: Valoración gestión de consultas escenario 2

Consultas		LaV	GaV
	Importancia	Valor	Valor
Evolución modelo	Alta	Alto	Bajo
Facilidad implementación	Media	Bajo	Alto
Coste por fuente	Media	Alto	Bajo
Coste por fuente añadida	Media	Bajo	Alto

7.2.2.9 Métrica y costes

En este paso se evalúa el impacto de las decisiones tomadas teniendo en cuenta los factores de decisión clave y la valoración de estos, para cada opción.

Tabla 26: Impacto arquitectura escenario 2

Arquitectura		Federado		Warehouse	
	Importancia	Valor	Impacto	Valor	Impacto
Inversión inicial	Media	Medio	Medio	Alto	Medio
Actualización	Alta	Alto	Positivo	Bajo	Negativo
Escalado datos	Media	Bajo	Medio	Alto	Medio
Rendimiento	Media	Bajo	Medio	Alto	Medio
Disponibilidad	Media	Bajo	Medio	Alto	Medio
Control datos	Baja	Alto	Medio	Alto	Medio
Evolución modelo	Alta	Alto	Positivo	Bajo	Negativo
Coste por fuente	Media	Alto	Medio	Bajo	Medio
Coste por fuente añadida	Media	Bajo	Positivo	Alto	Negativo

Así, en lo que respecta a la arquitectura (Tabla 26), los factores más importantes son la capacidad de actualización, de evolución del modelo y el coste de añadir nuevas fuentes. En estos tres factores, el impacto es positivo en el federado y negativo en el *warehouse*, por lo que parece claro que el mejor tipo de arquitectura de integración para este escenario es la federación.

Los puntos a favor del modelo *warehouse* son el escalado de datos, el rendimiento y la disponibilidad. Estos puntos pueden solucionarse con un modelo

híbrido que haga una copia local de las bases de datos que se actualicen diariamente. Esta actualización diaria solucionaría también, en el caso de seleccionar *warehouse*, el problema del modelo *warehouse* de actualización de los datos, pero no el de evolución del modelo.

Tal y como se adelantó, el usuario debe tener la capacidad de crear sus propios flujos de integración pero, por otro lado, el uso de este tipo sistemas no debería quedarse simplemente en el desarrollo una aplicación que obtenga los SNPs, sino que además debería de facilitar su posterior análisis.

Teniendo en cuenta estos dos factores, dentro de los modelos de federación propuestos en el marco metodológico, se ha escogido la federación por *workflows*. Por un lado proporcionan al usuario la capacidad de gestionar el flujo de integración de los datos, conectando conforme a sus necesidades unos módulos con otros sin necesidad de tener grandes conocimientos de informática ni de programación. Por otro lado, los módulos finales pueden ser módulos de análisis que permitan realizar procesados sobre los datos.

Sin embargo, teniendo en cuenta que el usuario al que está dirigida esta aplicación puede tener problemas en desarrollar los flujos de integración, se creará también una vista web de una manera transparente al usuario permita llamar a los *workflows* de manera automática.

La realización de las copias locales para el modelo federado es compatible con la selección de un modelo federado de integración por *workflows*.

Tabla 27: Impacto modelo de integración escenario 2

Integración		**Abierta**		**Cerrada**	
	Importancia	Valor	Impacto	Valor	Impacto
Usuario no experto	Alta	Alto	Positivo	Medio	Medio
Acceso fuentes	Baja	Alto	Medio	Bajo	Medio
Coste por fuente	Media	Bajo	Medio	Alto	Medio
Coste por fuente añadida	Media	Medio	Medio	Medio	Medio

En el caso de la selección del tipo de integración (Tabla 27), la elección no es tan simple, debido a que la diferencia entre los dos modelos de integración no está muy definida. Es por eso que se ha escogido un modelo de integración "semi-abierta" donde, por un lado, se realiza una integración cerrada de cara al usuario de la web, mientras que el conjunto de workflows que se utilizan definen la posibilidad de realizar una integración abierta.

Tabla 28: Impacto estrategia de integración escenario 2

Estrategia		**Top-down**		**Bottom-up**	
	Importancia	Valor	Impacto	Valor	Impacto
Nº Bases datos	Alta	Alto	Positivo	Bajo	Negativo
Acceso escritura	Baja	Bajo	Medio	Alto	Medio
Evolución Modelo	Alta	Alto	Positivo	Bajo	Negativo
Coste por fuente	Media	Medio	Medio	Alto	Medio
Coste por fuente añadida	Media	Medio	Medio	Alto	Medio

En cuanto a los costes de la selección de estrategia, resumidos en la Tabla 28, el coste de implementación tanto en diseño, como posteriormente de un modelo *bottom-up* utilizando *workflows*, no sería alto, debido a la propia naturaleza de la manera en la que se recuperan los datos utilizando *workflows*. Sin embargo, debido a la cantidad de bases de datos que pudiesen integrarse, el número de entidades con las que trabajaría el usuario sería muy grande. Se considera que es suficiente con definir una serie de entidades, que son las que se utilizarán en el proceso de integración. Es por eso que en este caso está justificado el modelo *top-down*.

Tabla 29: Impacto tipo de integración semántica escenario 2

Integración semántica		**Agregación**		**Fusión**	
	Importancia	Valor	Impacto	Valor	Impacto
Registros repetidos	Alta	Bajo	Negativo	Alto	Positivo
Granularidad	Media	Bajo	Medio	Medio	Medio
Integración horizontal	Alta	Bajo	Negativo	Bajo	Negativo
Coste por fuente	Media	Bajo	Medio	Medio	Medio
Coste por fuente añadida	Media	Bajo	Medio	Medio	Medio

Integración semántica		Abstracción		Suplementación	
	Importancia	Valor	Impacto	Valor	Impacto
Registros repetidos	Alta	Alto	Positivo	Alto	Positivo
Granularidad	Media	Alto	Medio	Alto	Medio
Integración horizontal	Alta	Bajo	Negativo	Alto	Positivo
Coste por fuente	Media	Alto	Medio	Alto	Medio
Coste por fuente añadida	Media	Alto	Medio	Alto	Medio

En el caso de la elección del tipo de integración semántica, que se resume en la Tabla 29, parece necesario el uso de dos tipos de modelos de integración semántica. Por un lado integración vertical, que requiere integrar los registros repetidos, y por otro un modelo de integración horizontal. La integración horizontal debe ser implementada usando suplementación, ya que su impacto es positivo con respecto a la integración horizontal, mientras que para la vertical se utilizará la fusión, porque tiene igual impacto positivo que la abstracción pero su coste de implementación es menor.

Tabla 30: Impacto gestión de consultas escenario 2

Consultas		LaV		GaV	
	Importancia	Valor	Impacto	Valor	Impacto
Evolución modelo	Alta	Alto	Positivo	Bajo	Negativo
Facilidad implementación	Media	Bajo	Medio	Alto	Medio
Coste por fuente	Media	Alto	Medio	Bajo	Medio
Coste por fuente añadida	Media	Bajo	Medio	Alto	Medio

En cuanto a la selección del modelo de consultas (Tabla 30), independientemente de que la elección de workflows e integración top-down imposibilita la elección del GaV, la posibilidad de un modelo que se modifique a lo largo del tiempo llevaría a la selección del LaV, donde las vistas vengan representadas por los *wrappers* que adaptan los datos de las fuentes locales al modelo global.

El esquema de procesado y ejecución de consultas lo definirá el sistema de gestión del *workflow* a partir del que se implementará la aplicación de integración.

El modelo general, al utilizar *workflows,* será el de servicios que reciben una consulta, a través del *wrapper* la transforman al lenguaje de la fuente de datos y la ejecutan. En el caso de la integración vertical, el mediador divide la consulta en cada una de las fuentes de datos que utilizan el mismo esquema visto anteriormente.

Finalmente se ejecutan las consultas utilizando el esquema de cada fuente de datos.

7.2.2.10 Aspectos tecnológicos

7.2.2.10.1 Seguridad y privacidad de los datos

En este caso, al no utilizar en ninguno de los puntos datos de pacientes ni bases de datos que no sean públicas, no es necesario tener en cuenta aspectos de seguridad.

Aún en el caso de que alguna de las bases de datos que se puedan incluir en un futuro pueda contener datos de pacientes, como pueden ser bases de datos de estudios de asociación, la única información que saldrá de esa base de datos hacia el sistema serán los identificadores de genes o variaciones asociados con determinada enfermedad, en ningún caso los datos de los pacientes.

7.2.2.10.2 Framework de integración

La selección del *framework* de integración es un aspecto que condiciona los resultados de implementación. Esta elección viene condicionada en este caso por la arquitectura federada con *workflows*.

Es necesario, por tanto, encontrar un sistema de *workflows* que permita cumplir con todos los requisitos especificados anteriormente.

De entre todos los *frameworks* de *workflows*, se ha escogido Biomoby, por varios aspectos. En primer lugar se ha convertido en un estándar de facto en la implementación de servicios en bioinformática, existiendo una gran cantidad de servicios, todos ellos disponibles en Moby Central. Estos servicios abarcan desde

herramientas de acceso a bases de datos, a herramientas de análisis que podrían utilizarse dentro del *workflow* que se propone en este trabajo.

Por otro lado, Biomoby proporciona una ontología básica con varios tipos de entidades, objetos y *namespaces* que permiten realizar una anotación semántica en todas las fases del desarrollo, imprescindible para la correcta integración.

Además, Biomoby presenta su propio modelo de datos estándar, publicado en Moby Central, que además puede ser extendido ajustándose a las necesidades de integración.

Finalmente, la utilización de servicios Biomoby implica la publicación tanto de los servicios como de los objetos creados en Moby Central, de manera que cualquiera de estos servicios u objetos podrán ser reutilizados por otros investigadores, en aplicaciones que podrán tener que ver o no con los objetivos de este escenario.

7.2.3 Desarrollo

Una vez establecido el uso de Biomoby para el desarrollo del sistema de integración, se dividió la funcionalidad de esta en diferentes módulos, cada uno de los cuales implementado como un servicio. Además, se buscó que los módulos pudiera ser reutilizables para cualquier otro futuro uso. Un esquema de la arquitectura escogida puede verse en la Figura 34. El sistema parte de una entrada que es la enfermedad de la cual se está buscando información y, a través de las bases de datos A, que podría ser PharmGKB, obtiene los SNPs, a través de la base de datos B, que podría ser KEGG, se obtienen genes y a partir de la base de datos C (dbSNP) obtiene información sobre los SNPs, lo cual genera una lista de SNPs por cada base de datos de origen. Posteriormente, un modulo integra y ordena los SNPs según los criterios especificados.

Para conseguir todas las funcionalidades especificadas, cada uno de los módulos expone una interfaz Biomoby para su interconexión y acceso a los datos biomédicos utilizando servicios web o directamente conexión SQL cuando sea posible.

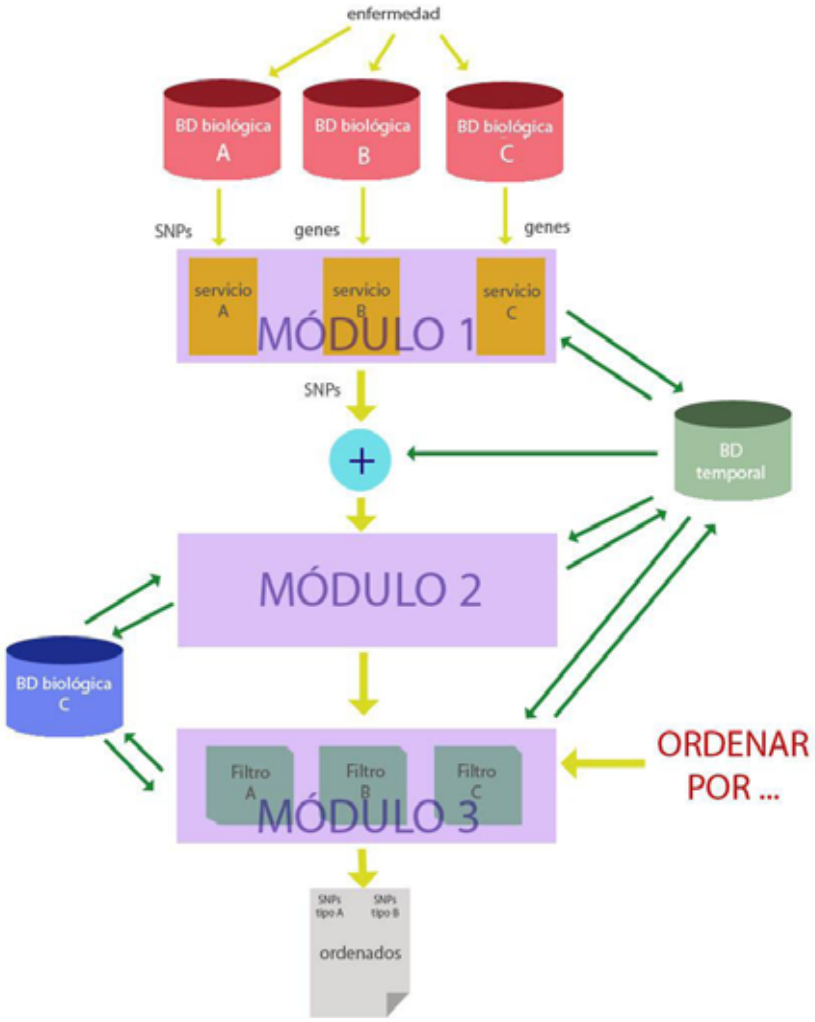


Figura 34: Arquitectura escenario 2

De todas las fuentes de datos seleccionadas, se decidió implementar solamente un subconjunto de ellas. Cualquier otra fuente de datos podrá ser añadida de la misma forma. De esta manera se obtienen los siguientes módulos:

- GetDiseasePhenotypeFeatures: A partir de un identificador o string con el nombre de la enfermedad, obtiene información de la enfermedad, descripción y sus identificadores.
- GetGeneFromDiseasePhenotype: A partir de un identificador de un fenotipo de enfermedad, usando las referencias cruzadas y los sinónimos se obtienen se obtienen un conjunto de identificadores de genes de las bases de datos de Ensembl.
- GetBasicGeneDataFromGenBankID: A partir de una colección de identificadores de genes (formato genbank) obtiene una lista de genes e información asociada.

- GetGeneFromDiseasePhKEGG: A partir del identificador de enfermedad recupera un conjunto de identificadores de genes de la base de datos KEGG
- GetGeneFromDieasePhPharmGKB: A partir de un identificador de enfermedad recupera un conjunto de identificadores de genes de la base de datos PharmGKB.
- GetGeneFromDiseasePhGO: A partir de un identificador de enfermedad recupera un conjunto de identificadores de genes de la base de datos GO.
- GetSNPFromDiseaseGene: A partir de los identificadores de genes se obtienen los SNPs conocidos que aparecen en esos genes a través de la base de datos de Ensembl.
- GetSNPFromDiseasePh: A partir de un identificador de enfermedad recupera un conjunto de SNPs a través de la base de datos PharmGKB.
- FilterSNPbyCodingRegion: Puntúa los SNPs según las opciones de búsqueda del investigador (Sinónimo, intrónico o no sinónimo) y devuelve una lista de los SNPs con su puntuación.
- FilterTagSNP: Puntúa los SNPs según si es o no TagSNP y devuelve un lista de los SNPs con su puntuación.
- FilterSNPsbyAlleleFreq: Puntúa los SNPs según su frecuencia alélica y devuelve una lista de los SNPs con su puntuación.
- GetOrderedSNPS: a partir de un conjunto de SNPs se ordenan a partir de un valor score que viene dado en función de su puntuación en los filtros. Devuelve una lista de los SNPs ordenados por su puntuación.

De la misma manera, se extendieron una serie de Objetos Biomoby para adaptarlos a las necesidades de este sistema.

- Gene_feature: recoge características de los genes que no aparecen en el Gene original, pero que son necesarias para el desarrollo de este trabajo.
- SNP_Feature: que engloba las características de un SNP necesarias para ordenar los SNPs, pero que son de uso común, como pueden ser la frecuencia alélica, si es tagSNP o no, o si es sinónimo, no sinónimo o intrónico.

En lo que se refiere a la arquitectura de integración, los servicios que actúan de servicio *wrapper* son los que están relacionados con las bases de datos, mientras que los servicios que actúan como mediadores son los que integran, tanto la información de todos los genes recuperada, como la información de todos los SNPs recuperada.

Los datos que se recuperan de diferentes bases de datos pueden tener diferentes tipos de conflictos (semánticos, de formato, etc.). Para resolver estos problemas cada módulo Biomoby transforma los datos de su formato original al modelo común, en este caso el modelo Biomoby o a las extensiones desarrolladas ad-hoc de este.

7.2.3.1 Herramienta web

Con el objetivo de que un usuario sin conocimientos de informática ni de workflows pueda usar el sistema de integración, se desarrolló también un sistema de información web que permite la ejecución automática de todos los servicios en el orden correcto.

El sistema permite recuperar la información necesaria de una serie de bases de datos concretas, integrarla y presenta al usuario una lista con los SNPs seleccionados ordenados.

El usuario introduce información sobre la enfermedad que quiere buscar y/o una lista de genes sobre los que quiere buscar, las bases de datos sobre las que quiere extender la búsqueda, así como una serie de criterios sobre los SNPs que son los que el sistema utilizará para ordenar lo SNPs.

El sistema se desarrolló utilizando JEE y el patrón MVC, donde el modelo es el sistema de integración.

7.2.4 Pruebas

En la Figura 35 se muestra un ejemplo de la ejecución del sistema. Un usuario, usando el cliente web, que internamente llama a un cliente Biomoby, enviándole los datos de la enfermedad o del conjunto de genes, las bases de datos que quiere utilizar, indicando además los criterios que quiere seguir para ordenar los SNPs.

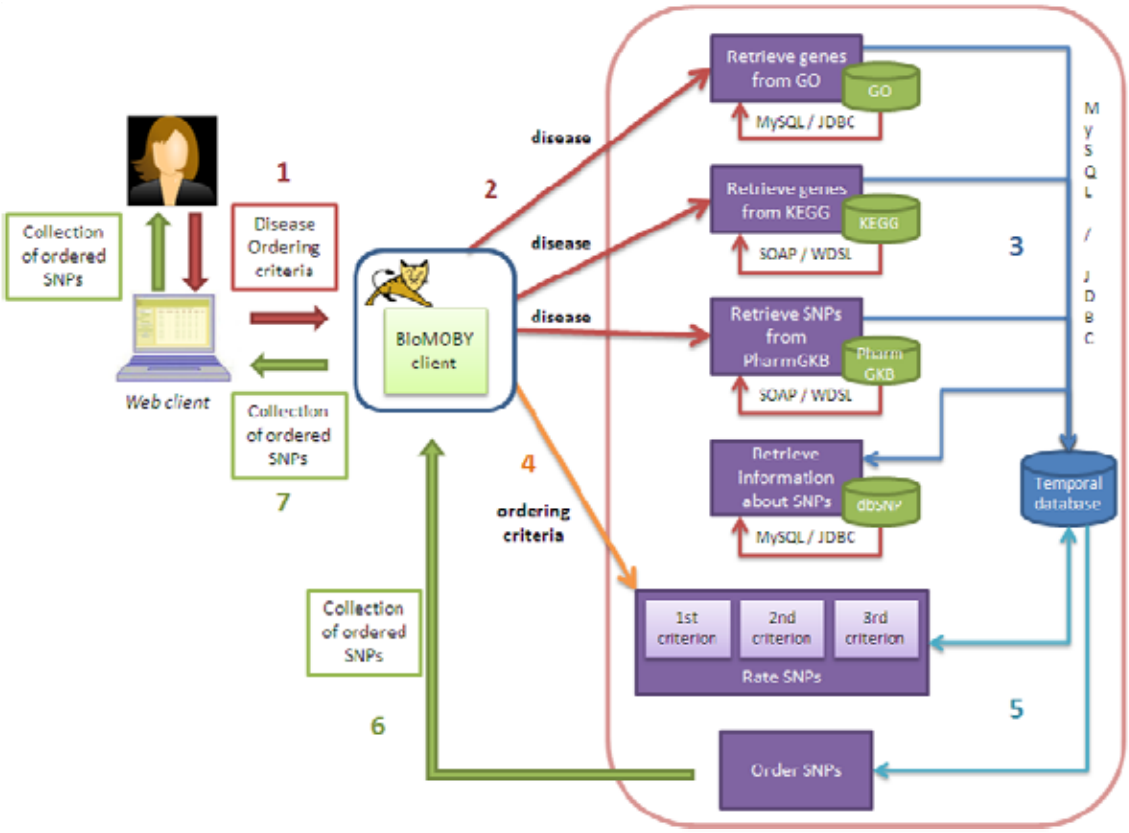


Figura 35: Ejemplo de ejecución escenario 2

Posteriormente, la aplicación web llama al cliente Biomoby, proporcionándole la enfermedad de búsqueda o el conjunto de genes y los criterios de ordenación. A partir de ahí se llama a los servicios Biomoby.

En este ejemplo concreto, la primera llamada se realiza a un servicio que recupera genes relacionados con una enfermedad a partir de la bases de datos de Gene Ontology a través de JDBC. Por motivos de rendimiento, se pueden almacenar los resultados en una base de datos temporal.

De manera similar, otros servicios realizan la misma búsqueda en las bases de datos KEGG y PharmGKB.

Otro servicio recupera los genes almacenados en la base de datos y a través de dbSNP recupera los SNPs asociados a esos genes, que pueden ser almacenados a su vez en la base de datos.

Ese mismo servicio también recupera información adicional de los SNPs, que se utilizará en la posterior fase de ordenación.

De acuerdo con los criterios indicados por el investigador, otro servicio puntúa los SNPs teniendo en cuenta los datos recuperados de dbSNP. Finalmente, el servicio devuelve una colección de SNPs ordenados por los criterios indicados.

El cliente Biomoby devuelve esa lista de clientes ordenados a la aplicación web que se los muestra al usuario.

Todos los servicios implementados para este sistema se han registrado como servicios Biomoby, es por ello que son accesibles para cualquier investigador a través del repositorio central Moby. Por lo tanto, estos servicios pueden usarse a través de un cliente Biomoby y utilizarse o integrarse con otros servicios distintos a través de otros clientes Biomoby, como puede ser Taverna (Hull et al., 2006).

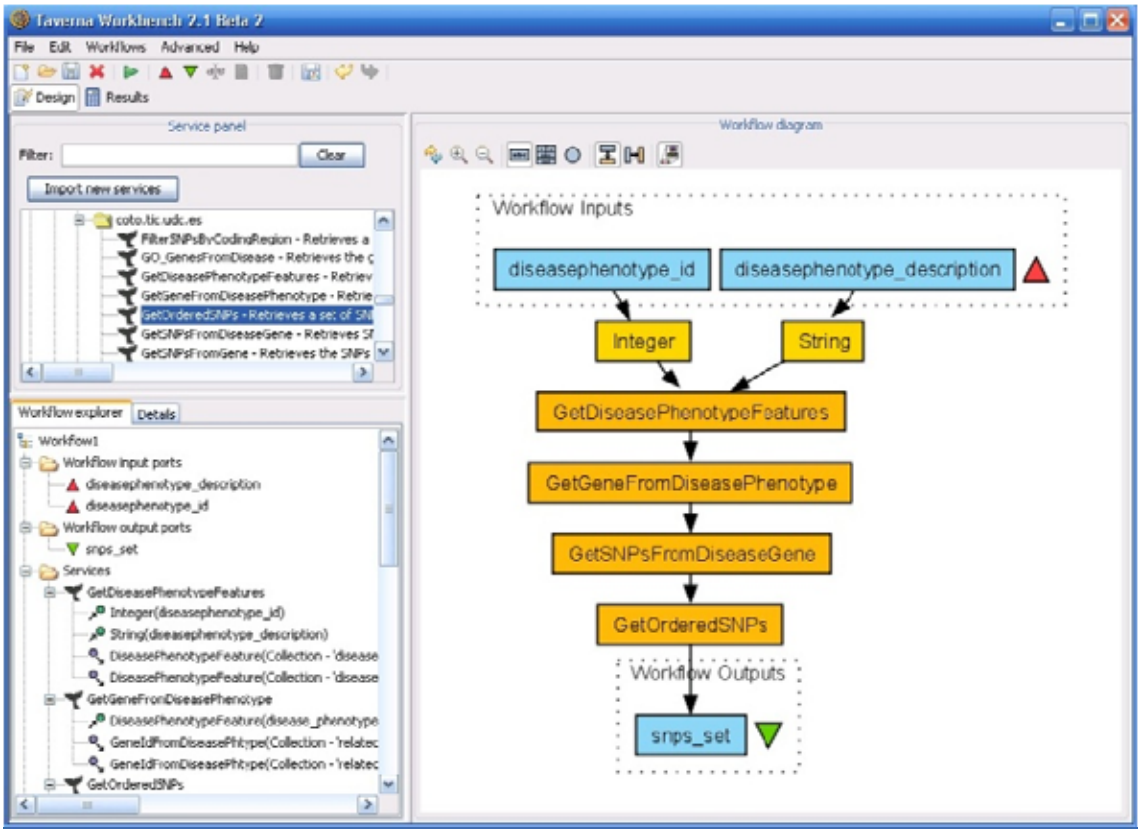


Figura 36: Ejecución del workflow en Taverna

En la Figura 36 se muestra un ejemplo de la utilización de Taverna para probar algunos de los servicios que se desarrollaron de manera independiente.

A continuación se muestra un ejemplo de ejecución del sistema, utilizando los siguientes parámetros:

- Como enfermedad se introdujo “glioma”
- Se escogió la búsqueda de genes en Gene Ontology y KEGG
- Como tipo de SNP, se escogieron “all types of SNPs”
- Se seleccionaron sólo los “tag SNP”

Los resultados de la consulta se pueden ver en la Figura 37. Para cada uno de los SNPs se puede ver el identificador del SNP, el cromosoma donde se sitúa, el nombre del gen, si es o no una región codificante, su frecuencia y su puntuación.

