



Universidad Cenfotec
Maestría en Tecnología de Bases de Datos

Documento final de Proyecto de Investigación
Aplicada 2

**Gobernabilidad sobre los objetos de bases
de datos heterogéneas, mediante el
diccionario de datos empresarial**

Elaborado por:
Mora Varela, Brian

Agosto, 2015

DECLARACIÓN JURADA

Yo, Brian Mora Varela, mayor, soltero, estudiante de la carrera de Maestría Profesional en Tecnologías de Bases de Datos de la universidad CENFOTEC, domiciliado en San pablo de Barva en Heredia, portador de la cédula de identidad 1-1219-0916, en este acto, debidamente apercibido y entendido de las penas y consecuencias con las que se castiga en el código penal, el delito de perjurio ante quienes se constituyen en el Tribunal Examinador de mi trabajo de tesis para optar por el título de Maestría Profesional en Tecnologías de Bases de Datos, juro solemnemente que mi trabajo de investigación, titulado **Gobernabilidad sobre los objetos de bases de datos heterogéneas, mediante el diccionario de datos empresarial** es una obra original, que ha respetado todo lo preceptuado por las leyes Penales, así como la Ley de Derechos de Autor y Derechos Conexos, número 6683 de 25 noviembre de 1982 y sus reformas, publicado en la gaceta número 226 de 25 noviembre de 1982, incluyendo el numeral 70 de dicha ley que advierte: Artículo 70: Es permitido citar a un autor transcribiendo los pasajes pertinentes, siempre que estos no sean tantos y seguidos, que pueda considerarse como una producción simulada y sustancial, que redunde en perjuicio del autor y de la obra original. En fe de lo anterior firmo en la ciudad de San José, el 24 del mes de agosto del año 2015. Autorizo la consulta de este trabajo con fines exclusivamente académicos.

Brian Mora Varela
Cédula: 1-1219-0916

DEDICATORIA

Este trabajo se lo dedico a **Dios**, por darme le regalo de la vida y por el permitirme desarrollarme en mi carrera profesional y personal.

A **mi abuela Argery Araya Cascante**, por haberme ayudado siempre de forma incondicional cuando más lo necesité.

A **Claudio Araya Cascante**, por ser un ejemplo a seguir, por las enseñanzas y valores que lo han caracterizado durante toda su vida.

A **mis profesores y compañeros**, por haberme dado la oportunidad de aprender junto a ellos.

“La dicha de la vida consiste en tener siempre algo que hacer, alguien a quien amar y alguna cosa que esperar”. **Thomas Chalmers.**

AGRADECIMIENTOS

Agradezco a **Dios** que me permitió la realización de este trabajo; a **mi familia**, por brindarme el apoyo incondicional mientras yo realizaba las investigaciones; a **los profesores**, por ser los guías y a los **compañeros** en la aventura del aprendizaje.

Tabla de Contenido

Índice de tablas	14
Índice de figuras	17
1. Capítulo 1: Introducción	21
1.1. Antecedentes del problema.....	21
1.2. Justificación	23
1.3. Viabilidad.....	25
1.3.1. Factibilidad técnica.....	25
1.3.2. Factibilidad económica.....	25
1.3.3. Factibilidad operacional	30
1.4. Planteamiento del problema.....	31
1.4.1. Problema general	31
1.4.2. Subproblemas del proyecto	31
1.5. Objetivos del proyecto	31
1.5.1. Objetivo general	32
1.5.2. Objetivos específicos.....	32
1.6. Propuesta de solución	32
1.7. Cronograma de desarrollo de la propuesta.....	34
1.8. Marco de referencia organizacional y socioeconómico.....	38
1.8.1. Reseña histórica.....	38

1.8.2.	Misión.....	40
1.8.3.	Visión	41
1.8.4.	Organigrama	41
1.8.5.	Valores de la organización	43
1.9.	Alcances y limitaciones del proyecto	44
1.9.1.	Alcance	44
1.9.2.	Limitaciones	44
1.10.	Estado de la cuestión	44
1.10.1.	Planificación	45
1.10.1.1.	Formulación de la pregunta	45
	Objetivo	45
	Problema	45
	Pregunta	45
	Palabras clave	45
	Intervención	45
	Población	46
	Aplicación.....	46
1.10.1.2.	Selección de recursos.....	46
	Identificación de fuentes.....	46
	Definición de los criterios de selección.....	46
	Verificación de referencias	48
1.10.1.3.	Selección de estudios.....	48
	Procedimiento para la selección de estudios	48
1.10.2.	Ejecución	48