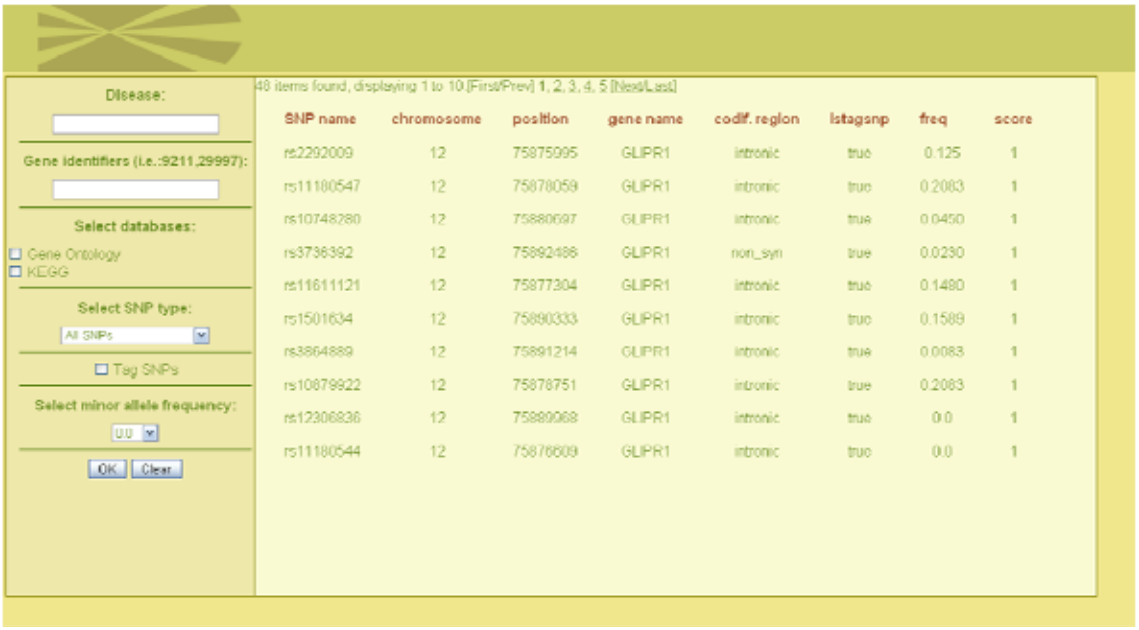


Figura 37: Resultados de la consulta

Con el objetivo de validar el sistema desarrollado, el laboratorio de Epidemiología Molecular de Enfermedades Infecciosas del Centro Nacional de Microbiología dependiente del ISCIII, ha probado el sistema para el diseño de un estudio de asociación con el objetivo de determinar la influencia de distintos SNPs en la aparición de sepsis y su pronóstico.

7.2.5 Discusión

El desarrollo de este sistema permite probar la efectividad del marco metodológico propuesto en esta tesis en un campo relacionado con la integración de datos para la relación entre genotipo-fenotipo. La posibilidad de usar *workflows* con Biomoby para desarrollar este modelo posibilita la reutilización individual de cada uno de los servicios o de conjuntos de *workflows*, así como la utilización de servicios

Biomoby externos para enriquecer la herramienta, tanto en tareas de recuperación de datos, como en tareas de análisis.

En este sistema toda la información se recupera de bases de datos externas o servicios web que acceden a ellas, utilizando una arquitectura federada. Esta aproximación tiene una ventaja muy importante, y es que la información recuperada está siempre actualizada, ya que las fuentes de datos originales son las responsables de mantener esta información al día.

Esta aproximación presenta también ciertas desventajas, como son la disponibilidad y el rendimiento. En primer lugar, si una fuente de datos no está disponible, no será posible recuperar la información que contiene. En las pruebas realizadas, este problema no apareció muy a menudo, ya que las fuentes se seleccionaron cuidadosamente, escogiendo aquellas que presentaban una mejor disponibilidad, son ampliamente conocidas y usadas por la comunidad. A pesar de este problema, el sistema maneja los errores de disponibilidad de alguna de las fuentes omitiendo esa información en la integración final. Debido a esto, si una fuente no está disponible temporalmente, los resultados obtenidos no incluyen información de esta fuente, pero la obtendrán del resto. En lo que se refiere al rendimiento, estos fallos de disponibilidad no suponen una bajada de rendimiento significativa, ya que el sistema intenta conectarse a cada fuente, pero si esta fuente no está disponible, el sistema continuará normalmente, procesando las otras fuentes.

Otro de los problemas de la aproximación seleccionada se refiere al tiempo de respuesta, ya que implica que el tiempo de acceso depende del tiempo de acceso a cada una de las fuentes a las que se accede de manera remota. La alternativa a esta opción utilizando este mismo modelo hubiese pasado por tener una copia local de cada una de las fuentes de datos, pero esto tendría como desventaja, sobre el rendimiento, la recuperación de datos no actualizados. Teniendo en cuenta que la opción escogida es la primera, el tiempo de respuesta de los servicios implementados depende básicamente del tiempo consumido por las fuentes de datos remotas durante el proceso de consulta. En general, los servicios a los que se accede por medio de acceso directo a la base de datos son más rápidos que aquellos que usan servicios web.

7.3 Sistema de integración de pacientes con trasplante de riñón

7.3.1 Presentación del caso

El trasplante de riñón ha extendido y mejorado la calidad de vida de la mayoría de los pacientes en la fase final de enfermedad renal. La mayoría de los trasplantes implican diferencias genéticas importantes entre el donante y el enfermo.

Como consecuencia de esta diferencia se produce una respuesta inmune del receptor al órgano que produce un rechazo. Para evitar esto, se le administran al donante, durante el resto de su vida, potentes fármacos inmunosupresores. Este tipo de fármacos deprimen el sistema inmune anfitrión de manera general, lo que provoca diversos efectos colaterales.

Por lo tanto, uno de los mayores desafíos en el trasplante de riñón es equilibrar la necesidad de inmunosupresión para prevenir el rechazo del nuevo órgano, mientras se minimiza a toxicidad de los fármacos proporcionados para la inmunosupresión.

El suministro a un paciente de un fármaco de inmunosupresión u otro puede variar significativamente su pronóstico. No hay que olvidar que la administración de fármacos tiene una importante tasa de mortalidad y morbilidad, debido a que generalmente, se presupone que el mismo fármaco afectará de la misma manera a un paciente que a otro.

La farmacogenética estudia la base genética de cómo el cuerpo responde a cierto fármaco. Desde el proceso de administración del fármaco hasta que llega al objetivo, éste pasa por diferentes fases (absorción, metabolización, transporte, degradado, etc.). El perfil genético de cada persona puede variar en alguna de estas fases y esto puede provocar efectos distintos en cada individuo.

Pero el estudio de los factores genéticos que pueden predisponer a un mejor pronóstico con respecto a la administración de inmunosupresores requiere de la integración de diversas bases de datos, que incluyan tanto datos clínicos como datos genéticos.

Este caso de prueba consiste en el análisis por medio de un sistema de integración de datos de un estudio de Flechner (Flechner et al., 2004), en el cual se intentarán responder las siguientes preguntas:

- ¿Con qué funciones están relacionados los genes que están infra o sobre expresados cuando existe daño causado por inmunosupresores?
- ¿Qué genes relacionados con enfermedades mendelianas aparecen infra o sobre expresados en pacientes tratados con inmunosupresores? Esto permitirá observar los genes relacionados con enfermedades mendelianas relacionadas con el riñón que tienen sobre o infra expresión en pacientes con rechazo.

Con el objetivo de probar la efectividad del marco metodológico expuesto en esta tesis, se desarrolló un sistema de información que permita integrar las diversas fuentes de datos necesarias para estudiar las respuestas farmacogenéticas a fármacos inmunosupresores para evitar el rechazo a un riñón trasplantado.

7.3.2 Análisis

7.3.2.1 Análisis de requisitos

A partir de un conjunto de bases de datos genéticas y clínicas, un usuario puede recuperar un conjunto de información interrelacionada a través de consultas complejas por medio de un sistema sencillo.

El usuario podrá crear las consultas a partir de las fuentes de datos disponibles y almacenarlas, de manera que pueda replicarlas o modificarlas de manera sencilla.

Los resultados deben de presentarse de manera que sea sencillo realizar una rápida observación de ellos vía web y, además, que sea fácil trabajar con ellos posteriormente, es decir, recuperarlos en XML o en un formato de base de datos.

Basándose en los requisitos del sistema de integración, se ha evaluado la importancia de los factores de decisión ordenados por decisiones clave.

En lo que concierne a la arquitectura (Tabla 31), la inversión inicial tiene una importancia relativa, ya que si bien no es fundamental, tiene importancia en la

implementación de la solución. La actualización de los resultados tiene una importancia alta debido a que el sistema debe tener los datos actualizados. Por otro lado, la disponibilidad también es importante, ya que, debido a la complejidad del escenario, la falta de una de las bases de datos podría hacer que no funcionase correctamente el sistema. El escalado y el rendimiento, siendo importantes, no son determinantes para el escenario descrito, por lo que se les asigna una importancia media. El control de datos tiene una importancia baja debido a que son, en su totalidad, datos públicos. Finalmente, una vez más, la capacidad de evolución del modelo es determinante, por lo que tiene una importancia alta.

En lo que respecta a la inclusión de nuevas fuentes, debido a la complejidad del sistema y a la posibilidad de ampliación que se espera, el coste por fuente en desarrollo será de una importancia media y la importancia del coste de añadir fuente posteriormente será alta.

Tabla 31: Importancia de los factores de decisión en arquitectura escenario 3

Arquitectura	
	Importancia
Inversión inicial	Media
Actualización	Alta
Escalado datos	Media
Rendimiento	Media
Disponibilidad	Alta
Control datos	Baja
Evolución modelo	Alta
Coste por fuente	Media
Coste por fuente añadida	Alta

En lo que respecta a la importancia de los factores de decisión relativos al modelo de integración (Tabla 32), los requisitos del escenario indican que el manejo por parte de un usuario no experto es de una importancia alta. Sin embargo el acceso directo a las fuentes originales no es necesario.

Tabla 32: Importancia de los factores de decisión en modelo de integración en escenario 3

Integración	
	Importancia
Usuario no experto	Alta
Acceso fuentes	Baja

En lo que respecta a la estrategia de integración (Tabla 33), como se vio anteriormente, la evolución del modelo tiene una importancia alta, de la misma manera que el acceso a numerosas bases de datos. Sin embargo, el acceso de escritura a las bases de datos no es necesario, por lo que tiene una importancia baja.

Tabla 33: Importancia de los factores de decisión en estrategia de integración en escenario 3

Estrategia	
	Importancia
Nº Bases datos	Alta
Acceso escritura	Baja
Evolución Modelo	Alta

En lo que respecta a los requisitos de integración semántica (Tabla 34), el sistema debe poder hacer frente a registros repetidos, tener en cuenta la granularidad de los datos y debe ser capaz de afrontar integración horizontal, por lo que la importancia de estos factores es alta.

Tabla 34: Importancia de los factores de decisión en tipo de integración semántica en escenario 3

Integración semántica	
	Importancia
Registros repetidos	Alta
Granularidad	Alta
Integración horizontal	Alta

Finalmente, de los factores de decisión para la gestión de consultas (Tabla 35), como se vio anteriormente, la evolución del modelo tiene una importancia alta, mientras que dada la complejidad del sistema entero, la facilidad de implementación tiene una importancia menor.

Tabla 35: Importancia de los factores de decisión en gestión de consultas en escenario 3

Consultas	
	Importancia
Evolución modelo	Alta
Facilidad implementación	Baja

7.3.2.2 Análisis de casos de uso

En el sistema existen dos actores principales, por un lado el usuario, que es el encargado de realizar las consultas, construirlas a partir de los atributos disponibles en la base de datos y gestionarlas. Por otro lado está el administrador, que es el encargado de gestionar y adaptar los orígenes de datos al sistema.

Los casos de uso son los siguientes:

Gestionar consultas: El usuario puede gestionar sus propias consultas a través de la interfaz del sistema de información. Esta gestión permite crear de una manera sencilla nuevas consultas o eliminar y visualizar las ya creadas y almacenadas, pudiendo variar los parámetros de valor de las mismas para guardarlas o ejecutarlas.

Búsqueda de consultas: El sistema permite al usuario llevar a cabo búsquedas de consultas que se han almacenado previamente, en base a distintos parámetros (nombre, contenido, anotaciones, creador, etc.).

Ejecución de consultas: El usuario puede ejecutar las consultas creadas previamente en el sistema.

Gestión de fuentes de datos: El administrador puede gestionar las fuentes de datos que el sistema debe integrar. Permitirá agregar o eliminar fuentes de datos y extraer metadatos para poder interactuar con ellas.

Gestión de integración: El administrador podrá gestionar las interconexiones entre las diferentes fuentes de datos para realizar el proceso de integración.

7.3.2.3 Análisis de datos a integrar

La primera fuente de datos que se utilizará serán datos genéticos obtenidos de los pacientes de Cleveland Clinic Fundation sobre un microarray HG_U95Av2 (Affymetrix Human Genome U95 versión 2) de Affymetrix GeneChip (Santa Clara, CA), estos datos están publicados en la base de datos GEO (Gen Expression Omnibus) (GEO, 2004), dataset GDS724. Los datos se refieren a donantes y a diversos tipos de pacientes, que incluyen pacientes con rechazo agudo confirmado por biopsia, pacientes sin rechazo después de un año del trasplante y pacientes con disfunción renal pero sin rechazo. Los datos se obtuvieron, o por medio de biopsia o por medio de análisis de linfocitos periféricos o por ambos.

Por otro lado, serán necesarios también los datos clínicos referidos a la situación y tratamiento de cada paciente de donde proceden los datos genéticos. Dichos datos pueden encontrarse también en el *dataset* del GEO. Consisten en edad, sexo, tratamiento anti-rechazo (ciclosporina, micofenolato mofetilo, predinsona, tacrolimus y sirolimus), estado del paciente (cadáver o dotante vivo), nivel de creatinina en suero en mg/dL, días desde el trasplante e histopatología según el rechazo (estadio A según la escala de Banffl (Racusen et al., 1996), límite de rechazo o normal) o según la disfunción renal (inhibición de calcineurina, necrosis tubular aguda o glomeruloesclerosis focal y segmentaria).

La información sobre enfermedades asociadas con la sobre o infra-expresión de algún gen fue obtenida de la base de datos OMIM (OMIM, 2011), que alberga información genética sobre enfermedades mendelianas.

Finalmente, es necesaria una base de datos con la que obtener las funciones con las que se relacionan los genes. Estos datos se obtendrán de la base de dato Gene Ontology (GO) (GO, 2011).

7.3.2.3.1 Contenido de los datos

Las bases de datos escogidas, de donde se obtendrán los datos, son altamente conocidas y fiables, soportadas por entidades como el GO Consortium o el NCBI, por lo que no suponen un problema la calidad y disponibilidad.

7.3.2.3.2 Interfaz

El NCBI y el EBI ponen a disposición las bases de datos GO y OMIM a través de servicios web, por lo que este será el interfaz que se utilizará para acceder a este tipo de bases de datos. Por otro lado, la base de datos con los datos de expresión está disponible sólo por FTP, por lo que es necesario bajarla a local. Esto datos están almacenados en forma de texto plano, por lo que será necesario un paso previo de almacenamiento de los datos de GEO en una base de datos relacional.

7.3.2.3.3 Actualizaciones

Cualquiera de estas fuentes de datos es suficientemente conocida como para asegurar su mantenimiento. Los datos recuperados de GO y OMIM son recuperados directamente de las fuentes, con lo que están actualizados. Los datos de GEO se refieren a estudios ya realizados por lo que se espera que no se modifiquen.

7.3.2.3.4 Modelo de datos y anotación semántica

Tanto los datos de GO como de OMIM presentan un modelo de datos bien definido y completamente anotado. De hecho, Gene Ontology es una ontología en sí, por lo que su modelo de datos está bien definido y bien anotado. El modelo de datos de los datos genéticos y de los datos clínicos de GEO es simple y puede ser estructurado fácilmente. Únicamente presenta cierta dificulta el hecho de que los datos genéticos se referencian en función del spot del microarray, con lo cual es necesario acceder a otra tabla de referencia del microarray para conocer el gen exacto.

7.3.2.3.5 Clasificación Busse

Según el modelo de Busse, el análisis de las fuentes de datos será el siguiente:

- Autonomía: Cada una de las bases de datos utilizadas pertenece a organismos internacionales, con lo cual tienen una total autonomía de diseño, de comunicaciones y de ejecución, salvo en el caso de la base de datos de GEO que, una vez instalada en local, pierde su autonomía a favor del usuario.
- Heterogeneidad
    - Sintáctica tecnológica y de interfaz: Las bases de datos de GO y de OMIM permiten el acceso a través de servicios web mientras que la base de datos de GEO está en un fichero plano y es accesible sólo

mediante FTP, pero al instalarla en local puede ser utilizada con la tecnología que se requiera.

  - Sintáctica de consulta: Las bases de datos GO y OMIM no permiten acceder a la totalidad del modelo a través de los servicios web, sin embargo el acceso ofrecido es suficiente para las necesidades del sistema a desarrollar. La base de datos GEO está en local por lo que ofrece la totalidad del modelo
  - De modelo y lógica: El modelo de datos de GO es un modelo basado en una ontología y su modelo es un grafo dirigido acíclico (GeneOntologyConsortium, 2001). El modelo de OMIM es originalmente un modelo que contiene dos entidades, una donde se representan los genes/variaciones y otra donde se representan las enfermedades. Finalmente el modelo GEO consiste en una entidad de muestra/paciente y otra de gen. Como se ha expresado anteriormente, en los dos primeros casos no es útil conocer el modelo ya que se oculta a través de la consulta.

- Distribución: Las bases de datos están localizadas en el NCBI (OMIM y GEO) y EBI (GO).

En definitiva, los datos que será necesario integrar para responder a las preguntas que se plantearon en el apartado anterior sobre este sistema son datos complejos, implican la integración de información clínica proveniente de una base de datos de pacientes con información genética (genes y sobre o infraexpresión) de esos mismos pacientes, que debe ser asociada por un lado a la función que realizan esos genes y a las enfermedades a las que están asociados.

Es por lo tanto necesario para realizar el proceso de integración, tener en cuenta asociaciones complejas como, por ejemplo, asociar el concepto de infra o sobre expresión de un gen a un tipo concreto de función y, esa función o gen, con una patología.

7.3.2.4 Selección de arquitectura

En primer lugar se descarta para el desarrollo de este sistema un sistema de navegación por links como arquitectura de integración, debido a que la finalidad es obtener un gran número de resultados de manera automática y la navegación por links es más adecuada para consultas puntuales. Por otro lado, debido a la variabilidad de las fuentes y, dado que estas pertenecen a distintos organismos, tampoco es aconsejable este tipo de integración.

Tabla 36: Valoración arquitectura escenario 3

Arquitectura		**Federado**	**Warehouse**
	Importancia	Valor	Valor
Inversión inicial	Media	Medio	Alto
Actualización	Alta	Alto	Bajo
Escalado datos	Media	Bajo	Alto
Rendimiento	Media	Bajo	Alto
Disponibilidad	Alto	Bajo	Alto
Control datos	Baja	Alto	Alto
Evolución modelo	Alta	Alto	Bajo
Coste por fuente	Media	Alto	Bajo
Coste por fuente añadida	Alta	Bajo	Alto

Para la elección de un modelo de datos federado o un modelo de datos basado en *warehouse*, además de los factores de la Tabla 36 es necesario tener en cuenta la complejidad de integración y las fuentes de datos disponibles.

Después de realizar el análisis de los datos, se decide que la base de datos proveniente de GEO, que contiene la información clínica de los pacientes y de los genes sobre o infra expresados de estos, se despliegue de manera local, creando un esquema de base de datos y un modelo lo más acorde con los requisitos del sistema, mientras que las otras dos bases de datos, GO y OMIM, permanecerán en sus esquemas originales.

Al igual que en los escenarios anteriores, las valoraciones altas para el warehouse vienen dadas por el escalado de los datos, el rendimiento, la disponibilidad

y el control de datos, además de un coste moderado de inclusión de fuentes en diseño, mientras que las altas para el federado son la actualización, el control de datos, un coste moderado de inversión inicial y sobre todo la posibilidad de evolución del modelo y un coste moderado de inclusión de fuentes nuevas en etapas posteriores al desarrollo.

7.3.2.5 Tipo de modelo de integración

La Tabla 37 resume la importancia de los factores para cada una de las opciones de modelo de integración.

Tabla 37: Valoración del modelo de integración en escenario 3

Integración		Abierta	Cerrada
	Importancia	Valor	Valor
Usuario no experto	Alta	Alto	Bajo
Acceso fuentes	Baja	Alto	Bajo
Coste por fuente	Media	Bajo	Alto
Coste por fuente añadida	Alta	Medio	Alto

La aplicación que se desea desarrollar está bastante bien definida, es un sistema de información que debe recuperar datos e integrarlos para mostrarlos al investigador. Desde el punto de vista de este usuario investigador, el valor de usuario no experto en el sistema cerrado debería de ser alto, ya que el usuario investigador no tiene responsabilidades de integración pero es quien hace uso del sistema.

Sin embargo, existe un usuario administrador, que es el que se tomará en consideración para este análisis, es el que crea las consultas y que podría eventualmente añadir otros tipos de datos al sistema, e incluso nuevas bases de datos. Debido a esto el valor de usuario no experto en el sistema abierto es alto y el acceso a las fuentes originales es alto.

7.3.2.6 Estrategia de integración

En este caso, el sistema puede requerir un número importante de bases de datos y la evolución del modelo pueden ser condicionante, como se deduce de la Tabla

38, en la que el valor de la estrategia *top-down* tiene un valor alto en número de bases de datos y en evolución del modelo.

Tabla 38: Valoración de estrategia de integración en el escenario 3

Estrategia		**Top-down**	**Bottom-up**
	Importancia	Valor	Valor
Nº Bases datos	Alta	Alto	Bajo
Acceso escritura	Baja	Bajo	Alto
Evolución Modelo	Alta	Alto	Bajo
Coste por fuente	Media	Medio	Alto
Coste por fuente añadida	Alto	Bajo	Alto

Llegados a este punto, se definirá el modelo final del proceso de integración y se buscará como obtener estos datos de las bases de datos originales.

Los datos necesarios en el punto final de integración son los siguientes:

- Genes: Nombre del gen, sobre o infra expresión, función.
- Tratamiento: Tratamiento que se aplicó al paciente.
- Estado clínico del paciente: Estado del paciente, si es trasplantado, días desde el trasplante.
- Fenotipo: Enfermedad mendeliana relacionada con el riñón.

7.3.2.7 Tipos de integración

Utilizando los requisitos, se deduce que el tipo de integración que se debe realizar en este piloto es un ejemplo claro de integración horizontal, que tiene un valor alto en suplementación. En la Tabla 39 se muestra un resumen de la valoración de los factores.

Los niveles de datos que se está tratando presentan, o pueden presentar, cierta diferencia en la granularidad, por lo que se podrían modelar con una abstracción, sin embargo, el hecho de que sea necesario completar los datos de los spots del microarray, con los datos de los genes provenientes del tipo de microarray, y estos a su vez, con las funciones de los genes provenientes de GO, implica que sería más aconsejable modelarlo con una suplementación.

Tabla 39: Valoración de tipos de integración semántica en el escenario 3

Integración semántica		Agregación	Fusión	Abstracción	Suplementación
	Importancia	Valor	Valor	Valor	Valor
Registros repetidos	Alta	Bajo	Alto	Alto	Alto
Granularidad	Alta	Bajo	Bajo	Alto	Alto
Integración horizontal	Alta	Bajo	Bajo	Bajo	Alto
Coste por fuente	Media	Bajo	Medio	Alto	Alto
Coste por fuente añadida	Alta	Bajo	Medio	Alto	Alto

Al utilizar la suplementación es necesario dotar al sistema de algún tipo de mecanismo de anotación semántica de los datos, por lo que será necesario incorporar ontologías al desarrollo.

7.3.2.8 Procesado, planificación y ejecución de consultas

En la Tabla 40 se muestran las valoraciones de los modelos de acceso a datos. La evolución del LaV tiene un valor alto, mientras que la facilidad de implementación un valor bajo, tal y como se vio en los otros escenarios.

El coste de implementación de fuentes en el LaV es relativamente alto con respecto al de GaV, pero una vez desarrollado el sistema, el coste de añadir nuevas fuentes en GaV es mucho mayor que el LaV.

Tabla 40: Valoración de la gestión de consultas en el escenario 3

Consultas		LaV	GaV
	Importancia	Valor	Valor
Evolución modelo	Alta	Alto	Bajo
Facilidad implementación	Baja	Bajo	Alto
Coste por fuente	Media	Alto	Bajo
Coste por fuente añadida	Alta	Bajo	Alto

7.3.2.9 *Métrica y costes*

En este paso se evalúa el impacto de las decisiones tomadas teniendo en cuenta los factores de decisión clave y la valoración de estos para cada opción.

Tabla 41: Impacto arquitectura escenario 3

Arquitectura		**Federado**		**Warehouse**	
	Importancia	Valor		Valor	
Inversión inicial	Media	Medio	Medio	Alto	Medio
Actualización	Alta	Alto	Positivo	Bajo	Negativo
Escalado datos	Media	Bajo	Medio	Alto	Medio
Rendimiento	Media	Bajo	Medio	Alto	Medio
Disponibilidad	Alta	Bajo	Negativo	Alto	Positivo
Control datos	Baja	Alto	Medio	Alto	Medio
Evolución modelo	Alta	Alto	Positivo	Bajo	Negativo
Coste por fuente	Media	Alto	Medio	Bajo	Medio
Coste por fuente añadida	Alta	Bajo	Positivo	Alto	Negativo

Las necesidades de integración expuestas en los requisitos podrían sugerir el uso de un esquema de integración por *warehouse*, debido a que las consultas son complejas, a los valores de impacto positivo en escalado, rendimiento y disponibilidad (Tabla 41), y podría ser útil generar un modelo común local, sin embargo, a pesar de que la base de datos que se utiliza para este sistema tiene unas características en cuanto a número de pacientes y aspectos clínicos reducidos, estos podrían ampliarse para realizar estudios semejantes con mayor número de pacientes y mayor cantidad de datos. Esto obligaría a cambiar significativamente el modelo.

En una arquitectura *warehouse*, el modelo es muy rígido, es escogido ad-hoc y la totalidad de los datos se transforman al modelo, con lo cual una vez escogido este, cualquier cambio, como añadir nuevas variables clínicas, implica el cambio del modelo. Por otro lado, la inclusión de nuevos pacientes implicaría la transformación de estos datos al modelo para su almacenamiento y posterior análisis dentro del warehouse, que se refleja en los valores de impacto negativo en evolución de modelo.

En la etapa de selección de arquitectura, una vez más la actualización de las fuentes, la disponibilidad de estas y la evolución del modelo, tanto actualizándolo como añadiendo nuevas fuentes son los factores clave de cara a la elección correcta, tal y como muestran los valores positivos y negativos de la Tabla 41. Teniendo en cuenta esto, especialmente los puntos que tienen que ver con la evolución del modelo, se ha optado por la opción del modelo federado, asumiendo la penalización por la disponibilidad de los datos. Este posible lastre intenta minimizarse con una cuidada selección de las fuentes y el acceso en local de algunas de bases de datos.

Tabla 42: Impacto modelo de integración escenario 3

Integración		**Abierta**		**Cerrada**	
	Importancia	Valor		Valor	
Usuario no experto	Alta	Alto	Positivo	Bajo	Negativo
Acceso fuentes	Baja	Alto	Medio	Bajo	Medio
Coste por fuente	Media	Bajo	Medio	Alto	Medio
Coste por fuente añadida	Alta	Medio	Medio	Alto	Negativo

En el caso de la elección del modelo de integración, la opción de que el usuario experto pueda añadir nuevas fuentes de datos, incrementar el modelo y realizar consultas complejas, teniendo en cuenta todos estos factores, prima en la selección del modelo abierto (Tabla 42). Del mismo modo que en caso de estudio número 2, se considerará una aproximación semi-abierta, porque se proporcionará un interfaz web para que el usuario final pueda simplemente ejecutar las consultas. Desde el punto de vista de ese usuario final el sistema sería cerrado.

Tabla 43 Impacto estrategia de integración escenario 3

Estrategia		**Top-down**		**Bottom-up**	
	Importancia	Valor		Valor	
Nº Bases datos	Alta	Alto	Positivo	Bajo	Negativo
Acceso escritura	Baja	Bajo	Medio	Alto	Medio
Evolución Modelo	Alta	Alto	Positivo	Bajo	Negativo
Coste por fuente	Media	Medio	Medio	Alto	Medio
Coste por fuente añadida	Alto	Bajo	Positivo	Alto	Negativo

Como se ha visto, los requisitos del sistema definen un modelo global por medio de entidades biológicas genéricas que definen perfectamente los tipos de datos necesarios y no es necesario incluir todos los datos presentes en todas las bases de datos que se están integrando. Los requisitos (Tabla 43) fijan también la importancia de que el modelo pueda evolucionar y los aumentos de coste si este modelo es *bottom-up*. Por ese motivo en este caso no hay duda de que la estrategia debe ser *Top-down*.

Tabla 44: Impacto integración semántica escenario 3

Integración semántica		**Agregación**		**Fusión**	
	Importancia	Valor	Impacto	Valor	Impacto
Registros repetidos	Alta	Bajo	Negativo	Alto	Positivo
Granularidad	Alta	Bajo	Negativo	Bajo	Negativo
Integración horizontal	Alta	Bajo	Negativo	Bajo	Negativo
Coste por fuente	Media	Bajo	Medio	Medio	Medio
Coste por fuente añadida	Alta	Bajo	Positivo	Medio	Medio

Integración semántica		**Abstracción**		**Suplementación**	
	Importancia	Valor	Impacto	Valor	Impacto
Registros repetidos	Alta	Alto	Positivo	Alto	Positivo
Granularidad	Alta	Alto	Positivo	Alto	Positivo
Integración horizontal	Alta	Bajo	Negativo	Alto	Positivo
Coste por fuente	Media	Alto	Medio	Alto	Medio
Coste por fuente añadida	Alta	Alto	Negativo	Alto	Negativo

En cuanto a la selección del modelo de integración semántica, la tabla muestra claramente que es necesaria como mínimo una integración horizontal (suplementación) que contemple la integración de elementos repetidos y el control de la granularidad. Si bien en escenarios anteriores se podían simultanear modelos de integración horizontal y vertical, en este caso el modelo horizontal será suficiente.

Tabla 45: Impacto gestión de consultas escenario 3

Consultas		LaV		GaV	
	Importancia	Valor	Impacto	Valor	Impacto
Evolución modelo	Alta	Alto	Positivo	Bajo	Negativo
Facilidad implementación	Baja	Bajo	Medio	Alto	Medio
Coste por fuente	Media	Alto	Medio	Bajo	Medio
Coste por fuente añadida	Alta	Bajo	Positivo	Alto	Negativo

Tal y como se decidió al escoger una aproximación *top-down*, las relaciones entre las fuentes de datos y el esquema global se realizan desde el punto de vista del usuario, es decir, del esquema global y no desde el punto de vista de las fuentes. Por otro lado, es necesaria la capacidad de incluir nuevas fuentes de datos sin que se vea afectado el diseño del esquema global, tal y como muestra la Tabla 45. Por lo tanto, las consultas se realizarán contra vistas de las fuentes de datos. Es decir, se utilizará un esquema de planificación de consultas *Local-As-View* (LaV).

La planificación de las consultas utilizando este esquema requiere, en primer lugar, la definición de una consulta sobre el esquema global. Posteriormente, esta consulta debe descomponerse en consultas a las vistas de las fuentes de datos originales o de los mediadores intermedios en el caso de que los haya. Estas descomposiciones implican, tal y como se indicó anteriormente, la traducción de los términos generales a los términos específicos, de cada fuente de datos utilizando algún tipo de ontología o regla de transformación que puede implicar recuperar varios términos a partir de traducciones o relaciones dentro de la ontología.

7.3.2.10 Aspectos tecnológicos

7.3.2.10.1 Seguridad y privacidad de los datos

A pesar de que las bases de datos a las que se accede no proporcionan identificadores como el nombre o el número de paciente, sí que se muestran datos genéticos que eventualmente podrían identificar a algún paciente o donante. Es necesario tener en cuenta que, a pesar de que la base de datos de donde provienen

estos datos es pública (GEO), el sistema podría ampliarse con usuarios clínicos reales, por lo que tiene que presentar medidas de seguridad.

Es por ello que, teniendo en cuenta la normativa de protección de datos aplicable, será necesario securizar el sistema de información con las medidas necesarias para que sólo las personas autorizadas puedan acceder a estos registros.

Afortunadamente, la única base de datos que podría requerir securización es la proveniente del GEO, con lo cual no habría que hacer una gestión segura de dos fuentes distintas, ya que las otras bases de datos son públicas.

7.3.2.10.2 Framework de integración

Los requisitos en cuanto a la implementación del sistema de integración descrito comprenden una gran cantidad de características complejas que provocan que el desarrollo desde cero sea terriblemente costoso, es por eso que se ha buscado un *framework* que permita realizar la integración de datos con las características expuestas anteriormente.

El *framework* escogido fue el sistema OntoMediator-QIS-OntoServer desarrollado por el Dr. Luis Marenco del Yale Center for Medical Informatics de la Universidad de Yale (Marenco et al., 2004, Marenco et al., 2009) (Figura 38). Las características principales de este sistema es que permite que el modelo esté basado en *mapping* y que describa transformaciones de una base de datos en conceptos equivalentes, que son definidos mediante reglas. Por otro lado, suporta un modelo de consultas federado en tiempo de ejecución, que permite la ejecución de las consultas sobre un sub-esquema de cada una de las fuentes de datos, así como sobre otros sub-esquemas que representan un vocabulario controlado que representa conceptos, términos y relaciones entre conceptos. Finalmente, el modelo requiere que los conceptos sean asociados a categorías de conceptos o clases que almacenan descripciones de los atributos o propiedades de cada clase.

El Ontomediator se puede dividir en los módulos Ontology Server (OS) y Data source Server (DSS).

El DSS puede considerarse un *wrapper* de alto nivel que contiene los metadatos de las fuentes de información, que incluyen anotaciones textuales y relaciones con vocabularios estándar.

Finalmente el OS mantiene el contenido de una o más ontologías y sus mapeados a los metadatos de las bases de datos del DSS, con el objetivo de que estas sean usadas en la federación. Estas ontologías pueden ser tanto generales, como puede ser UMLS, como específicas de cada fuente de datos y pueden contener también información de mapeado relacionada con las fuentes de datos. Siguiendo este esquema, ciertos conceptos se mapean a clases en ciertas bases de datos mientras que otros conceptos se mapean a atributos o a instancias de clases, que son análogos a las filas en las tablas de las bases de datos.

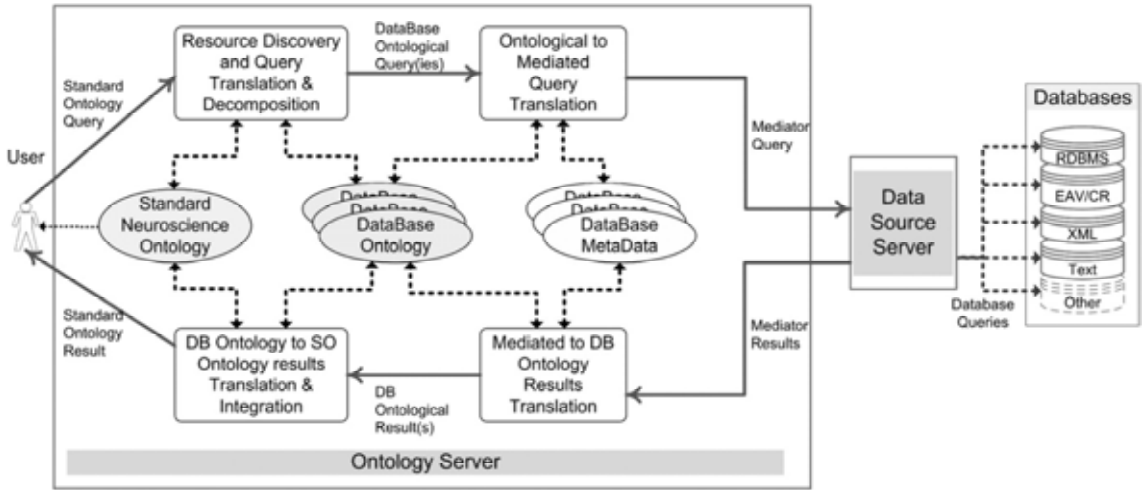


Figura 38: Ejemplo de aplicación del Ontology Server y del Data Source Server a un problema de neurociencia. Marenco et al. JAMIA 2009

7.3.3 Desarrollo

Utilizando por tanto el Ontomediator, se implementó un sistema para dar respuesta a las cuestiones planteadas anteriormente.

Con el objetivo de facilitar la recuperación de los datos y el rendimiento del sistema, se implementaron todas las bases en local.

La base de datos de GEO se implementó como estas tres entidades:

- Clinic_data: Contiene los datos clínicos de los pacientes.

- Genetic_data: Contiene los datos genéticos de cada uno de los pacientes.
- Microarray: Contiene los genes a los que se corresponde cada spot del microarray.

En primer lugar, fue necesario realizar la configuración de las fuentes de datos en el DSS (Figura 39). Este proceso consiste en añadir al repositorio del DSS la fuente de datos. El DSS admite diversos tipos de fuentes de datos: SQL, WebService, XML, etc.

Figura 39: Configuración de fuentes de datos en el DSS

Una vez escogida la fuente de datos, el DSS automáticamente extrae los metadatos de la base de datos (Figura 40), que consisten en el nombre de las tablas y los atributos. Es posible seleccionar una serie de campos (tablas, atributos), que el DSS entenderá como *términos preferidos*, y que serán los que posteriormente se utilizarán en el proceso de integración. De esta manera se descartan los términos de las bases de datos origen que no son útiles en el proceso de integración.

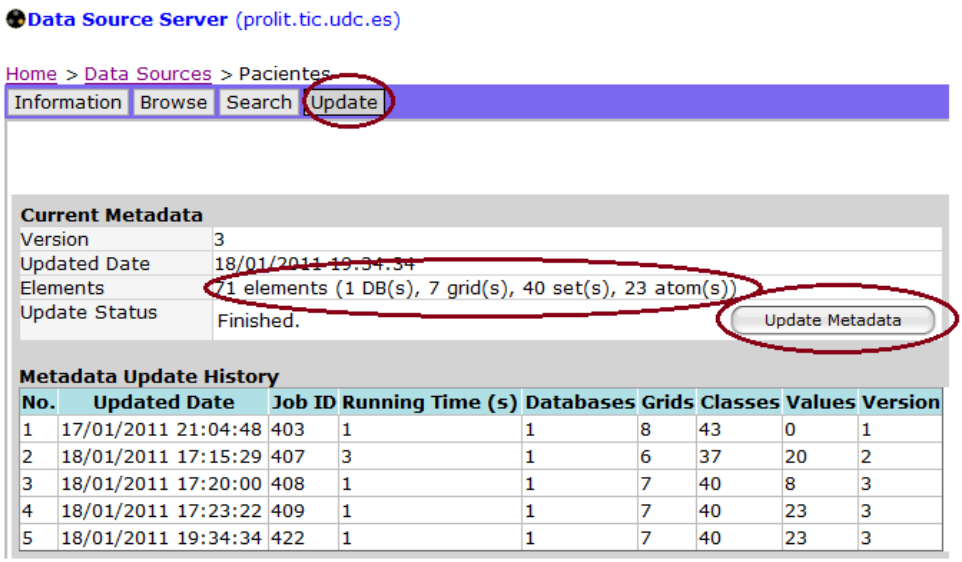


Figura 40: Extracción automática de los metadatos en el DSS

Una vez obtenidos los metadatos de las fuentes de datos, se desarrolla una ontología para cada una de las bases de datos involucradas en el sistema (Figura 41). Este paso es necesario para que el OS pueda establecer posteriormente relaciones semánticas entre los distintos tipos de datos.

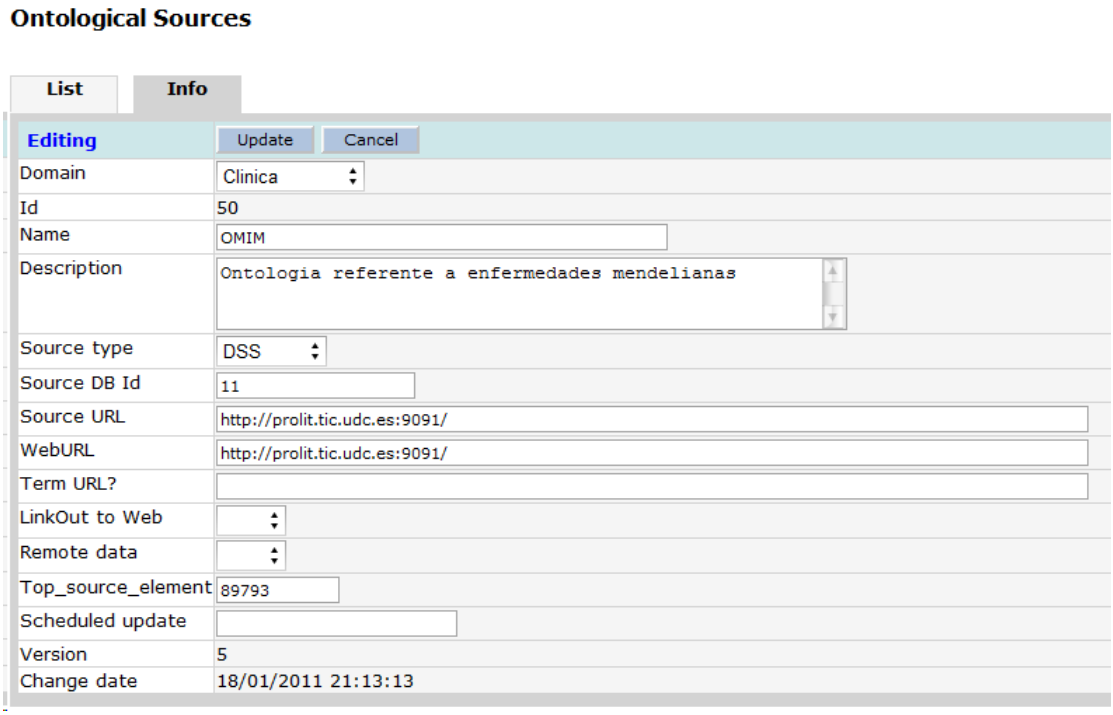


Figura 41: Configuración de la ontología de OMIM

Los campos generales con los que se configura la ontología son:

- Domain: Dominio (puede crearse uno ad-hoc) al que pertenezca la ontología.
- Name: Nombre de la ontología.
- Description: Breve descripción de la ontología.
- Source type: Este campo indica si la ontología obtiene información de los elementos del DSS o pertenece al grupo de las ontologías que se emplean para relacionar las distintas fuentes (tipo DSS) entre ellas.
- Source DB Id: Hace referencia al identificador de la base de datos del DSS sobre la cual se mapeará la ontología. Este identificador permite saber al OS cual es la fuente de datos sobre la que procesar las consultas y sobre la cual obtener los metadatos.
- Source URL: Servidor donde se encuentra el DSS.
- Remote data: Se marca *YES* si la base de datos de referencia de dicha ontología va a ser consultada. Se marca *NO* si la ontología no es tipo DSS.

A partir de los términos preferidos seleccionados en el DSS, se importan para construir la ontología propia.

Una vez realizado el import de los metadatos, los elementos marcados como términos preferidos aparecerán ahora como *term* en el dominio de la ontología.

Finalmente, cuando se han importado todos los términos de la ontología, se establecen las relaciones entre ellos en el *OntoEditor*.

Como se puede ver en la Figura 42, se pueden añadir subclases en cada término, de forma que las subclases se relacionarán con la superclase a través de la relación *IS_A*. Además, también permite establecer relaciones semánticas con otros términos contenidos en la ontología. En el caso de que no exista una relación específica, puede crearse.

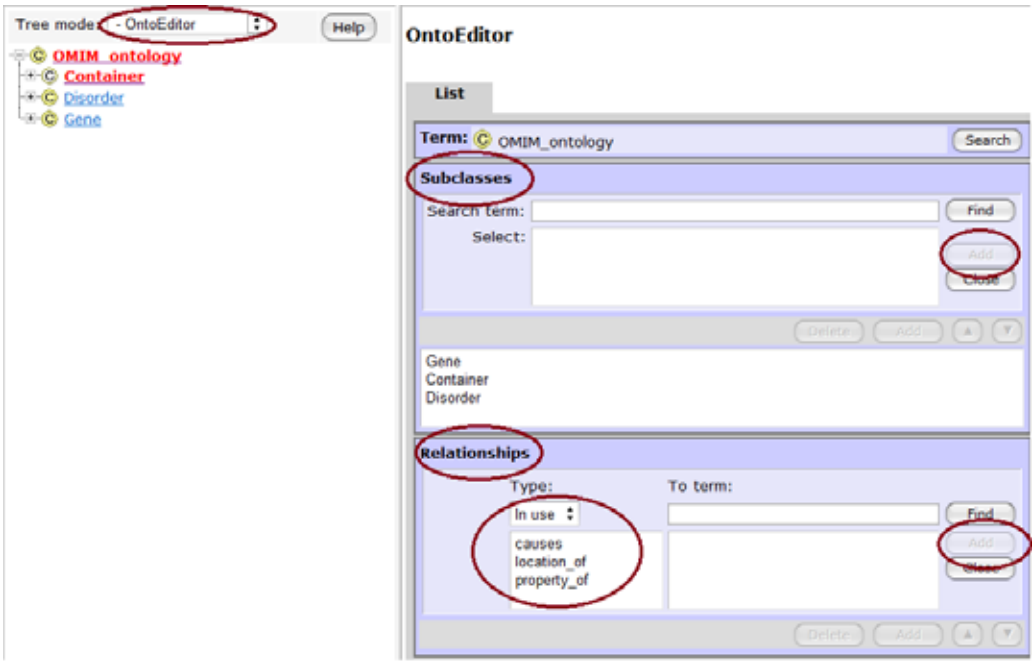


Figura 42: Edición de la ontología OMIM con el OntoEditor

Para que el OS pueda relacionar los términos de las ontologías, se definen elementos denominados *concept*. Todo término contenido en una ontología ha de ser mapeado a un único concepto. Estos conceptos son únicos para todas las ontologías.

Siguiendo este esquema se han creado las ontologías de las bases de datos OMIM (Figura 43).

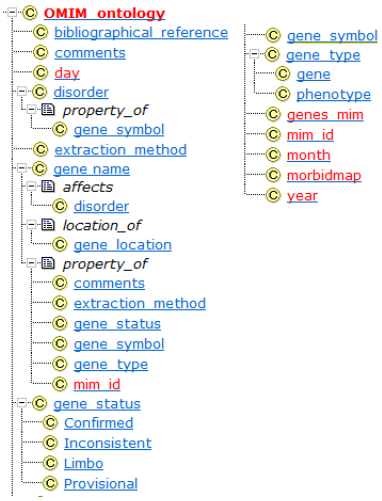


Figura 43: Esquema de la ontología OMIM

Al analizar el contenido de GEO y de GO, se decidió que, en lugar de implementar una fuente de datos para la base de datos de GEO y otra para la base de datos de GO, integrar estas dos fuentes de datos en un paso anterior. Esto es debido a que la relación de integración entre los genes que aparecen en GEO y sus términos dentro de GO era una integración simple que puede implementarse con una abstracción en lugar de con una suplementación. De esa manera se relacionaron todos los genes que aparecen en el estudio de GEO con su anotación GO antes de crear el dataset dentro del DSS. Posteriormente se desarrolló la ontología RTO que modela estas dos bases de datos GEO y GO que se puede ver en la Figura 44.

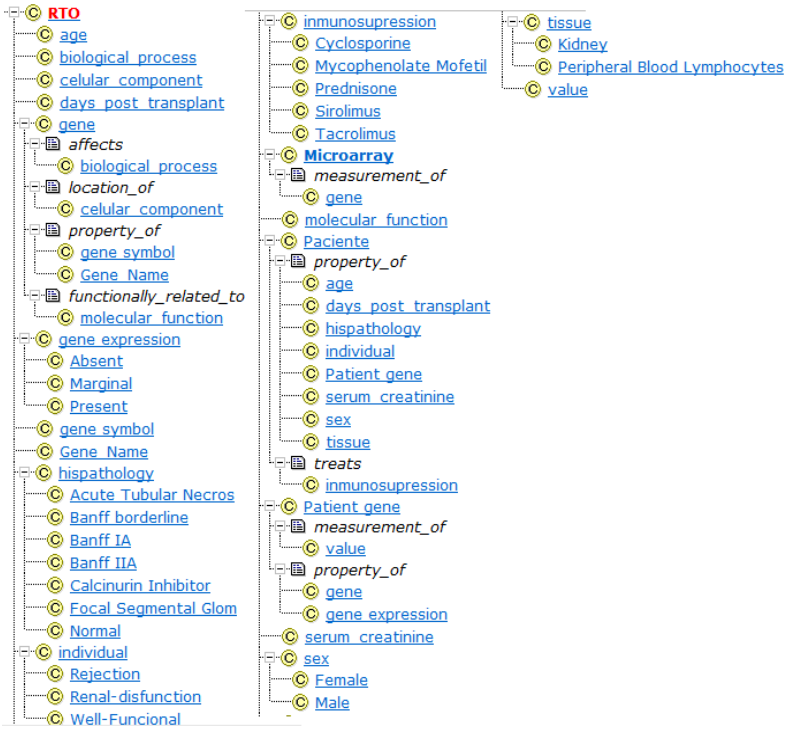


Figura 44: Esquema de la ontología RTO que modela GO y GEO

Este tipo de integración jerárquica en dos niveles simplifica el proceso, ya que el uso de anotaciones semánticas puede ser más costoso computacionalmente. De esta manera se simplifica el proceso de integración donde sea posible para reducir la complejidad computacional.

Posteriormente se desarrolló una ontología general siguiendo el modelo de datos desarrollado durante la definición del marco metodológico. Esta ontología, denominada RTO_general (Figura 45), es la empleada para integrar todas las fuentes

de datos. Los elementos que contiene esta ontología deben mapearse en los términos de las ontologías propias de cada fuente de datos.

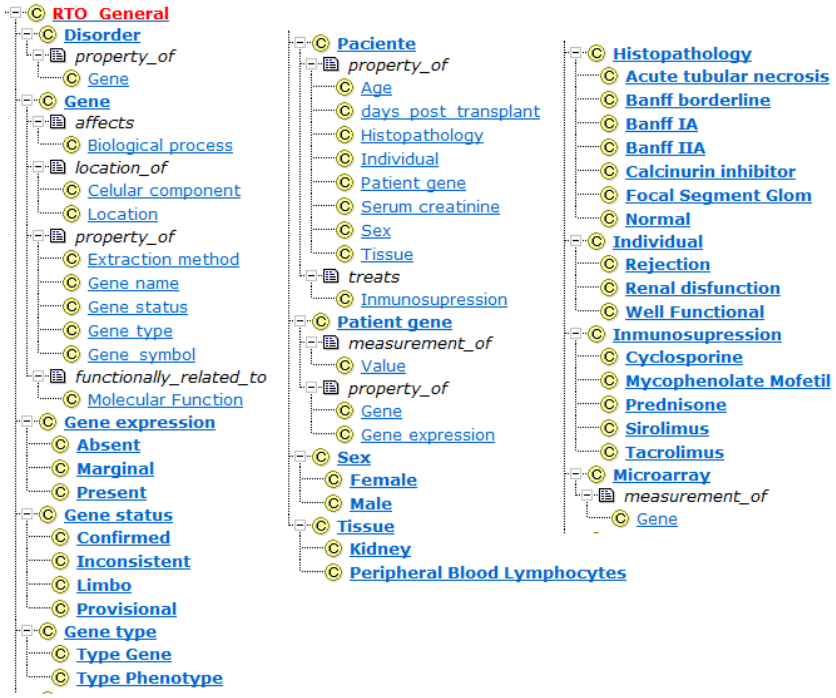


Figura 45: Ontología general (RTO)

El Ontology Server permite crear y gestionar las ontologías que el sistema emplea para definir las consultas semánticas y llevar a cabo los procesos de integración de los resultados.

El Ontology Server es el encargado de llevar a cabo la ejecución de las consultas. A continuación se describen los pasos que suceden al ejecutar las consultas e integrar los resultados (Figura 46):

1. Generar la consulta a través de los términos preferidos que aparecen en la ontología general como términos.
2. A partir de la consulta generada en el paso 1, y utilizando las reglas de asociación definidas anteriormente, se divide la consulta generando las subconsultas especificas para cada una de las ontologías propias de cada fuente de datos.
3. Adaptar la consulta a los términos específicos de cada fuente de datos. Este paso convierte los conceptos generales, definidos para la ontología que se emplea para llevar a cabo la consulta, en los términos específicos de cada sub-

ontología. Este paso permite que el OS convierta los términos de las sub-ontologías en sus respectivos campos dentro de la fuente de datos.