Ejecución de la selección.....	48
1.10.3. Análisis de resultados	49
2. Capítulo II: Marco Teórico	51
2.1. Diccionario de datos	51
2.1.1. Metadatos	52
2.1.2. Tipos de metadatos	53
2.2. Gobernabilidad de los datos.....	53
2.3. Administración de metadatos.....	53
2.4. Base de datos federada.....	54
2.4.1. Definición	54
2.4.2. Características.....	54
2.4.3. Diferencia entre bases de datos distribuidas y bases de datos federadas.....	55
2.5. Modelamiento de datos	56
2.6. Almacén de datos y sus características	57
2.7. Procesos de extracción, transformación y carga	58
2.8. Data Mart	58
2.9. Diseño multidimensional	59
2.10. Cubos multidimensionales	61
2.11. Área de datos de pruebas.....	61
2.12. Área de datos operacional	62
2.13. Herencia en el modelo relacional	62
2.14. SQL Incorporado.....	63
3. Capítulo III: Marco Metodológico.....	64
3.1. Tipo de investigación.....	64

3.2.	Alcance de la investigación	64
3.3.	Enfoque de la investigación.....	64
3.4.	Sujetos y fuentes de información.....	67
3.4.1.	Fuentes de información	67
3.4.2.	Sujetos de información	68
3.5.	Descripción de instrumentos.....	68
4.	Capítulo IV: Desarrollo de Prototipo	69
4.1.	Metodología seleccionada para el desarrollo del proyecto	69
4.2.	Selección de miembros del equipo de trabajo.....	70
4.3.	Definición de objetivos	72
4.4.	Definición de requerimientos.....	72
4.5.	Diseño y modelización.....	73
4.5.1.	Modelamiento de la arquitectura	74
4.5.2.	Modelado de datos.....	77
4.5.2.1.	Diseño - D001 - Servidores, plataformas, gestores e instancias.....	78
4.5.2.1.1.	Especificación del diseño	78
4.5.2.1.2.	Propuesta del diseño.....	78
4.5.2.1.2.1.	Modelo conceptual	78
4.5.2.1.2.2.	Modelo lógico	80
4.5.2.1.2.3.	Modelo físico.....	86
4.5.2.1.3.	Justificación del diseño	86
4.5.2.2.	Diseño D002. Administración de bases de datos y esquemas por instancia	88
4.5.2.2.1.	Especificación del diseño	88
4.5.2.2.2.	Propuesta del diseño.....	89

4.5.2.2.2.1.	Modelo conceptual	89
4.5.2.2.2.2.	Modelo lógico	91
4.5.2.2.2.3.	Modelo físico.....	96
4.5.2.2.3.	Justificación del diseño	96
4.5.2.3.	Diseño D003. Procesos por instancia	98
4.5.2.3.1.	Especificación del diseño	98
4.5.2.3.2.	Propuesta del diseño.....	99
4.5.2.3.2.1.	Modelo conceptual	99
4.5.2.3.2.2.	Modelo lógico	101
4.5.2.3.2.3.	Modelo físico.....	105
4.5.2.3.3.	Justificación del diseño	105
4.5.2.4.	Diseño D004. Registro de colaboradores	107
4.5.2.4.1.	Especificación del diseño	107
4.5.2.4.2.	Propuesta del diseño.....	108
4.5.2.4.2.1.	Modelo conceptual	108
4.5.2.4.2.2.	Modelo lógico	110
4.5.2.4.2.3.	Modelo físico.....	116
4.5.2.4.3.	Justificación del diseño	116
4.5.2.5.	Diseño D005. Propietarios por objeto	117
4.5.2.5.1.	Especificación del diseño	117
4.5.2.5.2.	Propuesta del diseño.....	117
4.5.2.5.2.1.	Modelo conceptual	117
4.5.2.5.2.2.	Modelo lógico	120
4.5.2.5.2.3.	Modelo físico.....	124

4.5.2.5.3.	Justificación del diseño	125
4.5.2.6.	Diseño D006. Comentarios por propietario.....	126
4.5.2.6.1.	Especificación del diseño	126
4.5.2.6.2.	Propuesta del diseño.....	127
4.5.2.6.2.1.	Modelo conceptual	127
4.5.2.6.2.2.	Modelo lógico	129
4.5.2.6.2.3.	Modelo físico.....	133
4.5.2.6.3.	Justificación del diseño	134
4.5.2.7.	Diseño D007. Conceptos por objeto.....	134
4.5.2.7.1.	Especificación del diseño	134
4.5.2.7.2.	Propuesta del diseño.....	135
4.5.2.7.2.1.	Modelo conceptual	135
4.5.2.7.2.2.	Modelo lógico	136
4.5.2.7.2.3.	Modelo físico.....	137
4.5.2.7.3.	Justificación del diseño	138
4.5.2.8.	Diseño D008. Controles por objeto	140
4.5.2.8.1.	Especificación del diseño	140
4.5.2.8.2.	Propuesta del diseño.....	141
4.5.2.8.2.1.	Modelo conceptual	141
4.5.2.8.2.2.	Modelo lógico	142
4.5.2.8.2.3.	Modelo físico.....	144
4.5.2.8.3.	Justificación del diseño	144
4.5.2.9.	Diseño D009. Servidores por ubicación geográfica	144
4.5.2.9.1.	Especificación del diseño	144

4.5.2.9.2.	Propuesta del diseño.....	145
4.5.2.9.2.1.	Modelo conceptual	145
4.5.2.9.2.2.	Modelo lógico	146
4.5.2.9.2.3.	Modelo físico.....	148
4.5.2.9.3.	Justificación del diseño	149
4.5.2.10.	Diseño D010. Administradores por servidores.....	150
4.5.2.10.1.	Especificación del diseño	150
4.5.2.10.2.	Propuesta del diseño.....	150
4.5.2.10.2.1.	Modelo conceptual	150
4.5.2.10.2.2.	Modelo lógico.....	151
4.5.2.10.2.3.	Modelo físico.....	153
4.5.2.10.3.	Justificación del diseño	153
4.5.2.11.	Diseño D011. Administradores de bases de datos por instancia.....	154
4.5.2.11.1.	Especificación del diseño	154
4.5.2.11.2.	Propuesta del diseño.....	154
4.5.2.11.2.1.	Modelo conceptual	154
4.5.2.11.2.2.	Modelo lógico.....	155
4.5.2.11.2.3.	Modelo físico.....	158
4.5.2.11.3.	Justificación del diseño	158
4.5.3.	Modelamiento de procesos para la integración de datos	159
4.5.3.1.	Proceso de extracción	159
4.5.3.2.	Proceso de transformación.....	163
4.5.3.3.	Proceso de carga	164
4.5.4.	Diseño multidimensional.....	165

4.5.4.1.	Modelo conceptual	165
4.5.4.2.	Modelo físico de datos.....	167
4.5.4.2.1.1.	Dimensiones regiones y países.....	168
4.5.4.2.1.2.	Dimensión conceptos	168
4.5.4.2.1.3.	Dimensión fechas	169
4.5.4.2.1.4.	Dimensión controles.....	170
4.5.4.2.1.5.	Dimensión comentarios.....	171
4.5.4.2.1.6.	Dimensiones propietarios titulares y propietarios delegados.....	172
4.5.4.2.1.7.	Dimensiones esquemas, bases de datos, instancias de bases de datos y servidores	173
4.5.4.2.1.8.	Dimensión objetos de base de datos.....	174
4.5.4.2.1.9.	Dimensión clases de objetos de bases de datos.....	175
4.5.1.	Justificación del diseño multidimensional.....	176
4.5.2.	Cubo multidimensional	176
4.5.3.	Recomendación de la herramienta para la visualización de los datos	177
4.5.4.	Modelamiento de pruebas.....	178
4.6.	Construcción del prototipo.....	178
4.7.	Evaluación del prototipo	178
4.7.1.	Ejecución de pruebas	179
4.7.2.	Análisis de los resultados de las pruebas.....	179
5.	Capítulo V: Comparación de herramientas	182
5.1.	Comparación de herramientas.....	182
5.2.	Ventajas y desventajas del desarrollo local de la solución de <i>software</i>	185
6.	Capítulo VI: Conclusiones, recomendaciones y trabajo futuro.....	187
6.1.	Conclusiones	187

Objetivo general	187
Objetivos específicos.....	187
6.2. Recomendaciones	189
6.3. Trabajo futuro	190
Referencias	191
Apéndices.....	196