4. Convertir las consultas semánticas al formato DSS: El OS convierte las consultas semánticas en el formato que sea comprensible por el DSS.
5. Enviar las consultas para que sean ejecutadas por el DSS: Este paso envía las consultas a sus correspondientes fuentes de datos originales para que sean ejecutadas por el wrapper apropiado del DSS.
6. Recuperar y convertir los resultados al formato de consultas del OS: Este paso recupera los resultados de la consultas ejecutadas previamente y los convierte al formato de resultados del OS.
7. Traducir los resultados: Convierte los resultados empleando los conceptos de la ontología general que se usó para la consulta original.
8. Integración de los resultados: Este paso integra los resultados que se obtuvieron de las distintas fuentes de datos en función de la estructura de la consulta planteada.

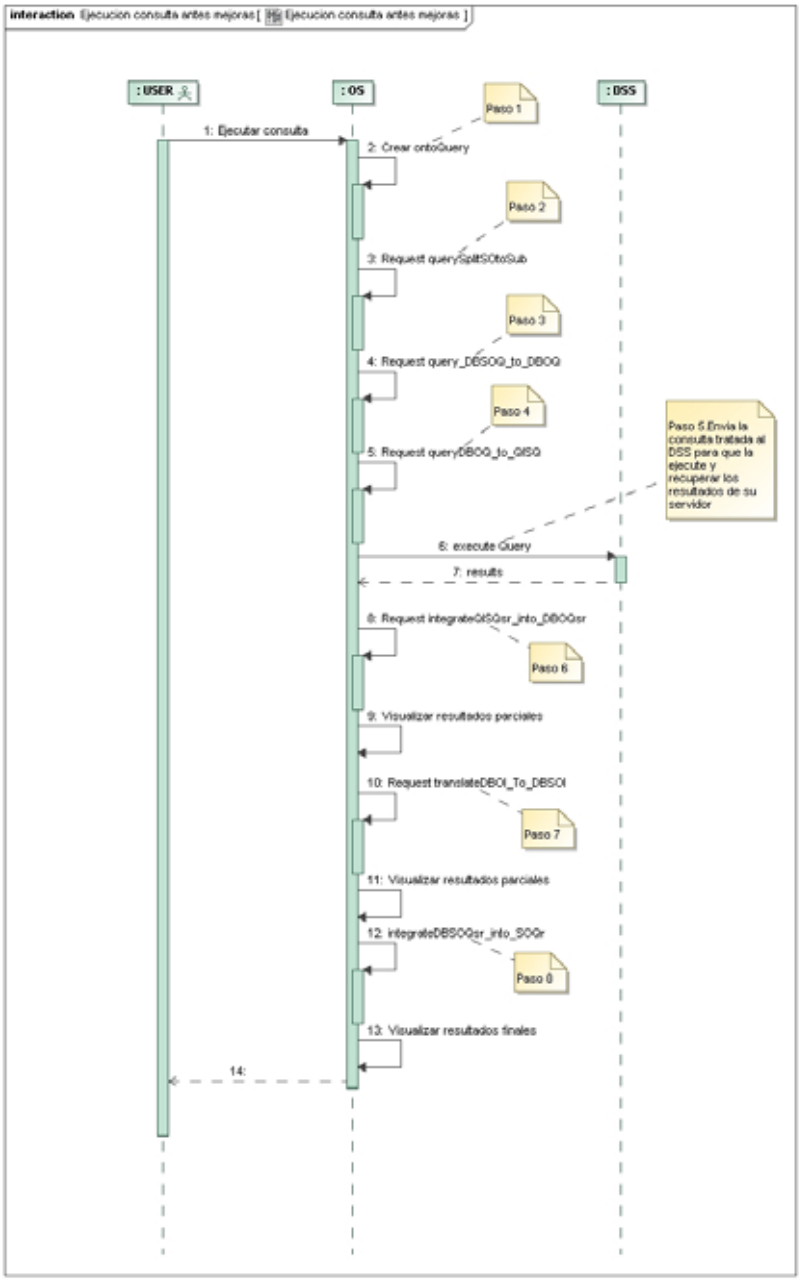


Figura 46: Diagrama de secuencia de funcionamiento del ontology server

Con el sistema de integración como modelo, se desarrolló también el sistema de información para desarrollar y ejecutar las consultas.

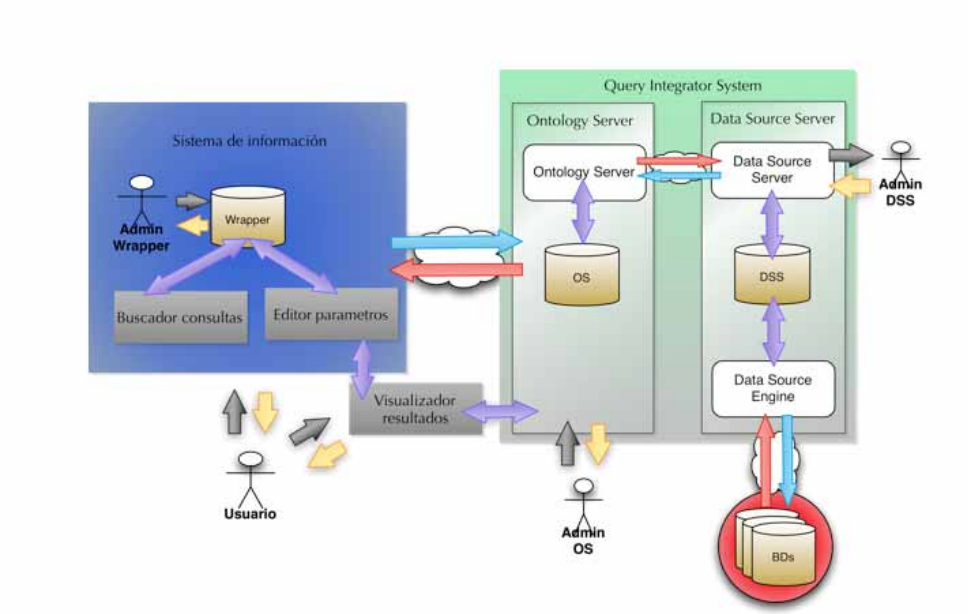


Figura 47: Arquitectura del sistema completo

El sistema de información permite, por un lado, la creación por medio de un administrador, de las consultas al sistema de integración, utilizando para ello cualquiera de los términos denominados “preferidos” que forman parte de las fuentes de datos.

Por otro lado, un usuario podrá ejecutar las consultas, modificando los parámetros o valores de las consultas desarrolladas por el usuario administrador.

El sistema de integración recibe una serie de peticiones en forma de consulta, y este, en base a los datos de los que dispone tanto a nivel de metadatos de las bases de datos, como la información proporcionada por las ontologías que gestiona, es capaz de integrar los datos y devolver una respuesta al usuario (Figura 47).

El sistema de información se desarrolló utilizando el patrón Model-View-Controller (MVC) (Reenskaug, 1978), utilizando tecnología .NET de Microsoft. La elección de esta tecnología se debe a una mejor integración con el sistema OntoMediator, aunque este último podría funcionar con cualquier tipo de tecnología, ya que utiliza servicios web como interfaz.

Como se ha dicho anteriormente, aunque los datos utilizados para este caso de estudio son públicos obtenidos de GEO, contiene datos clínicos que eventualmente

podrían ser privados. Por lo tanto, es necesario implementar medidas de acceso y autenticación que implican que la aplicación web tenga que implementar medidas de seguridad y gestión de usuarios.

7.3.4 Pruebas

Para comprobar el correcto funcionamiento del módulo de visualización y que los resultados devueltos por éste en los distintos formatos que proporciona son correctos, se ejecutaron un conjunto de consultas y se analizaron sus resultados.

Tanto aquellas consultas simples, en las que no existe integración de datos, como en consultas más complejas, con distintos tamaños de resultados se comprobó que la obtención de los resultados era correcta.

Para validar los resultados de integración obtenidos se empleó OntoMediator y se ejecutaron una serie de consultas complejas sobre el sistema desarrollado. En paralelo se incluyeron todos los datos en una base de datos relacional y se desarrollaron consultas SQL equivalentes a las consultas semánticas de OntoMediator, en las que la parte de integración de OntoMediator es sustituida por el experto que conoce todas las bases de datos y realiza las consultas a mano.

Los resultados se describen en una serie de tablas en el Anexo I, donde se muestra la consulta que se desea recuperar en lenguaje natural, la consulta XML generada por el sistema (Figura 48), su equivalente en SQL y se visualizan parte de los resultados.

Para cada una de las pruebas se han comprobado que los resultados obtenidos tanto en la salida XML, como en la salida MDB, eran equivalentes a los resultados de la consulta realizada a mano, con un rendimiento algo inferior, pero pudiendo manejar automáticamente problemas semánticos y de granularidad.

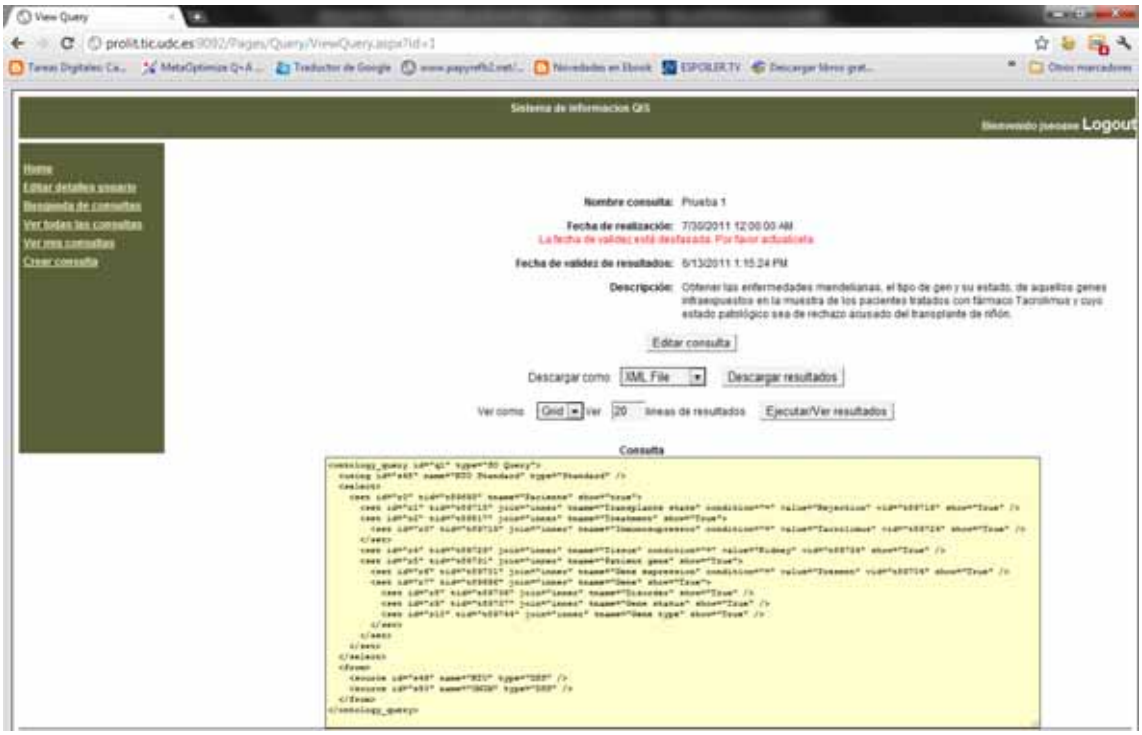


Figura 48: Visualización de la consulta almacenada

El sistema permite integrar información compleja que, posteriormente, puede ser utilizada para un análisis, desde fuentes distribuidas y heterogéneas.

El sistema desarrollado se está probando actualmente en el Yale Center for Medical Informatics por el equipo del Dr. Luis Marenco que desarrolló el *framework* Ontomediator para obtener una validación externa.

7.3.5 Discusión

El objetivo de este caso de estudio era demostrar que una herramienta de integración permitiría a los investigadores facilitar la obtención de hipótesis en farmacogenética, a partir de relacionar e integrar información sobre datos clínicos y genéticos sobre pacientes trasplantados y su reacción a los inmunosupresores.

En el artículo de Flechner (Flechner et al., 2004) a partir de los genotipos de los pacientes y de su histopatología y utilizando técnicas de *machine learning*, detectan que existen diferentes patrones genómicos para el rechazo/no rechazo de los trasplantes, identificando también una serie de genes sobre-expresados, que posteriormente relacionan con funciones de respuesta inflamatoria e inmune, e infra-expresados en metabolismo celular, que relacionan con el daño provocado por cierto

tipo de medicación anti-rechazo. Por otro lado, manualmente relacionan genes del estudio con genes encontrados en la literatura para comparar los resultados con respecto a sus funciones.

Todos los análisis que realizan en este artículo siguen el siguiente patrón:

- Análisis de datos genómicos/clínicos mediante aprendizaje máquina
- Relación manual de los resultados con respecto a fármacos empleados, funciones celulares o enfermedades.

Este método de trabajo es tedioso y puede provocar que se pasen por alto patrones de análisis debido a no tener todos los datos integrados.

La herramienta desarrollada, permite tener todos los datos necesarios para realizar el análisis, no sólo los datos genómicos y clínicos, sino también los relacionados con fármacos, funciones celulares o enfermedades mendelianas relacionadas, disponibles para poder seleccionar aquellos que sea interesante someter a análisis.

De esta manera, el investigador podría proponer sus hipótesis basándose en un gran conjunto de datos, seleccionar aquellos que le interesa analizar y, posteriormente, realizar un análisis utilizando estadística o aprendizaje máquina para relacionar los datos que quiera analizar automáticamente.

El desarrollo de una herramienta de integración tan compleja como esta debe cumplir una serie de requisitos y tener en cuenta una serie de factores. Estos factores han sido analizados en el marco metodológico que se presenta en esta tesis y han sido utilizados para la elección del *framework* de integración y para el desarrollo de esta herramienta.

El uso de este marco metodológico ha permitido el desarrollo de una herramienta de integración que cumple con los requisitos que se plantearon para este caso de estudio y que permite responder consultas complejas sobre datos de diversos niveles de información en salud.

7.4 Discusión del sistema de integración

La hipótesis presentada en esta tesis planteaba que el desarrollo de un marco metodológico para el desarrollo de un modelo de integración que favorezca el intercambio de datos biomédicos entre los distintos niveles de información en salud permite extraer nuevo conocimiento que favorezca tanto a la investigación biomédica como a la práctica clínica.

El desarrollo de un sistema de integración, como todo sistema software, requiere un proceso de análisis que permita evitar errores en el desarrollo y reducir costes al permitir tomar las decisiones correctas.

El marco metodológico desarrollado se basa en la aplicación de una serie de pasos para analizar los datos y las fuentes que se pretende integrar, así como el contexto definido por los requisitos, teniendo en cuenta una serie de parámetros o decisiones clave.

Para probar el funcionamiento de este marco metodológico, se escogió dentro del universo de sistemas de integración posibles tres tipos de problemas relacionados con la medicina personalizada, con la intención de abarcar un rango importante de las posibles aplicaciones de integración de información biomédica: aplicaciones de integración de información para el punto de atención al paciente, aplicaciones para gestión y análisis de datos en estudios de asociación que relacionen genotipo y fenotipo y finalmente aplicaciones de integración de datos para farmacogenética.

Para cada uno de estos tres tipos de problemas se seleccionó un escenario de aplicación para desarrollar un sistema de anotación automática de historia clínica, una herramienta de ayuda para realizar estudios de asociación y un sistema de información para el análisis de la reacción de pacientes trasplantados a fármacos anti-rechazo.

El primero de los ejemplos permite comprobar la aplicación del marco metodológico en una aplicación de ayuda a toma de decisiones en el ámbito clínico, incluyendo información de pacientes y su anotación con datos externos. Este tipo de sistema permite evaluar la interacción con usuarios no expertos, así como el acceso e integración de datos de pacientes. El segundo permite la integración de datos

relacionados con estudios de asociación, que incluye datos epidemiológicos, de enfermedades, etc. En este caso se avanza en un tipo de integración más complejo, que incluye integración horizontal. Finalmente, el tercer caso de estudio es el más complejo e incluye la integración de diversos tipos de datos genéticos, que incluyen datos de expresión. Estos datos se representan con distinta granularidad y requieren de la aplicación de tecnologías semánticas.

Al analizar los tipos de sistemas de integración obtenidos después de aplicar el marco metodológico a los tres escenarios, una de las características más llamativas es que los tres sistemas de integración presentan una arquitectura federada con una estrategia de modelado *top-down* y una gestión de consultas LaV. Si bien las últimas dos características están relacionadas entre ellas, el factor en común de las tres es la capacidad de evolución del sistema. Esto es debido a que un requisito imprescindible de los tres escenarios es la posibilidad de que los sistemas puedan evolucionar a lo largo del tiempo, añadiendo nuevas fuentes o modificando su esquema sin unos grandes costes de desarrollo. La importancia de la evolución en la selección de la arquitectura al analizar los sistemas desarrollados ya se apuntó al analizar los sistemas de integración existentes en la bibliografía en el capítulo III.

Los resultados en cada uno de los casos fueron buenos y se demostró que el marco metodológico facilitaba el desarrollo de este tipo de sistemas de integración, al ofrecer al desarrollador una serie de pasos y recomendaciones.

En primer lugar proporciona una guía que permite seleccionar adecuadamente las fuentes de información, basándose en una serie de características que deben tener las fuentes de datos para facilitar el desarrollo, la integración, el mantenimiento y el funcionamiento general del sistema.

Por otro lado ofrece una serie de pasos que el desarrollador debe de seguir en su etapa de análisis para formalizar el problema de integración. Dichos pasos no deben tomarse como una metodología estricta, sino que complementan el proceso de análisis permitiendo que el desarrollar tenga en cuenta una serie de factores que son cruciales no solo para el correcto proceso de desarrollo del sistema de integración,

sino también para su correcta evolución con respecto a añadir nuevas fuentes de datos, modificar las actuales o reflejar cambios en el modelo.

Finalmente se proporciona también una métrica que permite evaluar los factores mencionados en el apartado anterior de una manera cualitativa para poder escoger las opciones que mejor se adecuen en cada caso, para cada uno de los pasos mencionados anteriormente.

De esta manera, cuando el desarrollador se enfrenta a un problema de integración de datos, el marco metodológico presentado permite tener en cuenta una serie de factores que ayudan a solventar los problemas, como el de tratar con un gran número de bases de datos, la complejidad y variabilidad de los datos, los requisitos de actualización, la heterogeneidad y distribución las fuentes y la disparidad de formato y semántica de los datos.

Si bien este marco metodológico no pretende convertirse en una solución universal para los problemas de integración de datos, debido a la variabilidad y a la complejidad que representan, proporciona unos métodos, que explícita o implícitamente, obligan al desarrollador a analizar profundamente, tanto el escenario, como los tipos de datos que se desean integrar, seleccionando correctamente las fuentes más adecuadas.

Otra ventaja de este planteamiento es que permite evitar errores de diseño y planificar los costes de futuras modificaciones en el modelo o ampliación de las fuentes de datos.

Así, por ejemplo en el primer escenario de prueba, del punto de atención al paciente, se ha demostrado que no es necesario, para este problema en concreto, un sistema de integración complejo como el utilizado para el escenario tres, que es mucho más costoso de desarrollar y de mantener, mientras que el escenario tres no podría utilizar el sistema de integración planteado en el escenario 2, debido a la complejidad de los requisitos y de las relaciones que se desean integrar.

En el trabajo de Goble (Goble and Stevens, 2008), para defender la idea de la integración ligera que realizan los mashups, Goble defiende que para bien o para mal,

en bioinformática el desarrollo de aplicaciones se basa en el mantra "*just in time, just enough*" (solo lo suficiente, a tiempo) y es la biología la que importa, no la ingeniería, ya que la ingeniería presenta soluciones demasiado genéricas, complejas, de gran tiempo de desarrollo y que no se adecuan a las necesidades del usuario. El trabajo desarrollado en esta tesis demuestra que esto no es correcto, ya que una aproximación metodológica en el desarrollo de sistemas de integración puede ofrecer soluciones dimensionadas, correctas y adaptadas a las necesidades del problema.

8 Conclusiones

La tarea de integración de datos biomédicos es un proceso complejo que combina las complicaciones de integrar datos procedentes de fuentes de datos distribuidas y heterogéneas, con la complejidad de los datos biológicos, en cuanto a la representación y modelado de estos datos en entidades.

Las soluciones de integración de datos biomédicos se han enfocado tradicionalmente en problemas concretos. No existe ningún modelo genérico de integración de datos que pueda aplicarse a cualquier problema, debido a los distintos problemas de integración que pueden aparecer. Es por eso que se hace necesaria la creación de un marco metodológico que permita encontrar el modelo de integración más adecuado para cada problema.

En esta tesis se ha desarrollado un marco metodológico que permite identificar soluciones de integración adaptadas al problema a solucionar. Este marco metodológico se basa en tres pilares. En primer lugar, a partir del análisis del problema, identificar y categorizar correctamente los tipos de datos a integrar. En segundo lugar, a partir de una serie de pasos, se deben analizar una serie de cuestiones relacionadas con técnicas, métodos o arquitecturas de integración de datos a partir de una métrica creada a tal efecto. Finalmente, apoyándose en los valores obtenidos de la métrica y del análisis del problema, permite tomar las decisiones adecuadas para obtener un sistema de integración adecuado con el mínimo coste de diseño y mantenimiento.

Para comprobar el funcionamiento del marco metodológico desarrollado se escogieron tres escenarios de integración de datos relacionados con medicina translacional, que permitían integrar varios tipos de datos en diversos niveles de información en salud.

El primero de estos escenarios consistía en convertir un visor de historia clínica electrónica en un punto de atención al paciente donde, una vez que la historia se cargase, se identificasen automáticamente los conceptos de la historia clínica que estuviesen presentes en la ontología UMLS y se buscase información relacionada con esos conceptos en diversas fuentes de datos heterogéneas. Estas búsquedas se

realizaron expandiendo las búsquedas, utilizando traducciones, sinónimos y conceptos definidos en la ontología y utilizando un sistema de integración federado basado en wrapper/mediator, en el que el proceso de integración se realiza en dos fases, una fase de expansión semántica y una segunda de expansión por las distintas fuentes de datos heterogéneas.

El segundo caso de estudio consiste en una herramienta que permita diseñar estudios que asocien un determinado genotipo con un determinado fenotipo. De esta manera, partiendo del fenotipo que se desea estudiar, se buscan los genes o rutas metabólicas que estén relacionados con esta enfermedad y, posteriormente, las variaciones, SNPs en este caso, que estén relacionados con estos genes y rutas. Para desarrollar este sistema se optó por un sistema federado basado en workflows implementado sobre Biomoby, lo que permitió contar con la infraestructura de servicios y con el modelo de datos propio de Biomoby. Esta aproximación de integración permite la reutilización de cada uno de los servicios desarrollados individualmente; así como la inclusión, dentro del proceso de integración o análisis posterior, de otros servicios Biomoby externos sin problemas de interoperabilidad.

Finalmente, el tercero de los escenarios consiste en un sistema de información en farmacogenética que permite la recuperación de datos de diversas fuentes como datos genómicos, clínicos, fármacos o enfermedades mendelianas, realizando de manera sencilla, complejas consultas sobre rechazo de riñón trasplantado y reacciones con respecto a fármacos inmunosupresivos. Este sistema de integración es el más complejo de los tres, ya que requiere integración semántica compleja y tratamiento de la granularidad. Se ha implementado utilizando un sistema federado, desarrollando una ontología que funciona como modelo global y, posteriormente, mapeando cada fuente de datos a integrar en una ontología y relacionando esta con la del modelo global. Para realizar este complejo proceso se ha utilizado el framework de integración QIS (Query Integrator System).

El uso del marco metodológico desarrollado en esta tesis ha permitido agilizar el desarrollo de estos tres sistemas de integración de datos biomédicos, minimizando los errores de diseño y los tiempos de desarrollo y evolución del sistema de

integración, ofreciendo soluciones dimensionadas y adaptadas al usuario y al problema de integración al que se enfrente.

9 Conclusions

The task of biomedical data integration is a complex process which combines the complications of integrating data from distributed and heterogeneous data sources with the complexity of biological data, based on the representation and modeling of these entities.

Existing solutions in biomedical data integration have traditionally focused on specific problems. There is no generic model for data integration that can be applied to any problem, due to the different integration problems that can be found. That is why it is necessary to create a methodological framework that allows finding the most appropriate integration model for each problem.

In this thesis a methodological framework for developing integration solutions suited to the problem to solve has been presented. This methodological framework is based on three pillars. First, by analyzing the problem, this framework identifies and properly categorizes the kinds of data to integrate. Second, from a series of steps, the framework aims to consider a range of technical issues, methods or data integration architectures from a measure created for this purpose. Finally, based on the values obtained from the metric and the analysis of the problem, the methodological framework supports decisions in order to obtain the proper integration with the minimum cost design and maintenance.

In order to validate the methodological framework, three scenarios were chosen to integrate data related to translational medicine, which allowed integrating multiple data types at different levels of information in health.

The first scenario consisted in converting an electronic health record client in a point of care where, once the story was loaded, the concepts of the record that were present in the UMLS ontology were automatically identified and information related to these concepts were found in several distributed heterogeneous datasources. These searches were carried out to expand the queries, using translations, synonyms and concepts defined in the ontology and using a federated integration system based on the wrapper/mediator pattern. In this case, the integration process is performed in

two phases, a semantic expansion phase and a query expansion phase through the heterogeneous datasources.

The second scenario is a tool for supporting the design of association studies that associate a particular genotype with a particular phenotype. Thus, based on the phenotype to be studied, genes and pathways related with this phenotype are searched. Thereafter, the variations (SNPs in this study) related with these genes or pathways are searched. To develop this system, a workflow federated integration architecture based on Biomoby was chosen. This Biomoby architecture allows having the service infrastructure and the Biomoby data model. This integration approach allows reusing each of the services developed individually, and the inclusion, in the integration or subsequent analysis processes, of other external Biomoby services without interoperability problems.

Finally, the third scenario consists in a pharmacogenetics information system that allows data retrieval and integration from several genomic, clinical, pharmaceutical or genetic disease datasources, developing easily complex queries about kidney transplant rejection and negative reactions to immunosuppressive drugs. This integration system is the most complex of the three, since it requires complex semantic integration and treatment of data granularity. It has been implemented using a federated system, developing an ontology that operates as a global model and then mapping each datasource to be integrated into an ontology and relating this one with the global model. In order to perform this complex integration process the QIS framework (Query Integrator System) has been used.