Índice de tablas

Tabla 1. Costos estimados (elaboración propia del investigador)	26
Tabla 2. Flujos de efectivo proyectados (elaboración propia del investigador).....	29
Tabla 3. Cronograma del proyecto.	37
Tabla 4. Etapas para el desarrollo de modelos de datos (Kriegel, 2011).	57
Tabla 5. Comparativa entre los sistemas operacionales y analíticos, adaptado de Adamson (Adamson, 2010).	59
Tabla 6. Criterios de evaluación del modelo de datos.	65
Tabla 7 – Matriz RACI (elaboración propia del investigador).....	71
Tabla 8. Plan de comunicación (elaboración propia del investigador).....	72
Tabla 9. Simbología empleada en los diagramas de datos.	77
Tabla 10. Descripción de las entidades del modelo lógico para el diseño D001. Servidores, plataformas, gestores e instancias.....	79
Tabla 11. Descripción de las relaciones entre entidades para el diseño D001. Servidores, plataformas, gestores e instancias.....	80
Tabla 12. Modelo de datos lógico para el diseño D001. Servidores, plataformas, gestores e instancias.	81
Tabla 13. Descripción de las entidades del modelo lógico para el diseño D002. Administración de bases de datos y esquemas por instancia.....	89
Tabla 14. Descripción de las relaciones entre entidades para el diseño D002. Administración de bases de datos y esquemas por instancia.....	90

Tabla 15. Modelo de datos lógico para el diseño D002. Administración de bases de datos y esquemas por instancia.	91
Tabla 16. Descripción de las entidades del modelo lógico para el diseño D003. Procesos por instancia.	99
Tabla 17. Descripción de las relaciones entre entidades para el diseño D003. Procesos por instancia.	100
Tabla 18. Modelo de datos lógico para el diseño D003. Procesos por instancia.	101
Tabla 19. Descripción de las entidades del modelo lógico para el diseño D004. Registro de colaboradores.	108
Tabla 20. Descripción de las relaciones entre entidades para el diseño D004. Registro de colaboradores.	109
Tabla 21. Modelo de datos lógico para el diseño D004. Registro de colaboradores.	110
Tabla 22. Descripción de las entidades del modelo lógico para el diseño D005. Propietarios por objeto.	118
Tabla 23. Descripción de las relaciones entre entidades para el diseño D005-Propietarios por objeto.	119
Tabla 24. Modelo de datos lógico para el diseño D005. Propietarios por objeto.	120
Tabla 25. Descripción de las entidades del modelo lógico para el diseño D006. Comentarios por propietario.	127
Tabla 26. Descripción de las relaciones entre entidades para el diseño D006. Comentarios por propietario.	128
Tabla 27. Modelo de datos lógico para el diseño D006. Comentarios por propietario.	129
Tabla 28. Descripción de las entidades del modelo lógico para el diseño D007. Conceptos por objeto.	135
Tabla 29. Descripción de las relaciones entre entidades para el diseño D007. Conceptos por objeto.	135
Tabla 30. Modelo de datos lógico para el diseño D007. Conceptos por objeto.	136
Tabla 31. Descripción de las entidades del modelo lógico para el diseño D008. Controles por objeto.	141

Tabla 32. Descripción de las relaciones entre entidades para el diseño D008. Controles por objeto.	141
Tabla 33. Modelo de datos lógico para el diseño D008. Controles por objeto.	142
Tabla 34. Descripción de las entidades del modelo lógico para el diseño D009. Servidores por ubicación geográfica.	145
Tabla 35. Descripción de las relaciones entre entidades para el diseño D009. Servidores por ubicación geográfica.	146
Tabla 36. Modelo de datos lógico para el diseño D009. Servidores por ubicación geográfica.	146
Tabla 37. Descripción de las entidades del modelo lógico para el diseño D010. Administradores por servidores.	150
Tabla 38. Descripción de las relaciones entre entidades para el diseño D010. Administradores por servidores.	151
Tabla 39. Modelo de datos lógico para el diseño D010. Administradores por servidores.	151
Tabla 40. Descripción de las entidades del modelo lógico para el diseño D011. Administradores de bases de datos por instancia.	155
Tabla 41. Descripción de las relaciones entre entidades para el diseño D011. Administradores de bases de datos por instancia.	155
Tabla 42. Modelo de datos lógico para el diseño D011. Administradores de bases de datos por instancia.	156
Tabla 43. Conceptualización del modelo multidimensional, dimensiones y hechos.	165
Tabla 44. Matriz de funcionalidad del modelo.	179
Tabla 45. Comparativa de herramientas para la administración de metadatos.	182

Índice de figuras

Figura 1: Estructura de descomposición del trabajo (elaboración propia del investigador)	36
Figura 2: Organigrama de Grupo Financiero BAC Credomatic Network.....	41
Figura 3: Organigrama de la Dirección Regional de Informática para Centro América.....	42
Figura 4: Países en los que opera BAC Credomatic.....	43
Figura 5 – Diccionario de datos en la arquitectura de bases de datos adaptado de Abraham <i>et al.</i> (Abraham, Henry y S. Sudarshan, 2002). Elaboración propia del investigador.....	52
Figura 6: Base de datos federada (elaboración propia del investigador).....	55
Figura 7: Base de datos distribuida (elaboración propia del investigador).	56
Figura 8. Esquema de estrella.....	60
Figura 9. Esquema copo de nieve.....	60
Figura 10. Representación de cubo multidimensional.	61
Figura 11. Ejemplo de herencia en el modelo relacional.	63
Figura 12. Ontología (elaboración propia del investigador).....	65
Figura 13. Fases de la metodología <i>rapid warehousing</i> (adaptación de <i>SAS Institute</i> . Elaboración propia del investigador).....	69
Figura 14. Diagrama de casos de uso (elaboración propia del investigador).....	73
Figura 15. Arquitectura de la solución. Diagrama de componentes.....	74
Figura 16. Modelo conceptual del diseño D001. Servidores, plataformas, gestores e instancias.	79

Figura 17. Modelo físico del diseño D001. Servidores, plataformas, gestores e instancias.	86
Figura 18. Modelo conceptual del diseño D002. Administración de bases de datos y esquemas por instancia.	89
Figura 19. Modelo físico del diseño D002. Administración de bases de datos y esquemas por instancia (elaboración propia del investigador).	96
Figura 20. Arquitectura de gestores de bases de datos (elaboración propia del investigador).	97
Figura 21. Modelo conceptual del diseño D003. Procesos por instancia (elaboración propia del investigador).	99
Figura 22.- Modelo físico del diseño D003. Procesos por instancia (elaboración propia del investigador).	105
Figura 23. Modelo conceptual del diseño D004. Registro de colaboradores.	108
Figura 24. Modelo físico del diseño D004. Registro de colaboradores (elaboración propia del investigador).	116
Figura 25. Modelo conceptual del diseño D005. Propietarios por objeto.	118
Figura 26. Modelo físico del diseño D005. Propietarios por objeto (elaboración propia del investigador).	124
Figura 27. Restricción sobre los objetos de los delegados.	126
Figura 28. Modelo conceptual del diseño D006. Comentarios por propietario.	127
Figura 29. Modelo físico del diseño D006. Comentarios por propietario (elaboración propia del investigador).	134
Figura 30. Modelo conceptual del diseño D007. Conceptos por objeto.....	135
Figura 31. Modelo físico del diseño D007. Conceptos por objeto (elaboración propia del investigador).	137
Figura 32. Relación de un concepto con múltiples objetos (elaboración propia del investigador).	138
Figura 33. Relación de múltiples conceptos con un objeto (elaboración propia del investigador).	138

Figura 34. Agrupación de conceptos en jerarquías (elaboración propia del investigador).	139
Figura 35. Relación de múltiples conceptos con un objeto (elaboración propia del investigador).	139
Figura 36. Modelo conceptual del diseño D008. Controles por objeto.	141
Figura 37. Modelo físico del diseño D008. Controles por objeto (elaboración propia del investigador).	144
Figura 38. Modelo conceptual del diseño D009. Servidores por ubicación geográfica. ..	145
Figura 39. Modelo físico del diseño D009. Servidores por ubicación geográfica (elaboración propia del investigador).	149
Figura 40. Modelo conceptual del diseño D010. Administradores por servidores.	150
Figura 41. Modelo físico del diseño D010. Administradores por servidores (elaboración propia del investigador).	153
Figura 42. Modelo conceptual del diseño D011. Administradores de bases de datos por instancia.	154
Figura 43. Modelo físico del diseño D011. Administradores de bases de datos por instancia (elaboración propia del investigador).	158
Figura 44. Servidores vinculados en <i>SQL Server</i>	159
Figura 45. Diagrama de actividad del proceso de extracción.	160
Figura 46. Extracto del modelo físico del diseño D002. Administración de bases de datos y esquemas por instancia.	162
Figura 47. Ejemplo de catálogos de bases de datos en <i>SQL Server</i>	163
Figura 48. Modelo de datos físico para las dimensiones regiones y países.	168
Figura 49. Modelo de datos físico para la dimensión conceptos.	169
Figura 50. Modelo de datos físico para la dimensión fechas.	170
Figura 51. Modelo de datos físico para la dimensión controles.	171
Figura 52. Modelo de datos físico para la dimensión comentarios.	172