Using the methodological framework proposed in this thesis has allowed to speeding up the development of these three systems of biomedical data integration, minimizing design errors and the development and evolution times of the integration system, offering solutions suited to the user and to the integration problem to solve.

10 Futuros Desarrollos

Después de desarrollar el contenido de esta tesis, se han identificado una serie de nuevas líneas de trabajo en las que se podría profundizar para ofrecer sistemas de integración más complejos, con mayor capacidad y más simples para el usuario.

- En este trabajo, especialmente al utilizar el framework de integración QIS, se ha visto la gran utilizad de sistemas que automáticamente creen ontologías a partir de las fuentes de datos, ya sean como el Datasource Server del QIS o soluciones similares que pasan del modelo relacional a ontologías como el R2O y ODEMapster (Barrasa and Gómez-Pérez, 2006), así como la "semantización" de recursos, como el Bio2RDF (Belleau et al., 2008) o el NOR2O (Villazón-Terrazas et al., 2009). Es necesario profundizar en la inclusión de sistemas como los comentados en el marco metodológico, de manera que se permita el acceso semántico a fuentes que no han sido diseñadas para ser anotadas semánticamente.
- Una vez todas las fuentes de datos estén en un lenguaje de representación de ontologías como RDF o OWL, un sistema de integración podría aplicar sistemas de razonamiento utilizando en ontologías, ya sea basados en lógica de primer orden, lógica descriptiva o razonamiento basado en reglas. Existen frameworks de razonamiento basado en ontologías como JENA (McBride, 2002), que permitirían potenciar el proceso de integración.
- Una de las fuentes de datos que más información aportan en el práctica clínica son las imágenes biomédicas, no sólo las imágenes de modalidades radiológicas clásicas, sino también otros tipos de imágenes como las moleculares o histológicas. En este marco metodológico, la integración de las imágenes contemplada dentro del sistema se realiza por anotaciones realizadas en las imágenes, pero no basándose en el contenido de las imágenes. Existen diversos sistemas (Müller et al., 2004, Depeursinge et al., 2011) que permiten la recuperación de imágenes basada en contenido, que pueden servir para recuperar cierto

tipo de imágenes de una base de datos, la anotación automática de imágenes o la anotación manual utilizando un esquema estándar (Channin et al., 2009), para una posterior recuperación e integración.

- Otra importante mejora en lo referente a la relación con el usuario es el interfaz de consultas. En un futuro sería deseable que el usuario pudiese formular las consultas en lenguaje natural y el sistema permitiese procesar esta consulta por medio de técnicas de minería de textos biomédicos (Cohen and Hersh, 2005) (Krallinger et al., 2008) o procesado de lenguaje natural. En relación con esto también sería muy útil añadir anotación de historia clínica no sólo basada en los términos de alguna ontología, sino también basada en contexto (Bada and Hunter, 2011) (Kreuzthaler et al., 2011).
- Finalmente, en cuanto a las arquitecturas analizadas de integración de datos, existen interesantes aproximaciones con sistemas multi-agente, que distribuyen las responsabilidades de integración entre diversos agentes. En esta línea de distribución no solo del proceso de integración, sino también de las posibilidades de almacenamiento, existen arquitecturas P2P para integración de datos, como el Piazza (Halevy et al., 2003), o el GridVine (Cudre-Mauroux et al., 2007) pero no se ha encontrado ninguna arquitectura P2P para integración de datos biomédicos, por lo que sería útil el desarrollo de un prototipo de integración de datos biomédicos distribuyendo toda la información a integrar en diversos *peers* con responsabilidades similares.

Referencias

ACHARD, F., VAYSSEIX, G. & BARILLOT, E. 2001. XML, bioinformatics and data integration. *Bioinformatics,* 17**,** 115-125.

AHN, G.-J. & SANDHU, R. 2000. Role-based authorization constraints specification. *ACM Trans. Inf. Syst. Secur.,* 3**,** 207-226.

AKULA, S. P., MIRIYALA, R. N., THOTA, H., RAO, A. A. & GEDELA, S. 2009. Techniques for integrating -omics data. *Bioinformation,* 3**,** 284-6.

ALONSO-CALVO, R., MAOJO, V., BILLHARDT, H., MARTIN-SANCHEZ, F., GARCÍA-REMESAL, M. & PÉREZ-REY, D. 2007. An agent- and ontology-based system for integrating public gene, protein, and disease databases. *Journal of Biomedical Informatics,* 40**,** 17-29.

ANTEZANA, E., KUIPER, M. & MIRONOV, V. 2009. Biological knowledge management: the emerging role of the Semantic Web technologies. *Briefings in Bioinformatics,* 10**,** 392-407.

ANWAR, N. & HUNT, E. 2009. Francisella tularensis novicida proteomic and transcriptomic data integration and annotation based on semantic web technologies. *BMC Bioinformatics,* 10 Suppl 10**,** S3.

ARENS, Y., HSU, C.-N. & KNOBLOCK, C. A. 1998. Query processing in the SIMS information mediator. *Readings in agents.* Morgan Kaufmann Publishers Inc.

ARENSON, A. D. 2003. Federating data with Information Integrator. *Briefings in Bioinformatics,* 4**,** 375-381.

ARRAIS, J., PEREIRA, J., FERNANDES, J. & OLIVEIRA, J. Year. GeNS: a biological data integration platform. *In*, 2009. 850-855.

BADA, M. & HUNTER, L. 2011. Desiderata for ontologies to be used in semantic annotation of biomedical documents. *Journal of Biomedical Informatics,* 44**,** 94-101.

BAKER, P. G., BRASS, A., BECHHOFER, S., GOBLE, C., PATON, N. & STEVENS, R. 1998. TAMBIS--Transparent Access to Multiple Bioinformatics Information Sources. *Proc Int Conf Intell Syst Mol Biol,* 6**,** 25-34.

BARRASA, J. & GÓMEZ-PÉREZ, A. Year. Upgrading Relational Legacy Data to eh Semantic Web. *In:* 15th International World Wide Web Conference (WWW2006), 2006/05// 2006. ACM Press, 2.

BARSALOU, T., SIAMBELA, N., KELLER, A. M. & WIEDERHOLD, G. 1991. Updating relational databases through object-based views. *SIGMOD Rec.,* 20**,** 248-257.

BATINI, C., LENZERINI, M. & NAVATHE, S. B. 1986. A comparative analysis of methodologies for database schema integration. *ACM Comput. Surv.,* 18**,** 323-364.

BELLEAU, F., NOLIN, M.-A., TOURIGNY, N., RIGAULT, P. & MORISSETTE, J. 2008. Bio2RDF: Towards a mashup to build bioinformatics knowledge systems. *Journal of Biomedical Informatics,* 41**,** 706-716.

BENSON, D., BOGUSKI, M., LIPMAN, D. & OSTELL, J. 1990. The National Center for Biotechnology Information. *Genomics,* 6**,** 389-91.

BERNERS-LEE, T., HENDLER, J. & LASSILA, O. 2001. The Semantic Web – a new form of Web content that is meaningful to computers will unleash a revolution of new possibilities. . *Sci Am 2001,* 284**,** 34–43.

BERNSTAM, E. V., SMITH, J. W. & JOHNSON, T. R. 2010. What is biomedical informatics? *J Biomed Inform,* 43**,** 104-10.

BICHUTSKIY, V. Y., COLMAN, R., BRACHMANN, R. K. & LATHROP, R. H. 2007. Heterogeneous Biomedical Database Integration Using a Hybrid Strategy: A p53 Cantcer Research Database. *Cancer Inform,* 2**,** 277-87.

BIDGOOD, W. D., JR., HORII, S. C., PRIOR, F. W. & VAN SYCKLE, D. E. 1997. Understanding and using DICOM, the data interchange standard for biomedical imaging. *J Am Med Inform Assoc,* 4**,** 199-212.

BIRKLAND, A. & YONA, G. 2006. BIOZON: a hub of heterogeneous biological data. *Nucleic Acids Res,* 34**,** D235-42.

BITTNER, K. & SPENCE, I. 2003. *Use case modeling*, Addison-Wesley.

BLASCHKE, C., HIRSCHMAN, L., YEH, A. & VALENCIA, A. 2003. Critical assessment of information extraction systems in biology. *Comp Funct Genomics,* 4**,** 674-7.

BLOBEL, B. G., ENGEL, K. & PHAROW, P. 2006. Semantic interoperability--HL7 Version 3 compared to advanced architecture standards. *Methods Inf Med,* 45**,** 343-53.

BLOIS, M. S. 1984. *Information and medicine: the nature of medical descriptions*, University of California Press.

BODENREIDER, O. & STEVENS, R. 2006. Bio-ontologies: current trends and future directions. *Brief Bioinform,* 7**,** 256-74.

BRAZHNIK, O. & JONES, J. F. 2007. Anatomy of data integration. *Journal of Biomedical Informatics,* 40**,** 252-269.

BRAZMA, A., HINGAMP, P., QUACKENBUSH, J., SHERLOCK, G., SPELLMAN, P., STOECKERT, C., AACH, J., ANSORGE, W., BALL, C. A., CAUSTON, H. C., GAASTERLAND, T., GLENISSON, P., HOLSTEGE, F. C., KIM, I. F., MARKOWITZ, V., MATESE, J. C., PARKINSON, H., ROBINSON, A., SARKANS, U., SCHULZE-KREMER, S., STEWART, J., TAYLOR, R., VILO, J. & VINGRON, M. 2001. Minimum information about a microarray experiment (MIAME)-toward standards for microarray data. *Nat Genet,* 29**,** 365-71.

BRAZMA, A., KRESTYANINOVA, M. & SARKANS, U. 2006. Standards for systems biology. *Nat Rev Genet,* 7**,** 593-605.

BROOKSBANK, C. & QUACKENBUSH, J. 2006. Data standards: a call to action. *OMICS,* 10**,** 94-9.

BUKHMAN, Y. V. & SKOLNICK, J. 2001. BioMolQuest: integrated database-based retrieval of protein structural and functional information. *Bioinformatics,* 17**,** 468-78.

BUNEMAN, P. 1997. Semistructured data. *Proceedings of the sixteenth ACM SIGACT-SIGMOD-SIGART symposium on Principles of database systems.* Tucson, Arizona, United States: ACM.

BURGUN, A. & BODENREIDER, O. 2008. Accessing and integrating data and knowledge for biomedical research. *Yearb Med Inform***,** 91-101.

BUSH, N. E., BOWEN, D. J., WOOLDRIDGE, J., LUDWIG, A., MEISCHKE, H. & ROBBINS, R. 2004. What do we mean by Internet access? A framework for health researchers. *Prev Chronic Dis,* 1**,** A15.

BUSSE, S., KUTSCHE, R. D. & LESER, U. Year. Strategies for the Conceptual Design of Federated Information Systems. *In*, 2000. IOS Press, 23.

CABARCOS, A., SANCHEZ, T., SEOANE, J. A., AGUIAR-PULIDO, V., FREIRE, A., DORADO, J. & PAZOS, A. 2010. Retrieval and management of medical information from heterogeneous sources, for its integration in a medical record visualisation tool. *Int J Electron Healthc,* 5**,** 371-85.

CANNATA, N., MERELLI, E. & ALTMAN, R. B. 2005. Time to organize the bioinformatics resourceome. *PLoS Comput Biol,* 1**,** e76.

CIOS, K. J. & MOORE, G. W. 2002. Uniqueness of medical data mining. *Artif Intell Med,* 26**,** 1-24.

CLAUS, B. L. & UNDERWOOD, D. J. 2002. Discovery informatics: its evolving role in drug discovery. *Drug Discov Today,* 7**,** 957-66.

COHEN, A. M. & HERSH, W. R. 2005. A survey of current work in biomedical text mining. *Briefings in Bioinformatics,* 6**,** 57-71.

CONNOLLY, T. M. & BEGG, C. E. 2005. *Database systems: a practical approach to design, implementation, and management*, Addison-Wesley.

CORNELL, M., PATON, N. W., SHENGLI, W., GOBLE, C. A., MILLER, C. J., KIRBY, P., EILBECK, K., BRASS, A., HAYES, A. & OLIVER, S. G. Year. GIMS-a data warehouse for storage and analysis of genome sequence and functional data. *In:* Bioinformatics and

Bioengineering Conference, 2001. Proceedings of the IEEE 2nd International Symposium on, 2001. 15-22.
COSTA, C., FERREIRA, C., BASTIAO, L., RIBEIRO, L., SILVA, A. & OLIVEIRA, J. L. 2011. Dicoogle - an open source peer-to-peer PACS. *J Digit Imaging,* 24**,** 848-56.
COSTA, C., FREITAS, F., PEREIRA, M., SILVA, A. & OLIVEIRA, J. L. 2009. Indexing and retrieving DICOM data in disperse and unstructured archives. *Int J Comput Assist Radiol Surg,* 4**,** 71-7.
CRITCHLOW, T., FIDELIS, K., GANESH, M., MUSICK, R. & SLEZAK, T. 2000. DataFoundry: information management for scientific data. *IEEE Trans Inf Technol Biomed,* 4**,** 52-7.
CUDRE-MAUROUX, P., AGARWAL, S. & ABERER, K. 2007. GridVine: An infrastructure for peer information management. *IEEE Internet Computing,* 11**,** 36-44.
CHANNIN, D. S., MONGKOLWAT, P., KLEPER, V. & RUBIN, D. L. 2009. The Annotation and Image Mark-up project. *Radiology,* 253**,** 590-2.
CHAWATHE, S., GARCIA-MOLINA, H., HAMMER, J., IRELAND, K., PAPAKONSTANTINOU, Y., ULLMAN, J. & WIDOM, J. 1994. The TSIMMIS Project: Integration of Heterogenous Information Sources. *Information Processing Society of Japan (IPSJ 1994).* Tokyo, Japan.
CHEUNG, K.-H., KASHYAP, V., LUCIANO, J. S., CHEN, H., WANG, Y. & STEPHENS, S. 2008. Semantic mashup of biomedical data. *Journal of Biomedical Informatics,* 41**,** 683-686.
CHEUNG, K.-H., PRUD'HOMMEAUX, E., WANG, Y. & STEPHENS, S. 2009. Semantic Web for Health Care and Life Sciences: a review of the state of the art. *Briefings in Bioinformatics,* 10**,** 111-113.
CHUNG, S. Y. & WONG, L. 1999. Kleisli: a new tool for data integration in biology. *Trends Biotechnol,* 17**,** 351-5.
DAVIDSON, S. B., OVERTON, C. & BUNEMAN, P. 1995. Challenges in integrating biological data sources. *Journal of Computational Biology,* 2**,** 557-572.
DAVIDSON, S. B., OVERTON, C., TANNEN, V. & WONG, L. 1997. BioKleisli: a digital library for biomedical researchers. *International Journal on Digital Libraries,* 1**,** 36-53.
DEPEURSINGE, A., FISCHER, B., MULLER, H. & DESERNO, T. M. 2011. Prototypes for content-based image retrieval in clinical practice. *Open Med Inform J,* 5**,** 58-72.
DIAS, G. S., OLIVEIRA, J. L., VICENTE, J. & MARTIN-SANCHEZ, F. 2006. Integrating medical and genomic data: a successful example for rare diseases. *Stud Health Technol Inform,* 124**,** 125-30.
DOLIN, R. H., ALSCHULER, L., BEEBE, C., BIRON, P. V., BOYER, S. L., ESSIN, D., KIMBER, E., LINCOLN, T. & MATTISON, J. E. 2001. The HL7 Clinical Document Architecture. *J Am Med Inform Assoc,* 8**,** 552-69.
DONELSON, L., TARCZY-HORNOCH, P., MORK, P., DOLAN, C., MITCHELL, J. A., BARRIER, M. & MEI, H. 2004. The BioMediator system as a data integration tool to answer diverse biologic queries. *Stud Health Technol Inform,* 107**,** 768-72.
DROPTHINGS. 2012. *Dropthings* [Online]. Google. Available: http://code.google.com/p/dropthings/ [Accessed 10/02/2012 2012].
DUDLEY, J. T. & BUTTE, A. J. 2010. In silico research in the era of cloud computing. *Nat Biotechnol,* 28**,** 1181-5.
ECKMAN, B. A., KOSKY, A. S. & LAROCO, J., LEONARDO A. 2001. Extending traditional query-based integration approaches for functional characterization of post-genomic data. *Bioinformatics,* 17**,** 587-601.
EMIL, C. L. 1999. Conflicts in Policy-Based Distributed Systems Management. *IEEE Transactions on Software Engineering,* 25**,** 852-869.
ETZOLD, T., ULYANOV, A. & ARGOS, P. 1996. SRS: information retrieval system for molecular biology data banks. *Methods Enzymol,* 266**,** 114-28.
FISHER, J. & HENZINGER, T. A. 2007. Executable cell biology. *Nat Biotechnol,* 25**,** 1239-49.

FLECHNER, S. M., KURIAN, S. M., HEAD, S. R., SHARP, S. M., WHISENANT, T. C., ZHANG, J., CHISMAR, J. D., HORVATH, S., MONDALA, T., GILMARTIN, T., COOK, D. J., KAY, S. A., WALKER, J. R. & SALOMON, D. R. 2004. Kidney transplant rejection and tissue injury by gene profiling of biopsies and peripheral blood lymphocytes. *Am J Transplant,* 4**,** 1475-89.

FLICEK, P., AMODE, M. R., BARRELL, D., BEAL, K., BRENT, S., CHEN, Y., CLAPHAM, P., COATES, G., FAIRLEY, S., FITZGERALD, S., GORDON, L., HENDRIX, M., HOURLIER, T., JOHNSON, N., KAHARI, A., KEEFE, D., KEENAN, S., KINSELLA, R., KOKOCINSKI, F., KULESHA, E., LARSSON, P., LONGDEN, I., MCLAREN, W., OVERDUIN, B., PRITCHARD, B., RIAT, H. S., RIOS, D., RITCHIE, G. R., RUFFIER, M., SCHUSTER, M., SOBRAL, D., SPUDICH, G., TANG, Y. A., TREVANION, S., VANDROVCOVA, J., VILELLA, A. J., WHITE, S., WILDER, S. P., ZADISSA, A., ZAMORA, J., AKEN, B. L., BIRNEY, E., CUNNINGHAM, F., DUNHAM, I., DURBIN, R., FERNANDEZ-SUAREZ, X. M., HERRERO, J., HUBBARD, T. J., PARKER, A., PROCTOR, G., VOGEL, J. & SEARLE, S. M. 2011. Ensembl 2011. *Nucleic Acids Res,* 39**,** D800-6.

FORCADA, M., GINESTÍ-ROSELL, M., NORDFALK, J., O'REGAN, J., ORTIZ-ROJAS, S., PÉREZ-ORTIZ, J., SÁNCHEZ-MARTÍNEZ, F., RAMÍREZ-SÁNCHEZ, G. & TYERS, F. 2011. Apertium: a free/open-source platform for rule-based machine translation. *Machine Translation,* 25**,** 127-144.

FRIEDMAN, M., LEVY, A. & MILLSTEIN, T. 1999. Navigational plans for data integration. *Proceedings of the sixteenth national conference on Artificial intelligence and the eleventh Innovative applications of artificial intelligence conference innovative applications of artificial intelligence.* Orlando, Florida, United States: American Association for Artificial Intelligence.

FULLER, S. S., KETCHELL, D. S., TARCZY-HORNOCH, P. & MASUDA, D. 1999. Integrating knowledge resources at the point of care: opportunities for librarians. *Bull Med Libr Assoc,* 87**,** 393-403.

GALPERIN, M. Y. & COCHRANE, G. R. 2011. The 2011 Nucleic Acids Research Database Issue and the online Molecular Biology Database Collection. *Nucleic Acids Res,* 39**,** D1-6.

GARCIA-MOLINA, H., PAPAKONSTANTINOU, Y., QUASS, D., RAJARAMAN, A., SAGIV, Y., ULLMAN, J., VASSALOS, V. & WIDOM, J. 1997. The TSIMMIS Approach to Mediation: Data Models and Languages. *Journal of Intelligent Information Systems,* 8**,** 117-132.

GENEONTOLOGYCONSORTIUM 2001. Creating the gene ontology resource: design and implementation. *Genome Res,* 11**,** 1425-33.

GEO. 2004. *Kidney transplant rejection expression profiling* [Online]. Gene Expression Omnibus - NCBI. Available: http://www.ncbi.nlm.nih.gov/sites/GDSbrowser?acc=GDS724 [Accessed 03/15/2011 2011].

GO. 2011. *The Gene Ontolgy* [Online]. Available: http://www.geneontology.org/ [Accessed 20/03/2011 2011].

GOBLE, C., CORCHO, O., ALPER, P. & DE ROURE, D. 2006. e-Science and the Semantic Web: A symbiotic relationship. *Algorithmic Learning Theory, Proceedings,* 4264**,** 12-12.

GOBLE, C. & STEVENS, R. 2008. State of the nation in data integration for bioinformatics. *J Biomed Inform,* 41**,** 687-93.

GODARD, B., SCHMIDTKE, J., CASSIMAN, J. J. & AYME, S. 2003. Data storage and DNA banking for biomedical research: informed consent, confidentiality, quality issues, ownership, return of benefits. A professional perspective. *Eur J Hum Genet,* 11 Suppl 2**,** S88-122.

GONG, L., OWEN, R. P., GOR, W., ALTMAN, R. B. & KLEIN, T. E. 2008. PharmGKB: an integrated resource of pharmacogenomic data and knowledge. *Curr Protoc Bioinformatics,* Chapter 14**,** Unit14 7.

HAAS, L. M., SCHWARZ, P. M., KODALI, P., KOTLAR, E., RICE, J. E. & SWOPE, W. C. 2001. DiscoveryLink: a system for integrated access to life sciences data sources. *IBM Syst. J.,* 40**,** 489-511.

HALEVY, A. Y., IVES, Z. G., MORK, P. & TATARINOV, I. 2003. Piazza: data management infrastructure for semantic web applications. *Proceedings of the 12th international conference on World Wide Web.* Budapest, Hungary: ACM.

HEDELER, C., WONG, H. M., CORNELL, M. J., ALAM, I., SOANES, D. M., RATTRAY, M., HUBBARD, S. J., TALBOT, N. J., OLIVER, S. G. & PATON, N. W. 2007. e-Fungi: a data resource for comparative analysis of fungal genomes. *BMC Genomics,* 8**,** 426.

HEIMBIGNER, D. & MCLEOD, D. 1985. A federated architecture for information management. *ACM Trans. Inf. Syst.,* 3**,** 253-278.

HERNANDEZ, T. & KAMBHAMPATI, S. 2004. Integration of biological sources: current systems and challenges ahead. *ACM SIgmod Record,* 33**,** 51-60.

HODGSON, J. 2001. The headache of knowledge management. *Nat Biotechnol,* 19 Suppl**,** BE44-6.

HOMER, N., SZELINGER, S., REDMAN, M., DUGGAN, D., TEMBE, W., MUEHLING, J., PEARSON, J. V., STEPHAN, D. A., NELSON, S. F. & CRAIG, D. W. 2008. Resolving individuals contributing trace amounts of DNA to highly complex mixtures using high-density SNP genotyping microarrays. *PLoS Genet,* 4**,** e1000167.

HUBBARD, T., BARKER, D., BIRNEY, E., CAMERON, G., CHEN, Y., CLARK, L., COX, T., CUFF, J., CURWEN, V., DOWN, T., DURBIN, R., EYRAS, E., GILBERT, J., HAMMOND, M., HUMINIECKI, L., KASPRZYK, A., LEHVASLAIHO, H., LIJNZAAD, P., MELSOPP, C., MONGIN, E., PETTETT, R., POCOCK, M., POTTER, S., RUST, A., SCHMIDT, E., SEARLE, S., SLATER, G., SMITH, J., SPOONER, W., STABENAU, A., STALKER, J., STUPKA, E., URETA-VIDAL, A., VASTRIK, I. & CLAMP, M. 2002. The Ensembl genome database project. *Nucleic Acids Res,* 30**,** 38-41.

HULL, D., WOLSTENCROFT, K., STEVENS, R., GOBLE, C., POCOCK, M. R., LI, P. & OINN, T. 2006. Taverna: a tool for building and running workflows of services. *Nucleic Acids Res,* 34**,** W729-32.

HULL, R. 1997. Managing semantic heterogeneity in databases: a theoretical prospective. *Proceedings of the sixteenth ACM SIGACT-SIGMOD-SIGART symposium on Principles of database systems.* Tucson, Arizona, United States: ACM.

HWANG, D., RUST, A. G., RAMSEY, S., SMITH, J. J., LESLIE, D. M., WESTON, A. D., DE ATAURI, P., AITCHISON, J. D., HOOD, L., SIEGEL, A. F. & BOLOURI, H. 2005. A data integration methodology for systems biology. *Proceedings of the National Academy of Sciences of the United States of America,* 102**,** 17296-17301.

IMAI, T., MATSUDA, H., SEKIHARA, T., NAKANISHI, M. & HASHIMOTO, A. Year. Implementing an integrated system for heterogeneous molecular biology databases with intelligent agents. *In:* Communications, Computers and Signal Processing, 1997. '10 Years PACRIM 1987-1997 - Networking the Pacific Rim'. 1997 IEEE Pacific Rim Conference on, 1997. 807-810 vol.2.

INMON, W. H. 1992. *Building the data warehouse*, QED Technical Pub. Group.

JACKSON, M. 1995. *Software requirements & specifications*, ACM Press.

KARASAVVAS, K., BURGER, A. & BALDOCK, R. 2002. A Multi-agent Bioinformatics Integration System with Adjustable Autonomy. *In:* ISHIZUKA, M. & SATTAR, A. (eds.) *PRICAI 2002: Trends in Artificial Intelligence.* Springer Berlin / Heidelberg.

KARASAVVAS, K. A., BALDOCK, R. & BURGER, A. 2004. Bioinformatics integration and agent technology. *Journal of Biomedical Informatics,* 37**,** 205-219.

KAROLCHIK, D., BAERTSCH, R., DIEKHANS, M., FUREY, T. S., HINRICHS, A., LU, Y. T., ROSKIN, K. M., SCHWARTZ, M., SUGNET, C. W., THOMAS, D. J., WEBER, R. J., HAUSSLER, D. & KENT, W. J. 2003. The UCSC Genome Browser Database. *Nucleic Acids Res,* 31**,** 51-4.