Figura 53. Modelo de datos físico para las dimensiones propietarios titulares y propietarios delegados.	173
Figura 54. Modelo de datos físico para las dimensiones esquemas, bases de datos, instancias de bases de datos y servidores.	174
Figura 55. Modelo de datos físico para la dimensión conceptos.	175
Figura 56. Modelo de datos físico para la dimensión clases de objetos de base de datos.	175
Figura 57. Modelo de cubo multidimensional en <i>Visual Studio</i>	176
Figura 58. Visualización del cubo por medio de la herramienta <i>Tableau</i>	177
Figura 59. Fuentes de datos soportados por <i>Tableau</i>	178

1. Capítulo 1: Introducción

1.1. Antecedentes del problema

BAC Credomatic Network es una organización que se dedica a la prestación de servicios en el sector financiero, cuenta con operaciones en la región centroamericana, en países tales como Costa Rica, Panamá, Nicaragua, Honduras, Salvador y Guatemala, cada uno de los países posee sus propias unidades de negocio y de Tecnología de Información (TI), a estas unidades se les conoce como locales. La empresa cuenta con una unidad de TI regional, llamada DICA (Dirección de Informática para Centro América), la cual se encarga velar por el Gobierno de TI a nivel regional e involucra aspectos tales como administración de riesgos de TI y administración de recursos de TI, entre otros.

Como parte de la administración de riesgos de TI se encuentra el cumplimiento de las regulaciones. DICA debe velar porque los países cumplan con estas y a su vez con marcos de trabajo que tienen una relación directa con las áreas de TI, entre las cuales se puede mencionar:

- ✓ PCI, esta regulación expone un conjunto de normas de seguridad de datos que deben cumplir las empresas participantes en la industria de tarjetas de pago, esta norma es exigida por las marcas internacionales tales como VISA, *MasterCard* o *American Express* entre otras, de las cuales BAC Credomatic forma parte.
- ✓ SOX es una regulación exigida a las empresas que cotizan en la Bolsa de Valores, exige un subconjunto de controles internos relacionados con la confidencialidad, integridad y disponibilidad de datos, para asegurar exactitud de los informes financieros (sección 404).
- ✓ COBIT es un marco de trabajo, el cual especifica una serie de controles que pueden ser implementados en los departamentos de TI, entre los cuales se puede mencionar modelos de arquitectura de información empresarial, esquemas de clasificación de datos, diccionarios de datos empresariales, reglas

de sintaxis de datos y administración de la integridad de los datos. Actualmente, las entidades regulatorias de entidades financieras de varios países están exigiendo a BAC Credomatic, la implementación de dichos controles para poder operar en el mercado financiero.

- ✓ ITIL es un marco de trabajo, el cual especifica una serie de buenas prácticas enfocadas a la entrega de servicios de TI. Entre los temas que trata se encuentra administración de incidentes, administración de cambios, administración de niveles de servicio, administración de la capacidad, administración de la continuidad de los servicios de TI, entre otros; a BAC Credomatic se le exige la implementación de ITIL.
- ✓ La base de datos del *core* bancario que utiliza la compañía es DB2 para IBM i, el cual es utilizado para llevar acabo las transacciones financieras; sin embargo, las unidades de negocio, con el pasar del tiempo, han adquirido sistemas que se encuentran en otros gestores de bases de datos, tales como *Oracle*, *SQL Server* y DB2 para *Linux*, *Unix* y *Windows*. Esta diversificación de sistemas de bases de datos se debe, principalmente, a que en la elaboración de un proyecto de TI regional es necesario seguir un proceso de desarrollo, el cual no es lo suficientemente flexible para la entrega de los productos solicitados en el tiempo meta requerido por el negocio, y este termina optando por crear sus propios desarrollos a nivel local. En otros casos han adquirido *software* que tiene como requerimiento la implementación de otros motores de bases de datos diferentes a DB2 para IBM i.

La diversificación de los diferentes sistemas de bases de datos, en algunos casos, ha creado una especie de islas de información, estas se han convertido en algo difícil de administrar para DICA, ya que en ocasiones, DICA no conoce de su existencia, lo cual pone en riesgo de que no se identifique de manera oportuna si los datos contenidos en estas islas están incumpliendo con alguna regulación, lo que puede ocasionar que la organización incurra en sanciones económicas por parte de los entes reguladores. En las islas de información también se encuentra una serie de anomalías en los datos, tales como incompletitud, inconsistencia, estructuras de datos diferentes para almacenar la

misma información proveniente de otros sistemas y duplicidad, entre otros; adicionalmente, se encuentra que no cumplen con las normas de estandarización establecidas en la compañía.