KASPRZYK, A., KEEFE, D., SMEDLEY, D., LONDON, D., SPOONER, W., MELSOPP, C., HAMMOND, M., ROCCA-SERRA, P., COX, T. & BIRNEY, E. 2004. EnsMart: a generic system for fast and flexible access to biological data. *Genome Res,* 14**,** 160-9.

KERSEY, P., BOWER, L., MORRIS, L., HORNE, A., PETRYSZAK, R., KANZ, C., KANAPIN, A., DAS, U., MICHOUD, K., PHAN, I., GATTIKER, A., KULIKOVA, T., FARUQUE, N., DUGGAN, K., MCLAREN, P., REIMHOLZ, B., DURET, L., PENEL, S., REUTER, I. & APWEILER, R. 2005. Integr8 and Genome Reviews: integrated views of complete genomes and proteomes. *Nucleic Acids Res,* 33**,** D297-302.

KERSEY, P. J., MORRIS, L., HERMJAKOB, H. & APWEILER, R. 2003. Integr8: enhanced inter-operability of European molecular biology databases. *Methods Inf Med,* 42**,** 154-60.

KILIC, O., DOGAC, A. & EICHELBERG, M. 2010. Providing interoperability of eHealth communities through peer-to-peer networks. *IEEE Trans Inf Technol Biomed,* 14**,** 846-53.

KLEIN, T. E., CHANG, J. T., CHO, M. K., EASTON, K. L., FERGERSON, R., HEWETT, M., LIN, Z., LIU, Y., LIU, S., OLIVER, D. E., RUBIN, D. L., SHAFA, F., STUART, J. M. & ALTMAN, R. B. 2001. Integrating genotype and phenotype information: an overview of the PharmGKB project. Pharmacogenetics Research Network and Knowledge Base. *Pharmacogenomics J,* 1**,** 167-70.

KNOPPERS, B. M., JOLY, Y., SIMARD, J. & DUROCHER, F. 2006. The emergence of an ethical duty to disclose genetic research results: international perspectives. *Eur J Hum Genet,* 14**,** 1170-8.

KOHANE, I. S. 2000. Bioinformatics and clinical informatics: The imperative to collaborate. *Journal of the American Medical Informatics Association,* 7**,** 512-516.

KÖHLER, J. 2004. Integration of life science databases. *Drug Discovery Today: BIOSILICO,* 2**,** 9.

KÖHLER, J., PHILIPPI, S. & LANGE, M. 2003. SEMEDA: ontology based semantic integration of biological databases. *Bioinformatics,* 19**,** 2420-2427.

KOLATKAR, P. R., SAKHARKAR, M. K., TSE, C. R., KIONG, B. K., WONG, L., TAN, T. W. & SUBBIAH, S. 1998. Development of software tools at BioInformatics Centre (BIC) at the National University of Singapore (NUS). *Pac Symp Biocomput***,** 735-46.

KRALLINGER, M., VALENCIA, A. & HIRSCHMAN, L. 2008. Linking genes to literature: text mining, information extraction, and retrieval applications for biology. *Genome Biol,* 9 Suppl 2**,** S8.

KREUZTHALER, M., BLOICE, M. D., FAULSTICH, L., SIMONIC, K. M. & HOLZINGER, A. 2011. A comparison of different retrieval strategies working on medical free texts. *Journal of Universal Computer Science,* 17**,** 1109-1133.

KRISHNAMURTHY, R., LITWIN, W. & KENT, W. 1991. Language features for interoperability of databases with schematic discrepancies. *SIGMOD Rec.,* 20**,** 40-49.

KUHN, K. A., KNOLL, A., MEWES, H. W., SCHWAIGER, M., BODE, A., BROY, M., DANIEL, H., FEUSSNER, H., GRADINGER, R., HAUNER, H., HOFLER, H., HOLZMANN, B., HORSCH, A., KEMPER, A., KRCMAR, H., KOCHS, E. F., LANGE, R., LEIDL, R., MANSMANN, U., MAYR, E. W., MEITINGER, T., MOLLS, M., NAVAB, N., NUSSLIN, F., PESCHEL, C., REISER, M., RING, J., RUMMENY, E. J., SCHLICHTER, J., SCHMID, R., WICHMANN, H. E. & ZIEGLER, S. 2008. Informatics and medicine--from molecules to populations. *Methods Inf Med,* 47**,** 283-95.

KULIKOWSKI, C. A. 2002. The micro-macro spectrum of medical informatics challenges: from molecular medicine to transforming health care in a globalizing society. *Methods Inf Med,* 41**,** 20-4.

LAM, H. Y., MARENCO, L., CLARK, T., GAO, Y., KINOSHITA, J., SHEPHERD, G., MILLER, P., WU, E., WONG, G. T., LIU, N., CRASTO, C., MORSE, T., STEPHENS, S. & CHEUNG, K. H. 2007. AlzPharm: integration of neurodegeneration data using RDF. *BMC Bioinformatics,* 8 Suppl 3**,** S4.

LAMBRIX, P., STRÖMBÄCK, L. & TAN, H. 2009. Information Integration in Bioinformatics with Ontologies and Standards. *In:* BRY, F. & MALUSZYNSKI, J. (eds.) *Semantic Techniques for the Web.* Springer Berlin / Heidelberg.

LANDER, E. S., LINTON, L. M., BIRREN, B., NUSBAUM, C., ZODY, M. C., BALDWIN, J., DEVON, K., DEWAR, K., DOYLE, M., FITZHUGH, W., FUNKE, R., GAGE, D., HARRIS, K., HEAFORD, A., HOWLAND, J., KANN, L., LEHOCZKY, J., LEVINE, R., MCEWAN, P., MCKERNAN, K., MELDRIM, J., MESIROV, J. P., MIRANDA, C., MORRIS, W., NAYLOR, J., RAYMOND, C., ROSETTI, M., SANTOS, R., SHERIDAN, A., SOUGNEZ, C., STANGE-THOMANN, N., STOJANOVIC, N., SUBRAMANIAN, A., WYMAN, D., ROGERS, J., SULSTON, J., AINSCOUGH, R., BECK, S., BENTLEY, D., BURTON, J., CLEE, C., CARTER, N., COULSON, A., DEADMAN, R., DELOUKAS, P., DUNHAM, A., DUNHAM, I., DURBIN, R., FRENCH, L., GRAFHAM, D., GREGORY, S., HUBBARD, T., HUMPHRAY, S., HUNT, A., JONES, M., LLOYD, C., MCMURRAY, A., MATTHEWS, L., MERCER, S., MILNE, S., MULLIKIN, J. C., MUNGALL, A., PLUMB, R., ROSS, M., SHOWNKEEN, R., SIMS, S., WATERSTON, R. H., WILSON, R. K., HILLIER, L. W., MCPHERSON, J. D., MARRA, M. A., MARDIS, E. R., FULTON, L. A., CHINWALLA, A. T., PEPIN, K. H., GISH, W. R., CHISSOE, S. L., WENDL, M. C., DELEHAUNTY, K. D., MINER, T. L., DELEHAUNTY, A., KRAMER, J. B., COOK, L. L., FULTON, R. S., JOHNSON, D. L., MINX, P. J., CLIFTON, S. W., HAWKINS, T., BRANSCOMB, E., PREDKI, P., RICHARDSON, P., WENNING, S., SLEZAK, T., DOGGETT, N., CHENG, J. F., OLSEN, A., LUCAS, S., ELKIN, C., UBERBACHER, E., FRAZIER, M., et al. 2001. Initial sequencing and analysis of the human genome. *Nature,* 409**,** 860-921.

LEE, T. J., POULIOT, Y., WAGNER, V., GUPTA, P., STRINGER-CALVERT, D. W., TENENBAUM, J. D. & KARP, P. D. 2006. BioWarehouse: a bioinformatics database warehouse toolkit. *BMC Bioinformatics,* 7**,** 170.

LENZERINI, M. 2002. Data integration: a theoretical perspective. *Proceedings of the twenty-first ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems.* Madison, Wisconsin: ACM.

LEVY, A. Y., MENDELZON, A. O. & SAGIV, Y. 1995. Answering queries using views (extended abstract). *Proceedings of the fourteenth ACM SIGACT-SIGMOD-SIGART symposium on Principles of database systems.* San Jose, California, United States: ACM.

LITWIN, W., MARK, L. & ROUSSOPOULOS, N. 1990. Interoperability of multiple autonomous databases. *ACM Comput. Surv.,* 22**,** 267-293.

LOUIE, B., MORK, P., MARTIN-SANCHEZ, F., HALEVY, A. & TARCZY-HORNOCH, P. 2007. Data integration and genomic medicine. *Journal of Biomedical Informatics,* 40**,** 5-16.

LOVIS, C., COLAERT, D. & STROETMANN, V. N. 2008. DebugIT for patient safety - improving the treatment with antibiotics through multimedia data mining of heterogeneous clinical data. *Stud Health Technol Inform,* 136**,** 641-6.

LUDASCHER, B., GUPTA, A. & MARTONE, M. E. Year. Model-based mediation with domain maps. *In:* Data Engineering, 2001. Proceedings. 17th International Conference on, 2001 2001. 81-90.

MAIER, D., KALUS, W., WOLFF, M., KALKO, S. G., ROCA, J., MARIN DE MAS, I., TURAN, N., CASCANTE, M., FALCIANI, F., HERNANDEZ, M., VILLA-FREIXA, J. & LOSKO, S. 2011. Knowledge management for systems biology a general and visually driven framework applied to translational medicine. *BMC Syst Biol,* 5**,** 38.

MALIN, B. A. 2005. An evaluation of the current state of genomic data privacy protection technology and a roadmap for the future. *J Am Med Inform Assoc,* 12**,** 28-34.

MARENCO, L., WANG, R. & NADKARNI, P. 2009. Automated Database Mediation Using Ontological Metadata Mappings. *Journal of the American Medical Informatics Association,* 16**,** 723-737.

MARENCO, L., WANG, T. Y., SHEPHERD, G., MILLER, P. L. & NADKARNI, P. 2004. QIS: A framework for biomedical database federation. *J Am Med Inform Assoc,* 11**,** 523-34.

MARTIN-SANCHEZ, F. & MAOJO, V. 2009. Biomedical informatics and the convergence of Nano-Bio-Info-Cogno (NBIC) technologies. *Yearb Med Inform***,** 134-42.

MARTIN-SANCHEZ, F., MAOJO, V. & LOPEZ-CAMPOS, G. 2002. Integrating genomics into health information systems. *Methods Inf Med,* 41**,** 25-30.

MATTOS, N., KLEEWEIN, J., ROTH, M. & ZEIDENSTEIN, K. 1999. From object-relational to federated databases. *Invited Paper, in: AP Buchmann (Ed.): Datenbanksysteme in Büro, Technik und Wissenschaft.*

MCBRIDE, B. 2002. Jena: A semantic web toolkit. *IEEE Internet Computing,* 6**,** 55-58.

MCDONAGH, E. M., WHIRL-CARRILLO, M., GARTEN, Y., ALTMAN, R. B. & KLEIN, T. E. 2011. From pharmacogenomic knowledge acquisition to clinical applications: the PharmGKB as a clinical pharmacogenomic biomarker resource. *Biomark Med,* 5**,** 795-806.

MERALI, Z. & GILES, J. 2005. Databases in peril. *Nature,* 435**,** 1010-1.

MIN, H., MANION, F. J., GORALCZYK, E., WONG, Y.-N., ROSS, E. & BECK, J. R. 2009. Integration of prostate cancer clinical data using an ontology. *Journal of Biomedical Informatics,* 42**,** 1035-1045.

MORK, P., HALEVY, A. & TARCZY-HORNOCH, P. 2001. A model for data integration systems of biomedical data applied to online genetic databases. *Proc AMIA Symp***,** 473-7.

MUDUNURI, U., CHE, A., YI, M. & STEPHENS, R. M. 2009. bioDBnet: the biological database network. *Bioinformatics,* 25**,** 555-6.

MÜLLER, H., MICHOUX, N., BANDON, D. & GEISSBUHLER, A. 2004. A review of content-based image retrieval systems in medical applications—clinical benefits and future directions. *International Journal of Medical Informatics,* 73**,** 1-23.

NADKARNI, P. M., BRANDT, C., FRAWLEY, S., SAYWARD, F. G., EINBINDER, R., ZELTERMAN, D., SCHACTER, L. & MILLER, P. L. 1998. Managing attribute--value clinical trials data using the ACT/DB client-server database system. *J Am Med Inform Assoc,* 5**,** 139-51.

NETOMATIX. 2012. *HTMLParser.Net* [Online]. Available: http://www.netomatix.com/products/documentmanagement/HTMLParserNet.aspx [Accessed 14/02/2012 2012].

NIKOLAI, C. & MADEY, G. 2009. Tools of the trade: A survey of various agent based modeling platforms. *Journal of Artificial Societies and Social Simulation,* 12**,** 2.

NRC & RESEARCH, N. R. C. C. O. M. F. B. 1985. *Models for biomedical research: a new perspective*, National Academy Press.

OBAMA, B. 2007. Genomics and personalized medicine act of 2007. Washington, DC: U.S. Congress.

OLIVEIRA, J. L., DIAS, G., OLIVEIRA, I., ROCHA, P., HERMOSILLA, I., VICENTE, J., SPITERI, I., MARTIN-SÁNCHEZ, F. & PEREIRA, A. S. 2004. DiseaseCard: A Web-Based Tool for the Collaborative Integration of Genetic and Medical Information. *In:* BARREIRO, J. M., MARTIN-SANCHEZ, F., MAOJO, V. & SANZ, F. (eds.) *Biological and Medical Data Analysis.* Springer Berlin / Heidelberg.

OMIM. 2011. *Online Mendelian Inheritance in Man* [Online]. NCBI. Available: http://www.ncbi.nlm.nih.gov/omim [Accessed 20/03/2011 2011].

OVERBY, C. L., TARCZY-HORNOCH, P., HOATH, J. I., KALET, I. J. & VEENSTRA, D. L. 2010. Feasibility of incorporating genomic knowledge into electronic medical records for pharmacogenomic clinical decision support. *BMC Bioinformatics,* 11 Suppl 9**,** S10.

OZSU, M. T. 1991. *Principles of distributed database systems / M. Tamer Ozsu, Patrick Valduriez,* Englewood Cliffs, N.J. :, Prentice Hall.

PARK, J., PARK, B., JUNG, K., JANG, S., YU, K., CHOI, J., KONG, S., KIM, S., KIM, H., KIM, J. F., BLAIR, J. E., LEE, K., KANG, S. & LEE, Y. H. 2008. CFGP: a web-based, comparative fungal genomics platform. *Nucleic Acids Res,* 36**,** D562-71.

PASQUIER, C. 2008. Biological data integration using Semantic Web technologies. *Biochimie,* 90**,** 584-594.

PHILIPPI, S. 2004. Light-weight integration of molecular biological databases. *Bioinformatics,* 20**,** 51-57.

PHILIPPI, S. 2008. Data and knowledge integration in the life sciences. *Briefings in Bioinformatics,* 9**,** 451.

PHILIPPI, S. & KOHLER, J. 2006. Addressing the problems with life-science databases for traditional uses and systems biology. *Nat Rev Genet,* 7**,** 482-8.

POST, L. J. G., ROOS, M., MARSHALL, M. S., VAN DRIEL, R. & BREIT, T. M. 2007. A semantic web approach applied to integrative bioinformatics experimentation: a biological use case with genomics data. *Bioinformatics,* 23**,** 3080-3087.

PREECE, A., HUI, K., GRAY, A., MARTI, P., BENCH-CAPON, T., JONES, D. & CUI, Z. 2000. The KRAFT architecture for knowledge fusion and transformation. *Knowledge-Based Systems,* 13**,** 113-120.

PRESSMAN, R. S. 2010. *Software engineering: a practitioner's approach*, McGraw-Hill Higher Education.

QUAN, D. 2007. Improving life sciences information retrieval using semantic web technology. *Brief Bioinform,* 8**,** 172-82.

R. J. BAYARDO, J., BOHRER, W., BRICE, R., CICHOCKI, A., FOWLER, J., HELAL, A., KASHYAP, V., KSIEZYK, T., MARTIN, G., NODINE, M., RASHID, M., RUSINKIEWICZ, M., SHEA, R., UNNIKRISHNAN, C., UNRUH, A. & WOELK, D. 1997. InfoSleuth: agent-based semantic integration of information in open and dynamic environments. *SIGMOD Rec.,* 26**,** 195-206.

RACUSEN, L., RAYNER, D., TRPKOV, K., OLSEN, S. & SOLEZ, K. 1996. The Banff classification of renal allograft pathology: where do we go from here? *Transplant Proc,* 28**,** 486-8.

RAHM, E. & BERNSTEIN, P. A. 2001. A survey of approaches to automatic schema matching. *The VLDB Journal,* 10**,** 334-350.

REDDY, M. P., PRASAD, B. E., REDDY, P. G. & GUPTA, A. 1994. Methodology for integration of heterogeneous databases. *Ieee Transactions on Knowledge and Data Engineering,* 6**,** 920-933.

REENSKAUG, T. 1978. MVC Xerox PARC. Xerox.

RITTER, O., KOCAB, P., SENGER, M., WOLF, D. & SUHAI, S. 1994. Prototype implementation of the integrated genomic database. *Comput Biomed Res,* 27**,** 97-115.

ROBINSON, A. & RAHAYU, W. 2004. Genome Database Integration. *In:* LAGANÀ, A., GAVRILOVA, M. L., KUMAR, V., MUN, Y., TAN, C. J. K. & GERVASI, O. (eds.) *Computational Science and Its Applications – ICCSA 2004.* Springer Berlin / Heidelberg.

ROLDAN-GARCIA MDEL, M., NAVAS-DELGADO, I., KERZAZI, A., CHNIBER, O., MOLINA-CASTRO, J. & ALDANA-MONTES, J. F. 2009. KA-SB: from data integration to large scale reasoning. *BMC Bioinformatics,* 10 Suppl 10**,** S5.

RUTTENBERG, A., CLARK, T., BUG, W., SAMWALD, M., BODENREIDER, O., CHEN, H., DOHERTY, D., FORSBERG, K., GAO, Y., KASHYAP, V., KINOSHITA, J., LUCIANO, J., MARSHALL, M. S., OGBUJI, C., REES, J., STEPHENS, S., WONG, G. T., WU, E., ZACCAGNINI, D., HONGSERMEIER, T., NEUMANN, E., HERMAN, I. & CHEUNG, K. H. 2007. Advancing translational research with the Semantic Web. *BMC Bioinformatics,* 8 Suppl 3**,** S2.

RUTTENBERG, A., REES, J. A., SAMWALD, M. & MARSHALL, M. S. 2009. Life sciences on the Semantic Web: the Neurocommons and beyond. *Briefings in Bioinformatics,* 10**,** 193-204.

SCHADT, E. E., MONKS, S. A. & FRIEND, S. H. 2003. A new paradigm for drug discovery: integrating clinical, genetic, genomic and molecular phenotype data to identify drug targets. *Biochem Soc Trans,* 31**,** 437-43.

SCHOLLMEIER, R. Year. A definition of peer-to-peer networking for the classification of peer-to-peer architectures and applications. *In:* Peer-to-Peer Computing, 2001. Proceedings. First International Conference on, Aug 2001 2001. 101-102.

SHAH, S. P., HUANG, Y., XU, T., YUEN, M. M., LING, J. & OUELLETTE, B. F. 2005. Atlas - a data warehouse for integrative bioinformatics. *BMC Bioinformatics,* 6**,** 34.

SHETH, A. P. & LARSON, J. A. 1990. Federated database systems for managing distributed, heterogeneous, and autonomous databases. *ACM Comput. Surv.,* 22**,** 183-236.

SHVAIKO, P. & EUZENAT, J. 2005. A Survey of Schema-Based Matching Approaches

Journal on Data Semantics IV. *In:* SPACCAPIETRA, S. (ed.). Springer Berlin / Heidelberg.
SIEPEL, A., FARMER, A., TOLOPKO, A., ZHUANG, M., MENDES, P., BEAVIS, W. & SOBRAL, B. 2001. ISYS: a decentralized, component-based approach to the integration of heterogeneous bioinformatics resources. *Bioinformatics,* 17**,** 83-94.
SIMON, H. A. 1996. *The sciences of the artificial*, MIT Press.
SLOMAN, M. & LUPU, E. 2002. Security and management policy specification. *Network, IEEE,* 16**,** 10-19.
SMEDLEY, D., HAIDER, S., BALLESTER, B., HOLLAND, R., LONDON, D., THORISSON, G. & KASPRZYK, A. 2009. BioMart--biological queries made easy. *BMC Genomics,* 10**,** 22.
SMITH, A. K., CHEUNG, K. H., YIP, K. Y., SCHULTZ, M. & GERSTEIN, M. K. 2007a. LinkHub: a Semantic Web system that facilitates cross-database queries and information retrieval in proteomics. *BMC Bioinformatics,* 8 Suppl 3**,** S5.
SMITH, B., ASHBURNER, M., ROSSE, C., BARD, J., BUG, W., CEUSTERS, W., GOLDBERG, L. J., EILBECK, K., IRELAND, A., MUNGALL, C. J., LEONTIS, N., ROCCA-SERRA, P., RUTTENBERG, A., SANSONE, S. A., SCHEUERMANN, R. H., SHAH, N., WHETZEL, P. L. & LEWIS, S. 2007b. The OBO Foundry: coordinated evolution of ontologies to support biomedical data integration. *Nat Biotechnol,* 25**,** 1251-5.
SOLDATOVA, L. N. & KING, R. D. 2005. Are the current ontologies in biology good ontologies? *Nat Biotechnol,* 23**,** 1095-8.
SPELLMAN, P. T., MILLER, M., STEWART, J., TROUP, C., SARKANS, U., CHERVITZ, S., BERNHART, D., SHERLOCK, G., BALL, C., LEPAGE, M., SWIATEK, M., MARKS, W. L., GONCALVES, J., MARKEL, S., IORDAN, D., SHOJATALAB, M., PIZARRO, A., WHITE, J., HUBLEY, R., DEUTSCH, E., SENGER, M., ARONOW, B. J., ROBINSON, A., BASSETT, D., STOECKERT, C. J., JR. & BRAZMA, A. 2002. Design and implementation of microarray gene expression markup language (MAGE-ML). *Genome Biol,* 3**,** RESEARCH0046.
STEIN, L. D. 2003. Integrating biological databases. *Nat Rev Genet,* 4**,** 337-45.
STEVENS, R., BAKER, P., BECHHOFER, S., NG, G., JACOBY, A., PATON, N. W., GOBLE, C. A. & BRASS, A. 2000a. TAMBIS: transparent access to multiple bioinformatics information sources. *Bioinformatics,* 16**,** 184-5.
STEVENS, R., GOBLE, C. A. & BECHHOFER, S. 2000b. Ontology-based knowledge representation for bioinformatics. *Brief Bioinform,* 1**,** 398-414.
SUJANSKY, W. 2001. Heterogeneous Database Integration in Biomedicine. *Journal of Biomedical Informatics,* 34**,** 285-298.
SWEENEY, L. 1997. Guaranteeing anonymity when sharing medical data, the Datafly System. *Proc AMIA Annu Fall Symp***,** 51-5.
SZALAY, A. & GRAY, J. 2006. 2020 computing: science in an exponential world. *Nature,* 440**,** 413-4.
TAYLOR, C. F., FIELD, D., SANSONE, S. A., AERTS, J., APWEILER, R., ASHBURNER, M., BALL, C. A., BINZ, P. A., BOGUE, M., BOOTH, T., BRAZMA, A., BRINKMAN, R. R., MICHAEL CLARK, A., DEUTSCH, E. W., FIEHN, O., FOSTEL, J., GHAZAL, P., GIBSON, F., GRAY, T., GRIMES, G., HANCOCK, J. M., HARDY, N. W., HERMJAKOB, H., JULIAN, R. K., JR., KANE, M., KETTNER, C., KINSINGER, C., KOLKER, E., KUIPER, M., LE NOVERE, N., LEEBENS-MACK, J., LEWIS, S. E., LORD, P., MALLON, A. M., MARTHANDAN, N., MASUYA, H., MCNALLY, R., MEHRLE, A., MORRISON, N., ORCHARD, S., QUACKENBUSH, J., REECY, J. M., ROBERTSON, D. G., ROCCA-SERRA, P., RODRIGUEZ, H., ROSENFELDER, H., SANTOYO-LOPEZ, J., SCHEUERMANN, R. H., SCHOBER, D., SMITH, B., SNAPE, J., STOECKERT, C. J., JR., TIPTON, K., STERK, P., UNTERGASSER, A., VANDESOMPELE, J. & WIEMANN, S. 2008. Promoting coherent minimum reporting guidelines for biological and biomedical investigations: the MIBBI project. *Nat Biotechnol,* 26**,** 889-96.
TAYLOR, C. F., PATON, N. W., LILLEY, K. S., BINZ, P. A., JULIAN, R. K., JR., JONES, A. R., ZHU, W., APWEILER, R., AEBERSOLD, R., DEUTSCH, E. W., DUNN, M. J., HECK, A. J., LEITNER, A., MACHT, M., MANN, M., MARTENS, L., NEUBERT, T. A., PATTERSON, S. D., PING, P.,

SEYMOUR, S. L., SOUDA, P., TSUGITA, A., VANDEKERCKHOVE, J., VONDRISKA, T. M., WHITELEGGE, J. P., WILKINS, M. R., XENARIOS, I., YATES, J. R., 3RD & HERMJAKOB, H. 2007. The minimum information about a proteomics experiment (MIAPE). *Nat Biotechnol,* 25**,** 887-93.

THORN, C. F., KLEIN, T. E. & ALTMAN, R. B. 2010. Pharmacogenomics and bioinformatics: PharmGKB. *Pharmacogenomics,* 11**,** 501-5.

TOBIAS, R. & HOFMANN, C. 2004. Evaluation of free Java-libraries for social-scientific agent based simulation. *Journal of Artificial Societies and Social Simulation,* 7.

TOPEL, T., KORMEIER, B., KLASSEN, A. & HOFESTADT, R. 2008. BioDWH: a data warehouse kit for life science data integration. *J Integr Bioinform,* 5.

TRISSL, S., ROTHER, K., MULLER, H., STEINKE, T., KOCH, I., PREISSNER, R., FROMMEL, C. & LESER, U. 2005. Columba: an integrated database of proteins, structures, and annotations. *BMC Bioinformatics,* 6**,** 81.

VAN HOYWEGHEN, I. & HORSTMAN, K. 2008. European practices of genetic information and insurance: lessons for the Genetic Information Nondiscrimination Act. *JAMA,* 300**,** 326-7.