Actualmente la empresa cuenta con una implementación de diccionario de datos, pero se encuentra limitada al flujo de desarrollo en DB2 para IBM i. Este componente se desea hacer disponible para las otras bases de datos que posee la organización, de manera tal que se pueda aprovechar los servicios que actualmente brinda.

La empresa no cuenta con una herramienta que le permita visualizar de manera integrada los objetos de bases de datos registrados en los distintos gestores, tales como tablas, columnas, vistas, disparadores, funciones, índices, secuencias, procedimientos almacenados, restricciones, esquemas y catálogos de bases de datos, que permitan a los usuarios, llevar a cabo clasificaciones de estos por medio del establecimiento de relaciones entre palabras clave definidas en un diccionario de datos empresarial y los objetos del repositorio integrado, con la finalidad de identificar cuáles objetos están relacionados con el cumplimiento de regulaciones, y así validar la aplicación de controles adecuados para evitar sanciones.

La organización cuenta con los lineamientos y metodologías para clasificar la sensibilidad, criticidad, propiedad de los datos, así como los controles de seguridad que le son aplicados; sin embargo, no existe un mecanismo automatizado, ni procesos que permitan su aplicación en motores de bases de datos diferentes a los sistemas IBM i de manera estandarizada.

1.2. Justificación

Es de gran importancia para la empresa BAC Credomatic contar con una herramienta que le permita a los usuarios gobernar de manera integrada los objetos que tienen en diferentes motores de bases de datos, para clasificarlos con base en la criticidad, sensibilidad, propiedad (quién es el dueño) y los controles de seguridad de manera estandarizada. Esto debido a que es un requerimiento solicitado por las entidades reguladoras impuestas a la organización; de no cumplirse se incurre en sanciones

económicas. Adicionalmente, es un objetivo del Gobierno de TI que la empresa se ha propuesto cumplir.

La clasificación de los datos permite responder a preguntas que son de gran interés para la organización, tales como: ¿quiénes son los propietarios de los datos?, ¿qué controles de seguridad se le aplican a un objeto en particular?, ¿en cuales entidades se encuentra la información considerada como crítica?, ¿cuáles son los objetos que están relacionados con alguna regulación en particular?

La importancia de identificar a los propietarios de los objetos radica en que estos son los usuarios en los cuales la organización delega la autoridad necesaria para la toma de decisiones con respecto a la información contenida en los sistemas. Un ejemplo práctico de la utilidad de esto es cuando se produce una alteración de los datos por error, y es necesario realizar cambios en los datos para solventar el problema, esto requiere un control de cambios, el cual debe ser aprobado por el propietario de los objetos, por lo que se requiere conocer quién es el propietario, para posteriormente solicitar el permiso correspondiente y proceder a realizar la modificación.

La importancia de identificar cuáles controles de seguridad se está aplicando a los objetos de la base de datos, radica en que permite identificar si estos controles son los apropiados o no. Un ejemplo práctico es cuando existe una tabla que contiene datos enmascarados a la hora de realizar las consultas; sin embargo, si alguna regulación solicita que el dato debe ser cifrado físicamente en la base de datos, se podría identificar que no se está cumpliendo con lo solicitado.

La importancia de identificar en cuáles entidades se encuentra la información considerada como crítica, radica en que permite identificar si a esas entidades se les está aplicando los controles de seguridad apropiados.

La importancia de identificar a cuáles objetos les aplica una regulación en particular, radica en que permite identificar rápidamente a cuáles objetos se les debe aplicar un cambio en los controles como respuesta a un cambio de alguna regulación.

Este proyecto disminuye los costos en la gestión y control de los datos, ya que no es necesario el estar desarrollando reportes en cada uno de los motores que posee la organización para responder a las preguntas descritas anteriormente.

La empresa está viendo la necesidad de gobernar sus objetos contenidos en los gestores de bases de datos heterogéneos, ya que cada vez se está volviendo más compleja y difícil de administrar. Solo el año pasado adquirieron dos bancos nuevos, uno en Panamá y otro en Guatemala, como parte del proceso de fusión fue necesario integrar las bases de datos; además, el negocio se encuentra en constante cambio y sus datos también.