VENTER, J. C., ADAMS, M. D., MYERS, E. W., LI, P. W., MURAL, R. J., SUTTON, G. G., SMITH, H. O., YANDELL, M., EVANS, C. A., HOLT, R. A., GOCAYNE, J. D., AMANATIDES, P., BALLEW, R. M., HUSON, D. H., WORTMAN, J. R., ZHANG, Q., KODIRA, C. D., ZHENG, X. H., CHEN, L., SKUPSKI, M., SUBRAMANIAN, G., THOMAS, P. D., ZHANG, J., GABOR MIKLOS, G. L., NELSON, C., BRODER, S., CLARK, A. G., NADEAU, J., MCKUSICK, V. A., ZINDER, N., LEVINE, A. J., ROBERTS, R. J., SIMON, M., SLAYMAN, C., HUNKAPILLER, M., BOLANOS, R., DELCHER, A., DEW, I., FASULO, D., FLANIGAN, M., FLOREA, L., HALPERN, A., HANNENHALLI, S., KRAVITZ, S., LEVY, S., MOBARRY, C., REINERT, K., REMINGTON, K., ABU-THREIDEH, J., BEASLEY, E., BIDDICK, K., BONAZZI, V., BRANDON, R., CARGILL, M., CHANDRAMOULISWARAN, I., CHARLAB, R., CHATURVEDI, K., DENG, Z., DI FRANCESCO, V., DUNN, P., EILBECK, K., EVANGELISTA, C., GABRIELIAN, A. E., GAN, W., GE, W., GONG, F., GU, Z., GUAN, P., HEIMAN, T. J., HIGGINS, M. E., JI, R. R., KE, Z., KETCHUM, K. A., LAI, Z., LEI, Y., LI, Z., LI, J., LIANG, Y., LIN, X., LU, F., MERKULOV, G. V., MILSHINA, N., MOORE, H. M., NAIK, A. K., NARAYAN, V. A., NEELAM, B., NUSSKERN, D., RUSCH, D. B., SALZBERG, S., SHAO, W., SHUE, B., SUN, J., WANG, Z., WANG, A., WANG, X., WANG, J., WEI, M., WIDES, R., XIAO, C., YAN, C., et al. 2001. The sequence of the human genome. *Science,* 291**,** 1304-51.

VILLAZÓN-TERRAZAS, B., GÓMEZ-PÉREZ, A. & CALBIMONTE, J. P. Year. NOR2O: a Library for Transforming Non-Ontological Resources to Ontologies. *In:* Extended Semantic Web Conference (ESWC 2009), 2010 2009. Springer.

WANG, X., GORLITSKY, R. & ALMEIDA, J. S. 2005. From XML to RDF: how semantic web technologies will change the design of 'omic' standards. *Nat Biotechnol,* 23**,** 1099-103.

WEINSTEIN, J. N. 2002. 'Omic' and hypothesis-driven research in the molecular pharmacology of cancer. *Curr Opin Pharmacol,* 2**,** 361-5.

WIDOM, J. 1995. Research problems in data warehousing. *Proceedings of the fourth international conference on Information and knowledge management.* Baltimore, Maryland, United States: ACM.

WIEDERHOLD, G. 1992. Mediators in the architecture of future information systems. *Computer,* 25**,** 38-49.

WIEDERHOLD, G. 1996. *Intelligent integration of information*, Kluwer.

WILLIAMS, N. 1997. How to get databases talking the same language. *Science,* 275**,** 301-2.

WONG, L. 1995. The Kleisli/CPL Extensible Query Optimizer Programmer Guide. Singapure: Institute of Systems Science.

WONG, L. 2002. Technologies for integrating biological data. *Briefings in Bioinformatics,* 3**,** 389-404.

XWOSW 2012. XWiki – Open Source Wiki and Content-Oriented Application Platform.

YAF. 2012. *Yet Another Forum Open Source .Net forum* [Online]. Available: http://yetanotherforum.net/ [Accessed 10/02/2012 2012].

ZAPLETAL, E., RODON, N., GRABAR, N. & DEGOULET, P. Year. Methodology of integration of a clinical data warehouse with a clinical information system: The HEGP case. *In*, 2010. 193-197.

ZDOBNOV, E. M., LOPEZ, R., APWEILER, R. & ETZOLD, T. 2002. The EBI SRS server--recent developments. *Bioinformatics,* 18**,** 368-73.

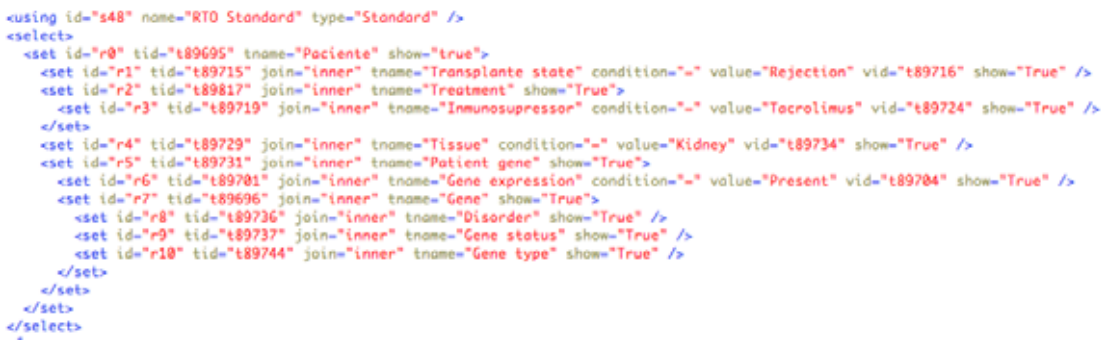
ZHANG, Z., CHEUNG, K.-H. & TOWNSEND, J. P. 2009. Bringing Web 2.0 to bioinformatics. *Briefings in Bioinformatics,* 10**,** 1-10.

ANEXO I

Este anexo describe los resultados obtenidos en la comparación de las consultas del sistema de integración de pacientes con trasplante de riñón (escenario número 3) con su equivalente en el caso de esas mismas consultas ejecutadas contra una base de datos relacional con todos los datos disponibles. El objetivo de estas pruebas es mostrar la complejidad que puede tener el tipo de consulta realizada contra una base de datos relacional, comparada con la facilidad de uso del sistema de integración, así como comprobar la correcta recuperación de datos por parte del sistema de integración.

En las siguientes tablas se describen las distintas consultas de prueba, su tamaño, así como su equivalente en SQL y la visualización de parte de los resultados.

- Prueba 1: Obtener los genes infra-expresados en la muestra de los pacientes tratados con el fármaco Tacrolimus y cuyo estado patológico sea rechazo agudo del trasplante, en los que la muestra se haya tomado de riñón.
    - Consulta QIS:



    - Consulta SQL

```
SELECT
patient,tissue,individual,inmunosupression,id_ref,abs_call,ENTREZ_GENE_
ID,gene_status,disorder,type FROM (SELECT
patient,tissue,individual,id_ref,abs_call,ENTREZ_GENE_ID,inmunosupressi
on FROM (SELECT
pac_data_inm.patient,tissue,individual,id_ref,value,abs_call,inmunosupr
ession FROM (SELECT pac_data.patient,tissue,individual,inmunosupression
FROM (SELECT * FROM `Pacientes`.`clinic_data` WHERE individual = 'Acute
Rejection' AND tissue = 'kidney') pac_data LEFT OUTER JOIN
`Pacientes`.`Inmunosupression` inmuno ON pac_data.patient =
inmuno.patient WHERE inmunosupression = 'Tacrolimus') pac_data_inm LEFT
OUTER JOIN (SELECT * FROM `Pacientes`.`Genetic_Data` WHERE abs_call =
```

```
'Present') gen_data ON pac_data_inm.patient = gen_data.patient)
patient_gene_data LEFT OUTER JOIN `Pacientes`.`microarray` microarray
ON patient_gene_data.id_ref = microarray.id) pacientes LEFT OUTER JOIN
(SELECT gene_status,gene_dis.geneID,disorder,type FROM (SELECT
location,gene_status,gene_symbol,genemap.mim_id,comments,geneID,disorde
r FROM `OMIM`.`genemap` genemap INNER JOIN `OMIM`.`morbid_map`
morbidmap ON genemap.mim_id = morbidmap.mim_id) gene_dis INNER JOIN
`OMIM`.`mim2gene` mim2gene ON gene_dis.mim_id = mim2gene.mim_id) omim
ON pacientes.ENTREZ_GENE_ID = omim.geneID
```

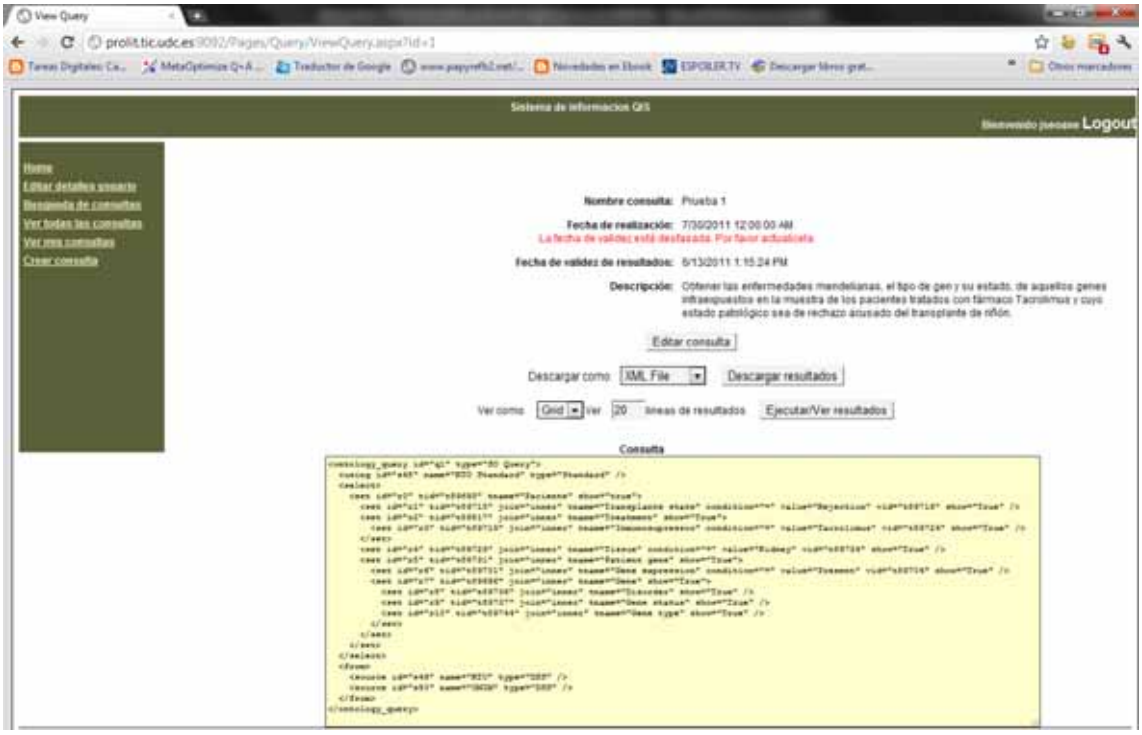


Figura 49: Consulta generada prueba 1

- Resultado: En la Figura 50 se muestra una captura de pantalla de una parte del conjunto de genes asociado a esas características. En el conjunto de datos recuperados aparecen únicamente dos de los pacientes del conjunto del estudio cumplen estos requisitos.

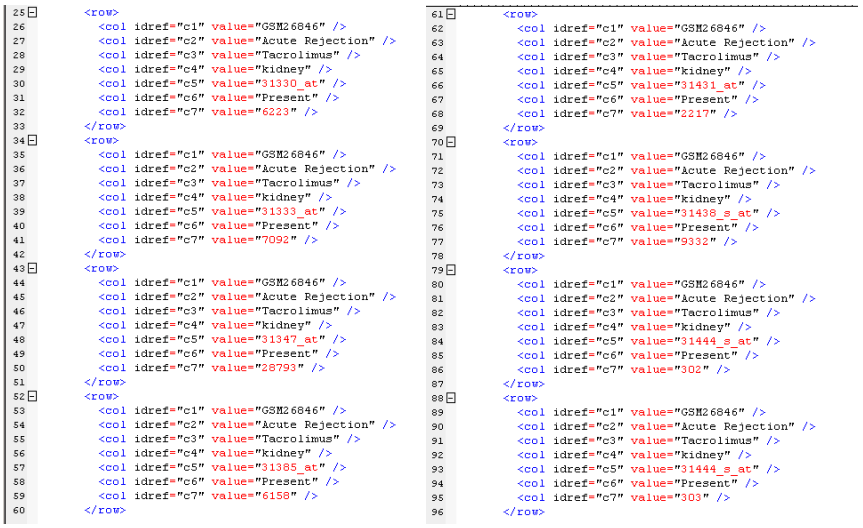


Figura 50: Resultados prueba 1

- Prueba 2: Obtener el nombre del gen y la función molecular, de los genes sobre-expresados, de los pacientes cuyo estado patológico sea normal después de recibir un trasplante y su nivel de creatinina no supere el valor de 1,2.
    - Consulta QIS

```
<using id="s48" name="RTO Standard" type="Standard" />
<select>
  <set id="r0" tid="t89695" tname="Paciente" show="true">
    <set id="r1" tid="t89727" join="inner" tname="Serum creatinine" condition="&lt;" value="1.2" show="True" />
    <set id="r2" tid="t89715" join="inner" tname="Transplante state" condition="=" value="Well Functional" vid="t89718" show="True" />
    <set id="r3" tid="t89729" join="inner" tname="Tissue" condition="=" value="Kidney" vid="t89734" show="True" />
    <set id="r4" tid="t89731" join="inner" tname="Patient gene" show="True">
      <set id="r5" tid="t89701" join="inner" tname="Gene expression" condition="=" value="Marginal" vid="t89703" show="True" />
      <set id="r6" tid="t89696" join="inner" tname="Gene" show="True">
        <set id="r7" tid="t89699" join="inner" tname="Biological process" show="True" />
        <set id="r8" tid="t89700" join="inner" tname="Celular component" show="True" />
        <set id="r9" tid="t89725" join="inner" tname="Molecular function" show="True" />
        <set id="r10" tid="t89736" join="inner" tname="Disorder" show="True" />
        <set id="r12" tid="t89737" join="inner" tname="Gene status" show="True" />
        <set id="r13" tid="t89744" join="inner" tname="Gene type" show="True" />
      </set>
    </set>
  </set>
</select>
```

    - Consulta SQL

```
SELECT * FROM (SELECT
patient,tissue,individual,id_ref,abs_call,ENTREZ_GENE_ID,GO_BIOLOGICAL_
PROCESS,GO_MOLECULAR_FUNCTION,GO_CELLULAR_COMPONENT FROM (SELECT
pac_data_inm.patient,tissue,individual,id_ref,value,abs_call FROM
(SELECT pac_data.patient,tissue,individual FROM (SELECT * FROM
`Pacientes`.`clinic_data` WHERE individual = 'Well-Functioning' AND
tissue = 'kidney' AND serum_creatinine<1.2) pac_data LEFT OUTER JOIN
`Pacientes`.`Inmunosupression` inmuno ON pac_data.patient =
inmuno.patient) pac_data_inm LEFT OUTER JOIN (SELECT * FROM
`Pacientes`.`Genetic_Data` WHERE abs_call = 'Marginal') gen_data ON
pac_data_inm.patient = gen_data.patient) patient_gene_data LEFT OUTER
JOIN `Pacientes`.`microarray` microarray ON patient_gene_data.id_ref =
microarray.id) pacientes LEFT OUTER JOIN (SELECT
gene_status,gene_dis.geneID,disorder,type FROM (SELECT
location,gene_status,gene_symbol,genemap.mim_id,comments,geneID,disorde
```

```
r FROM `OMIM`.`genemap` genemap INNER JOIN `OMIM`.`morbid_map`
morbidmap ON genemap.mim_id = morbidmap.mim_id) gene_dis INNER JOIN
`OMIM`.`mim2gene` mim2gene ON gene_dis.mim_id = mim2gene.mim_id) omim
ON pacientes.ENTREZ_GENE_ID = omim.geneID
```

- Resultado: En la Figura 51se muestran una parte de los resultados, que consisten en los genes de los pacientes con trasplante exitoso con el nivel de creatinina alto. Únicamente tres de los pacientes cumplen estas características.

Figura 51: Resultados prueba 2

- Prueba 3: Obtener los genes infra-expresados de los pacientes tratados con prednisona y que hayan mostrado rechazo agudo.
  - Consulta QIS

```
<using id="s48" name="RTO Standard" type="Standard" />
<select>
  <set id="r0" tid="t89695" tname="Paciente" show="true">
    <set id="r1" tid="t89715" join="inner" tname="Transplante state" condition="=" value="Rejection" vid="t89716" show="True" />
    <set id="r2" tid="t89817" join="inner" tname="Treatment" show="True">
      <set id="r3" tid="t89719" join="inner" tname="Inmunosupressor" condition="=" value="Prednisone" vid="t89722" show="True" />
    </set>
    <set id="r4" tid="t89731" join="inner" tname="Patient gene" show="True">
      <set id="r5" tid="t89701" join="inner" tname="Gene expression" condition="=" value="Present" vid="t89704" show="True" />
      <set id="r6" tid="t89696" join="inner" tname="Gene" show="True">
        <set id="r8" tid="t89736" join="inner" tname="Disorder" show="True" />
        <set id="r9" tid="t89737" join="inner" tname="Gene status" show="True" />
        <set id="r10" tid="t89744" join="inner" tname="Gene type" show="True" />
      </set>
    </set>
  </set>
</select>
```

  - Consulta SQL:

```
SELECT
patient,tissue,individual,inmunosupression,id_ref,abs_call,ENTREZ_G
ENE_ID,gene_status,disorder,type FROM (SELECT
patient,tissue,individual,inmunosupression,id_ref,abs_call,ENTREZ_G
```

```
ENE_ID FROM (SELECT pac_data_inm.patient,tissue,individual,inmunosupression,id_ref,valu e,abs_call FROM (SELECT pac_data.patient,tissue,individual,inmunosupression FROM (SELECT * FROM `Pacientes`.`clinic_data` WHERE individual = 'Acute Rejection') pac_data LEFT OUTER JOIN `Pacientes`.`Inmunosupression` inmuno ON pac_data.patient = inmuno.patient WHERE inmuno.inmunosupression = 'Prednisone') pac_data_inm LEFT OUTER JOIN (SELECT * FROM `Pacientes`.`Genetic_Data` WHERE abs_call='Present') gen_data ON pac_data_inm.patient = gen_data.patient) patient_gene_data LEFT OUTER JOIN `Pacientes`.`microarray` microarray ON patient_gene_data.id_ref = microarray.id) pacientes LEFT OUTER JOIN (SELECT gene_status,gene_dis.geneID,disorder,type FROM (SELECT location,gene_status,gene_symbol,genemap.mim_id,comments,geneID,dis order FROM `OMIM`.`genemap` genemap INNER JOIN `OMIM`.`morbid_map` morbidmap ON genemap.mim_id = morbidmap.mim_id) gene_dis INNER JOIN `OMIM`.`mim2gene` mim2gene ON gene_dis.mim_id = mim2gene.mim_id) omim ON pacientes.ENTREZ_GENE_ID = omim.geneID
```

- Resultados: En la Figura 52 se muestra una parte de los resultados que consisten en un conjunto de genes infra-expresados. Dichos datos afectan a 13 pacientes que presentan rechazo agudo.

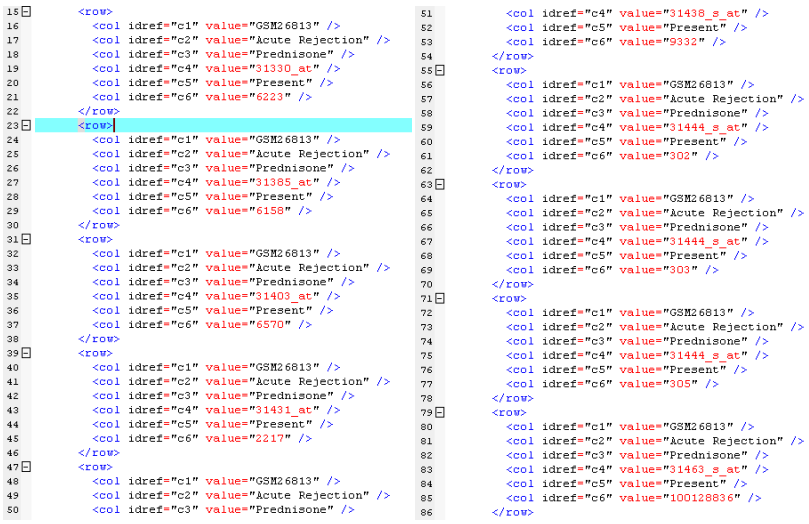


Figura 52: Resultados prueba 3

- Prueba 4: Obtener todas las enfermedades en las que los genes de los pacientes que presenten rechazo presenten infra-expresión.
  - Consulta QIS

```
<using id="s48" name="RTO Standard" type="Standard" />
<select>
  <set id="r0" tid="t89695" tname="Paciente" show="true">
    <set id="r1" tid="t89726" join="inner" tname="Age" show="True" />
    <set id="r2" tid="t89698" join="inner" tname="days_post_transplant" show="True" />
    <set id="r3" tid="t89707" join="inner" tname="Histopathology" show="True" />
    <set id="r4" tid="t89727" join="inner" tname="Serum creatinine" show="True" />
    <set id="r5" tid="t89728" join="inner" tname="Sex" show="True" />
    <set id="r6" tid="t89729" join="inner" tname="Tissue" show="True" />
    <set id="r7" tid="t89715" join="inner" tname="Transplante state" condition="=" value="Rejection" vid="t89716" show="True" />
    <set id="r8" tid="t89817" join="inner" tname="Treatment" show="True">
      <set id="r10" tid="t89719" join="inner" tname="Inmunosupressor" show="True" />
    </set>
    <set id="r9" tid="t89731" join="inner" tname="Patient gene" show="True">
      <set id="r11" tid="t89701" join="inner" tname="Gene expression" condition="=" value="Marginal" vid="t89703" show="True" />
      <set id="r12" tid="t89730" join="inner" tname="Value" show="True" />
      <set id="r13" tid="t89696" join="inner" tname="Gene" show="True">
        <set id="r14" tid="t89699" join="inner" tname="Biological process" show="True" />
        <set id="r15" tid="t89700" join="inner" tname="Celular component" show="True" />
        <set id="r16" tid="t89725" join="inner" tname="Molecular Function" show="True" />
        <set id="r17" tid="t89736" join="inner" tname="Disorder" show="True" />
        <set id="r18" tid="t89739" join="inner" tname="Extraction method" show="True" />
        <set id="r19" tid="t89737" join="inner" tname="Gene status" show="True" />
        <set id="r20" tid="t89705" join="inner" tname="Gene_symbol" show="True" />
        <set id="r21" tid="t89738" join="inner" tname="Location" show="True" />
      </set>
    </set>
  </set>
</select>
```

- Consulta SQL:

```
SELECT * FROM (SELECT
patient,tissue,individual,age,sex,histopathology,serum_creatinine,inmun
osupression,days_post_transplant,id_ref,value,abs_call,gene_title,gene_
symbol,ENTREZ_GENE_ID,GO_BIOLOGICAL_PROCESS,GO_CELLULAR_COMPONENT,GO_MO
LECULAR_FUNCTION FROM (SELECT
clinic_data.patient,tissue,individual,age,sex,histopathology,serum_crea
tinine,days_post_transplant,inmunosupression,id_ref,value,abs_call FROM
(SELECT
clinic.patient,tissue,individual,age,sex,histopathology,serum_creatinin
e,days_post_transplant,inmunosupression FROM `Pacientes`.`clinic_data`
clinic LEFT OUTER JOIN `Pacientes`.`inmunosupression` inmuno ON
clinic.patient = inmuno.patient WHERE individual = 'Acute Rejection')
clinic_data LEFT OUTER JOIN `Pacientes`.`Genetic_data` genetic_data ON
clinic_data.patient = genetic_data.patient WHERE
genetic_data.abs_call='Marginal') patient_genetic_data LEFT OUTER JOIN
`Pacientes`.`microarray` microarray ON patient_genetic_data.id_ref =
microarray.id) patient_data  LEFT OUTER JOIN ( SELECT
location,title,method,comments,genemap.gene_symbol,geneID,disorder,type
FROM `OMIM`.`genemap` genemap INNER JOIN `OMIM`.`morbid_map` morbidmap
INNER JOIN `OMIM`.`mim2gene` mim2gene ON genemap.mim_id =
morbidmap.mim_id AND genemap.mim_id = mim2gene.mim_id) omim ON
patient_data.ENTREZ_GENE_ID = omim.geneID
```

- Resultado: Se obtuvieron todas las enfermedades relacionadas con la infra-expresión de los genes relacionados con el estudio de los pacientes con rechazo.

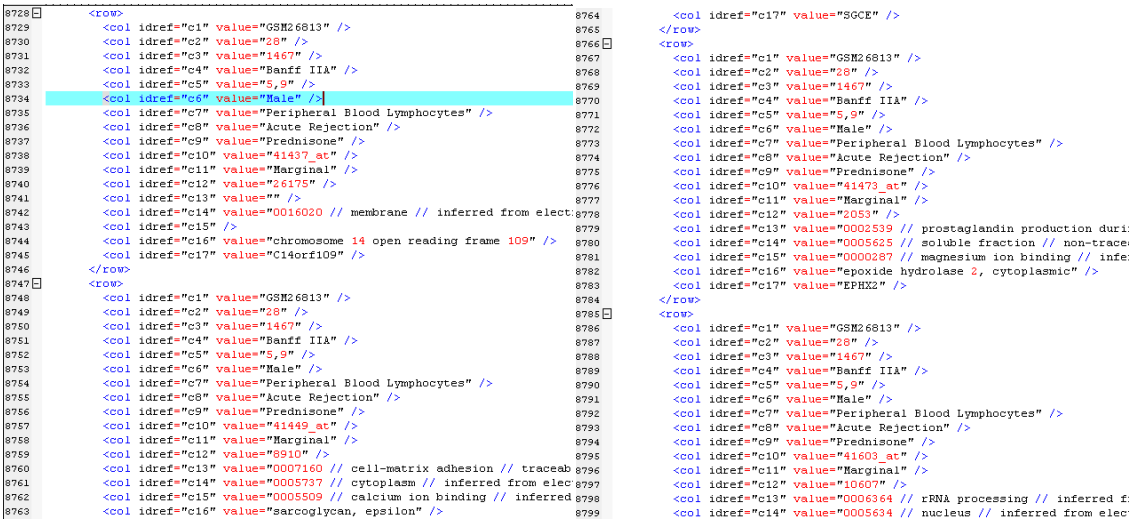


Figura 53: Resultado prueba 4

- Prueba 5: Obtener los genes de los pacientes que presenten rechazo agudo Banf II y un nivel de creatinina superior a 1
    - Consulta QIS

```
<using id="s48" name="RTO Standard" type="Standard" />
<select>
  <set id="r0" tid="t89695" tname="Paciente" show="true">
    <set id="r1" tid="t89726" join="inner" tname="Age" show="True" />
    <set id="r2" tid="t89707" join="inner" tname="Histopathology" condition="=" value="Banff IIA" vid="t89711" show="True" />
    <set id="r3" tid="t89727" join="inner" tname="Serum creatinine" condition="&gt;" value="1" show="True" />
    <set id="r4" tid="t89728" join="inner" tname="Sex" show="True" />
    <set id="r5" tid="t89729" join="inner" tname="Tissue" show="True" />
    <set id="r6" tid="t89715" join="inner" tname="Transplante state" show="True" />
    <set id="r7" tid="t89731" join="inner" tname="Patient gene" show="True">
      <set id="r8" tid="t89701" join="inner" tname="Gene expression" show="True" />
      <set id="r9" tid="t89696" join="inner" tname="Gene" show="True">
        <set id="r10" tid="t89699" join="inner" tname="Biological process" show="True" />
        <set id="r11" tid="t89700" join="inner" tname="Celular component" show="True" />
        <set id="r12" tid="t89736" join="inner" tname="Disorder" show="True" />
        <set id="r13" tid="t89706" join="inner" tname="Gene name" show="True" />
        <set id="r14" tid="t89737" join="inner" tname="Gene status" show="True" />
        <set id="r15" tid="t89744" join="inner" tname="Gene type" show="True" />
        <set id="r16" tid="t89738" join="inner" tname="Location" show="True" />
        <set id="r17" tid="t89725" join="inner" tname="Molecular Function" show="True" />
      </set>
    </set>
  </set>
</select>
```

    - Consulta SQL:

```
SELECT * FROM (SELECT
patient,tissue,individual,id_ref,abs_call,ENTREZ_GENE_ID,GO_BIOLOGI
CAL_PROCESS,GO_MOLECULAR_FUNCTION,GO_CELLULAR_COMPONENT FROM
(SELECT
pac_data_inm.patient,tissue,individual,id_ref,value,abs_call FROM
(SELECT pac_data.patient,tissue,individual FROM (SELECT * FROM
`Pacientes`.`clinic_data` WHERE histopathology = 'Banff IIA' AND
serum_creatinine>1) pac_data LEFT OUTER JOIN
`Pacientes`.`Inmunosupression` inmuno ON pac_data.patient =
inmuno.patient) pac_data_inm LEFT OUTER JOIN (SELECT * FROM
```

```
`Pacientes`.`Genetic_Data`) gen_data ON pac_data_inm.patient =
gen_data.patient) patient_gene_data LEFT OUTER JOIN
`Pacientes`.`microarray` microarray ON patient_gene_data.id_ref =
microarray.id) pacientes LEFT OUTER JOIN (SELECT
gene_status,gene_dis.geneID,disorder,type,location FROM (SELECT
location,gene_status,gene_symbol,genemap.mim_id,comments,geneID,dis
order FROM `OMIM`.`genemap` genemap INNER JOIN `OMIM`.`morbid_map`
morbidmap ON genemap.mim_id = morbidmap.mim_id) gene_dis INNER JOIN
`OMIM`.`mim2gene` mim2gene ON gene_dis.mim_id = mim2gene.mim_id)
omim ON pacientes.ENTREZ_GENE_ID = omim.geneID
```

- Resultado: Se recupera la información de los genes de los 3 pacientes que presentan rechazo agudo Banff II y nivel de creatinina superior a 2

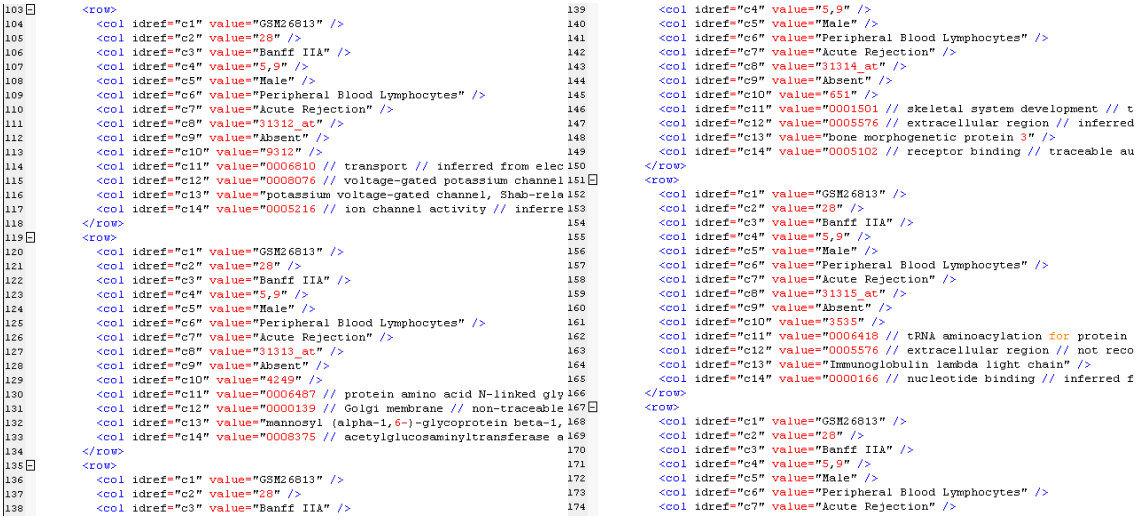


Figura 54: Resultados prueba 5

- Prueba 6: Obtener las enfermedades asociadas y los procesos biológicos de los genes sobre-expresados para los pacientes que presenten Glomuroesclerosis focal segmentada, cuya muestra haya sido obtenida del riñón, así como el tratamiento que han recibido los pacientes en este caso.
  - Consulta QIS

```
<using id="s48" name="RTO Standard" type="Standard" />
<select>
  <set id="r0" tid="t89695" tname="Paciente" show="true">
    <set id="r1" tid="t89707" join="inner" tname="Histopathology" condition="=" value="Focal Segment Glom" vid="t89713" show="True" />
    <set id="r2" tid="t89729" join="inner" tname="Tissue" condition="=" value="Kidney" vid="t89734" show="True" />
    <set id="r3" tid="t89817" join="inner" tname="Treatment" show="True">
      <set id="r4" tid="t89719" join="inner" tname="Inmunosupressor" show="True" />
    </set>
    <set id="r5" tid="t89731" join="inner" tname="Patient gene" show="True">
      <set id="r6" tid="t89701" join="inner" tname="Gene expression" condition="=" value="Present" vid="t89704" show="True" />
      <set id="r7" tid="t89696" join="inner" tname="Gene" show="True">
        <set id="r8" tid="t89699" join="inner" tname="Biological process" show="True" />
        <set id="r9" tid="t89708" join="inner" tname="Celular component" show="True" />
        <set id="r10" tid="t89725" join="inner" tname="Molecular Function" show="True" />
        <set id="r11" tid="t89736" join="inner" tname="Disorder" show="True" />
        <set id="r12" tid="t89737" join="inner" tname="Gene status" show="True" />
        <set id="r13" tid="t89744" join="inner" tname="Gene type" show="True" />
        <set id="r14" tid="t89738" join="inner" tname="Location" show="True" />
        <set id="r15" tid="t89812" join="inner" tname="comments" show="True" />
      </set>
    </set>
  </set>
</select>
```

- Consulta SQL:

```
SELECT * FROM (SELECT patient,tissue,individual,histopathology,inmunosupression,id_ref,value,abs_call,GO_BIOLOGICAL_PROCESS,GO_MOLECULAR_FUNCTION,GO_CELLULAR_COMPONENT,ENTREZ_GENE_ID FROM (SELECT pac_data_inm.patient,tissue,individual,histopathology,inmunosupression,id_ref,value,abs_call FROM (SELECT pac_data.patient,tissue,individual,histopathology,inmunosupression FROM (SELECT * FROM `Pacientes`.`clinic_data` WHERE histopathology = 'Focal Segmental Glomerulosclerosis' AND tissue = 'kidney') pac_data LEFT OUTER JOIN `Pacientes`.`Inmunosupression` inmuno ON pac_data.patient = inmuno.patient) pac_data_inm LEFT OUTER JOIN (SELECT * FROM `Pacientes`.`Genetic_Data` WHERE abs_call = 'Present') gen_data ON pac_data_inm.patient = gen_data.patient) patient_gene_data LEFT OUTER JOIN `Pacientes`.`microarray` microarray ON patient_gene_data.id_ref = microarray.id) pacientes LEFT OUTER JOIN (SELECT location,gene_status,gene_symbol,genemap.mim_id,comments,geneID,disorder FROM `OMIM`.`genemap` genemap INNER JOIN `OMIM`.`morbid_map` morbidmap ON genemap.mim_id = morbidmap.mim_id) omim ON pacientes.ENTREZ_GENE_ID = omim.geneID
```

- Resultado: Se han recuperado la información asociada a los genes sobre expresados y a las enfermedades asociadas con ese gen del único paciente que cumple con los requisitos de la consulta

Figura 55: Resultados prueba 6

- Prueba 7: Obtener la información relativa a los genes relacionados con los pacientes trasplantados que han sido tratados con Sirolimus y cuyos datos genéticos fueses obtenidos a partir de una biopsia.
    - Consulta QIS:

    - Consulta SQL:

```
SELECT * FROM (SELECT
patient,tissue,individual,histopathology,serum_creatinine,inmunosup
ression,pac_gen.id_ref,abs_call,GO_BIOLOGICAL_PROCESS,GO_MOLECULAR_
FUNCTION,ENTREZ_GENE_ID FROM (SELECT
pac.patient,tissue,individual,histopathology,serum_creatinine,inmun
osupression,id_ref,abs_call FROM(SELECT
cli.patient,tissue,individual,histopathology,serum_creatinine,inmun
osupression FROM `Pacientes`.`clinic_data` cli INNER JOIN
`Pacientes`.`Inmunosupression` inmu ON cli.patient = inmu.patient
WHERE inmunosupression = 'Sirolimus' AND tissue = 'kidney') pac
LEFT OUTER JOIN `Pacientes`.`genetic_data` genetic_data ON
```

```
pac.patient = genetic_data.patient) pac_gen LEFT OUTER JOIN
`Pacientes`.`microarray` micro ON pac_gen.id_ref = micro.id)
pacientes LEFT OUTER JOIN (SELECT om.geneID,type,disorder,title
FROM (SELECT
location,gene_status,gene_symbol,genemap.mim_id,comments,geneID,dis
order,title FROM `OMIM`.`genemap` genemap INNER JOIN
`OMIM`.`morbid_map` morbidmap ON genemap.mim_id = morbidmap.mim_id)
om INNER JOIN `OMIM`.`mim2gene` mim2gene ON om.mim_id =
mim2gene.mim_id) omim ON pacientes.ENTREZ_GENE_ID = omim.geneID
```

- Resultados: Se obtuvo la información genética (genes y funciones del GO) de los tres pacientes que cumplen con estas características en el estudio.

Figura 56: Resultados prueba 7

- Prueba 8: Obtener las enfermedades genéticas asociadas a los genes sobre-expresados en biopsias de pacientes trasplantados que hayan sido tratados con Ciclofosfarina.
  - Consultas QIS

```
<ontology_query id="q1" type="SO Query">
  <using id="s48" name="RTO Standard" type="Standard" />
  <select>
    <set id="r0" tid="t89695" tname="Paciente" show="true">
      <set id="r1" tid="t89726" join="inner" tname="Age" show="True" />
      <set id="r2" tid="t89698" join="inner" tname="days_post_transplant" show="True" />
      <set id="r3" tid="t89707" join="inner" tname="Histopathology" show="True" />
      <set id="r4" tid="t89727" join="inner" tname="Serum creatinine" show="True" />
      <set id="r5" tid="t89728" join="inner" tname="Sex" show="True" />
      <set id="r6" tid="t89729" join="inner" tname="Tissue" condition="=" value="Kidney" vid="t89734" show="True" />
      <set id="r7" tid="t89817" join="inner" tname="Treatment" show="True">
        <set id="r8" tid="t89719" join="inner" tname="Inmunosupressor" condition="=" value="Cyclosporine" vid="t89720" show="True" />
      </set>
      <set id="r9" tid="t89731" join="inner" tname="Patient gene" show="True">
        <set id="r10" tid="t89701" join="inner" tname="Gene expression" condition="-" value="Absent" vid="t89702" show="True" />
        <set id="r12" tid="t89696" join="inner" tname="Gene" show="True">
          <set id="r13" tid="t89699" join="inner" tname="Biological process" show="True" />
          <set id="r14" tid="t89700" join="inner" tname="Celular component" show="True" />
          <set id="r15" tid="t89736" join="inner" tname="Disorder" show="True" />
          <set id="r16" tid="t89812" join="inner" tname="comments" show="True" />
          <set id="r17" tid="t89737" join="inner" tname="Gene status" show="True" />
          <set id="r18" tid="t89706" join="inner" tname="Gene name" show="True" />
        </set>
      </set>
    </set>
  </select>
  <from>
    <source id="s49" name="RTO" type="DSS" />
    <source id="s50" name="OMIM" type="DSS" />
  </from>
</ontology_query>
```

- Consulta SQL:

```
SELECT * FROM (SELECT patient,age,histopathology,serum_creatinine,days_post_transplant,inmunosupression,id_ref,abs_call,GO_BIOLOGICAL_PROCESS,GO_CELLULAR_COMPONENT,ENTREZ_GENE_ID FROM (SELECT pacientes.patient,age,histopathology,serum_creatinine,days_post_transplant,inmunosupression,id_ref,abs_call FROM(SELECT inmuno.patient,age,histopathology,serum_creatinine,days_post_transplant,inmunosupression FROM `Pacientes`.`clinic_data` clinic_data INNER JOIN `Pacientes`.`inmunosupression` inmuno ON clinic_data.patient = inmuno.patient WHERE inmunosupression = 'Cyclosporine') pacientes INNER JOIN `Pacientes`.`Genetic_data` genetic_Data ON pacientes.patient = genetic_Data.patient WHERE abs_call = 'Absent') gen_data LEFT OUTER JOIN `Pacientes`.`microarray` microarray ON gen_data.id_ref = microarray.id) pacientes LEFT OUTER JOIN (SELECT gene_status,title,genemap.mim_id,comments,geneID,disorder FROM `OMIM`.`genemap` genemap INNER JOIN `OMIM`.`morbid_map` morbidmap ON genemap.mim_id = morbidmap.mim_id) omim ON pacientes.ENTREZ_GENE_ID = omim.geneID
```

- Resultado: Se obtienen las enfermedades asociadas a los genes sobre-expresados de los 4 pacientes que han sido tratados con Ciclofosfatina.

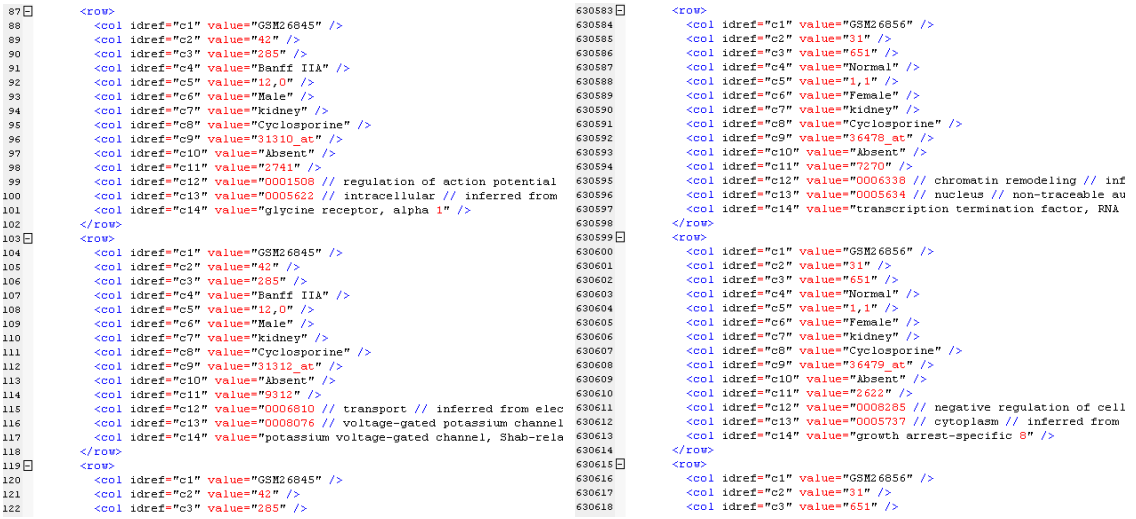


Figura 57: Resultados prueba 8

- Prueba 9: Obtener los genes sobre expresados de los pacientes que presenten histopatología grado Banff Borderline y cuyos datos hayan sido obtenidos de biopsia.
    - Consulta QIS:

```
<ontology_query id="q1" type="SO Query">
  <using id="s48" name="RTO Standard" type="Standard" />
  <select>
    <set id="r0" tid="t89695" tname="Paciente" show="true">
      <set id="r1" tid="t89726" join="inner" tname="Age" show="True" />
      <set id="r2" tid="t89728" join="inner" tname="Sex" show="True" />
      <set id="r3" tid="t89715" join="inner" tname="Transplante state" show="True" />
      <set id="r4" tid="t89729" join="inner" tname="Tissue" condition="=" value="Kidney" vid="t89734" show="True" />
      <set id="r5" tid="t89707" join="inner" tname="Histopathology" condition="=" value="Banff borderline" vid="t89709" show="True" />
      <set id="r6" tid="t89817" join="inner" tname="Treatment" show="True">
        <set id="r7" tid="t89719" join="inner" tname="Inmunosupressor" show="True" />
      </set>
      <set id="r8" tid="t89731" join="inner" tname="Patient gene" show="True">
        <set id="r9" tid="t89701" join="inner" tname="Gene expression" condition="=" value="Present" vid="t89704" show="True" />
        <set id="r10" tid="t89696" join="inner" tname="Gene" show="True">
          <set id="r11" tid="t89699" join="inner" tname="Biological process" show="True" />
          <set id="r12" tid="t89706" join="inner" tname="Gene name" show="True" />
          <set id="r13" tid="t89705" join="inner" tname="Gene_symbol" show="True" />
          <set id="r14" tid="t89738" join="inner" tname="Location" show="True" />
        </set>
      </set>
    </set>
  </select>
  <from>
    <source id="s49" name="RTO" type="DSS" />
    <source id="s50" name="OMIM" type="DSS" />
  </from>
</ontology_query>
```

    - Consulta SQL:

```
SELECT * FROM (SELECT
clinic_gen.patient,tissue,individual,age,sex,histopathology,inmunos
upression,id,abs_call,GO_BIOLOGICAL_PROCESS,ENTREZ_GENE_ID FROM
(SELECT
clinic.patient,tissue,individual,age,sex,histopathology,inmunosupre
ssion,id_ref,abs_call FROM (SELECT c.patient,
tissue,individual,age,sex,histopathology,inmunosupression FROM
`Pacientes`.`clinic_data` c INNER JOIN
```

```
`Pacientes`.`inmunosupression` i ON c.patient = i.patient WHERE
tissue = 'kidney' AND histopathology = 'Banff borderline') clinic
INNER JOIN `Pacientes`.`genetic_data` g ON clinic.patient =
g.patient WHERE abs_call = 'Present') clinic_gen LEFT OUTER JOIN
`Pacientes`.`microarray` mc ON clinic_gen.id_ref = mc.id) pacientes
LEFT OUTER JOIN ( SELECT location,gene_symbol,title,disorder,geneID
FROM `OMIM`.`genemap` gm LETF OUTER JOIN `OMIM`.`morbid_map` mm ON
gm.mim_id = mm.mim_id) omim ON pacientes.ENTREZ_GENE_ID =
omim.geneID
```

- Resultado: Se recuperan los genes y la información asociada a los genes sobre expresados de los cuatro pacientes con la patología de riñón señalada en la consulta.

Figura 58: Resultados prueba 9

- Prueba 10: Obtener los genes y la información asociada a ellos en el caso de que estén sobre-expresados en pacientes con un nivel de creatinina superior a 6 y con una necrosis tubular aguda, así como el tratamiento asociado a estos pacientes.
  - Consulta QIS:

```
<ontology_query id="q1" type="SO Query">
  <using id="s48" name="RTO Standard" type="Standard" />
  <select>
    <set id="r0" tid="t89695" tname="Paciente" show="true">
      <set id="r1" tid="t89726" join="inner" tname="Age" show="True" />
      <set id="r2" tid="t89698" join="inner" tname="days_post_transplant" show="True" />
      <set id="r3" tid="t89707" join="inner" tname="Histopathology" condition="=" value="Acute tubular necrosis" vid="t89708" show="True" />
      <set id="r4" tid="t89727" join="inner" tname="Serum creatinine" condition="&gt;" value="6" show="True" />
      <set id="r5" tid="t89728" join="inner" tname="Sex" show="True" />
      <set id="r6" tid="t89729" join="inner" tname="Tissue" show="True" />
      <set id="r7" tid="t89715" join="inner" tname="Transplante state" show="True" />
      <set id="r8" tid="t89817" join="inner" tname="Treatment" show="True">
        <set id="r9" tid="t89719" join="inner" tname="Inmunosupressor" show="True" />
      </set>
      <set id="r10" tid="t89731" join="inner" tname="Patient gene" show="True">
        <set id="r11" tid="t89701" join="inner" tname="Gene expression" condition="=" value="Present" vid="t89704" show="True" />
        <set id="r12" tid="t89696" join="inner" tname="Gene" show="True">
          <set id="r13" tid="t89699" join="inner" tname="Biological process" show="True" />
          <set id="r14" tid="t89725" join="inner" tname="Molecular Function" show="True" />
          <set id="r15" tid="t89812" join="inner" tname="comments" show="True" />
          <set id="r16" tid="t89736" join="inner" tname="Disorder" show="True" />
          <set id="r17" tid="t89706" join="inner" tname="Gene name" show="True" />
          <set id="r18" tid="t89737" join="inner" tname="Gene status" show="True" />
          <set id="r19" tid="t89705" join="inner" tname="Gene_symbol" show="True" />
          <set id="r20" tid="t89738" join="inner" tname="Location" show="True" />
        </set>
      </set>
    </set>
  </select>
  <from>
    <source id="s49" name="RTO" type="DSS" />
    <source id="s50" name="OMIM" type="DSS" />
  </from>
</ontology_query>
```

- Consulta SQL:

```
SELECT * FROM (SELECT cli_gen.patient,tissue,individual,age,sex,histopathology,serum_creatinine,inmunosupression,id_ref,abs_call,ENTREZ_GENE_ID,GO_BIOLOGICAL_PROCESS,GO_MOLECULAR_FUNCTION FROM (SELECT cli.patient,tissue,individual,age,sex,histopathology,serum_creatinine,inmunosupression,id_ref,abs_call FROM (SELECT c.patient,tissue,individual,age,sex,histopathology,serum_creatinine,inmunosupression FROM `Pacientes`.`clinic_data` c LEFT OUTER JOIN `Pacientes`.`inmunosupression` i ON c.patient = i.patient WHERE serum_creatinine >6 AND histopathology = 'Acute tubular necrosis') cli LEFT OUTER JOIN `Pacientes`.`genetic_data` gd ON cli.patient = gd.patient WHERE abs_call = 'Present') cli_gen LEFT OUTER JOIN `Pacientes`.`microarray` mc ON cli_gen.id_ref = mc.id) pacientes LEFT OUTER JOIN (SELECT location,gene_symbol,title,disorder,geneID,gene_status,comments FROM `OMIM`.`genemap` gm INNER JOIN `OMIM`.`morbid_map` mm ON gm.mim_id = mm.mim_id) omim ON pacientes.ENTREZ_GENE_ID = omim.geneID
```

- Resultado: Bajo estos parámetros únicamente hay cuatro pacientes, se han recuperado los datos asociados a los genes sobre expresados.

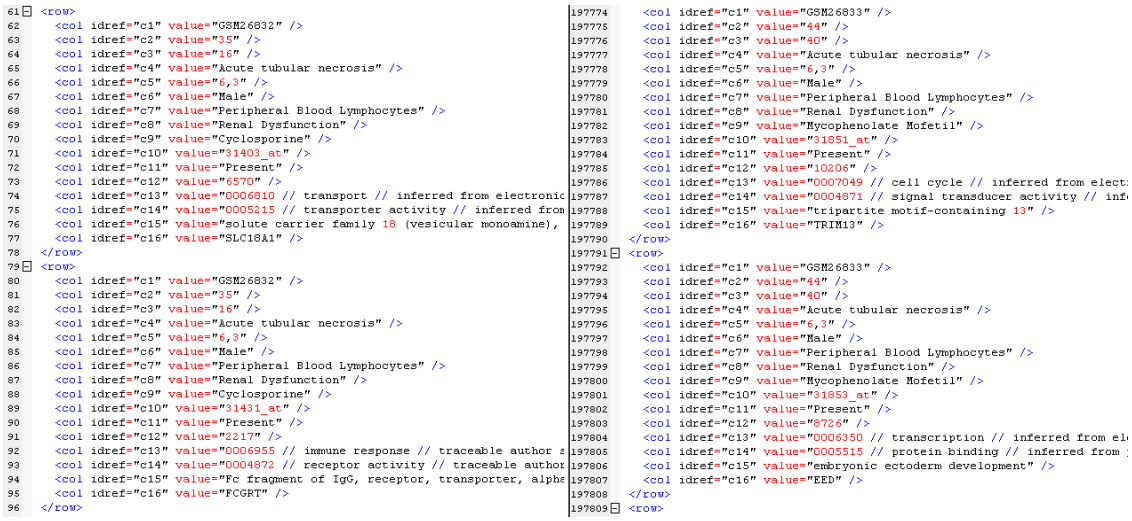


Figura 59: Resultados prueba 10

Para cada una de las pruebas se han comprobado que los resultados obtenidos tanto en la salida XML como en la salida MDB eran equivalentes a los resultados de la consulta realizada a mano.