1.3. Viabilidad

1.3.1. Factibilidad técnica

La empresa cuenta con un servidor que tiene instalada una base de datos *SQL Server* disponible con su licencia al día; adicionalmente, posee los conectores ODBC necesarios para configurar la conectividad entre las diferentes bases de datos con las que cuenta la organización (*DB2, Oracle y SQL Server*). Por medio de la creación de servidores vinculados se cuenta con el acceso necesario a los servicios de red para acceder a todos los servidores de producción; adicionalmente, la empresa cuenta con la infraestructura, *software* y *hardware* necesario para construir el modelo propuesto, por lo que, desde el punto de vista técnico, es factible su construcción.

1.3.2. Factibilidad económica

Los costos del desarrollo del modelo propuesto se estiman tomando en cuenta las etapas de un proyecto de desarrollo de *software*, que contempla análisis, diseño, desarrollo, pruebas y documentación. Cada una de esas etapas cuenta con un grupo de tareas, estas detallan el costo en horas hombre requeridas para su elaboración. Con base en la información expuesta por el Ministerio de Trabajo y Seguridad Social de Costa Rica (2014) y basándose en el salario de un licenciado universitario, de ₡609.355,75 colones, este se dolariza a un tipo de cambio de referencia de ₡538,00 colones por dólar, lo que da como resultado un salario mensual de \$1.132,63. Se

estima que un empleado labora 192 horas al mes, por lo que su costo por hora es de \$5,90 (\$1.132,63/192 horas), como se muestra en la **Tabla 1**.

Tabla 1. Costos estimados (elaboración propia del investigador)				
Costos Estimados				
ETAPAS	TAREAS DEL PROYECTO	HORAS	COSTO	TOTAL TAREA
ANALIZAR	Definición de requerimientos	32	\$5,9	\$188,8
	Analizar herramientas del mercado	30	\$5,9	\$177,0
	Subtotal	62	\$11,80	\$365,80
DISEÑAR	Diseñar de la arquitectura	6	\$5,9	\$35,4
	Diseñar el modelo de datos	32	\$5,9	\$188,8
	Diseñar los procesos para la integración de datos	22	\$5,9	\$129,8
	Diseñar los proceso de extracción	22	\$5,9	\$129,8
	Diseñar los proceso de transformación	22	\$5,9	\$129,8
	Diseñar los proceso de carga	22	\$5,9	\$129,8
	Diseñar cómo asociar conceptos y objetos de las bases de datos	22	\$5,9	\$129,8
	Diseñar cómo asociar propietarios y objetos de las bases de datos	22	\$5,9	\$129,8
	Diseñar cómo asociar controles de seguridad y objetos de las bases de datos	22	\$5,9	\$129,8

	Diseñar cómo asociar observaciones de los propietarios y objetos de las bases de datos	22	\$5,9	\$129,8
	Subtotal	214	\$59,00	\$1.262,60
DESARROLLAR	Desarrollar el modelo de datos	3	\$5,9	\$17,7
	Desarrollar los procesos para la integración de datos	12	\$5,9	\$70,8
	Desarrollar los procesos de extracción	12	\$5,9	\$70,8
	Desarrollar los procesos de transformación	12	\$5,9	\$70,8
	Desarrollar los procesos de carga	4	\$5,9	\$23,6
	Subtotal	43	\$29,50	\$253,70
PROBAR	Crear un plan de pruebas	16	\$5,9	\$94,4
	Ejecutar el plan de pruebas	6	\$5,9	\$35,4
	Evaluar la funcionalidad del modelo con base en los resultados de las pruebas	3	\$5,9	\$17,7
	Subtotal	25	\$17,70	\$147,50
DOCUMENTAR	Documentar el modelo	16	\$5,9	\$94,4
	Subtotal	16	\$5,90	\$94,40
Total		360,0		\$2.124

Fuente: elaboración propia del investigador.

En la Tabla 2 se presenta un flujo de efectivo cuyo objetivo es determinar el período de tiempo que se tardará aproximadamente en recuperar la inversión inicial, en caso que la empresa decida construir la solución propuesta.

Los ingresos se calculan con base en los ahorros que la compañía obtendría de implementar la solución, en un período de tiempo de seis meses, tomando en cuenta las horas hombre requeridas en realizar el análisis diario de las bases de datos, para determinar si existen objetos que no cumplen con alguna medida regulatoria. La estimación se realizó de la siguiente manera: un empleado tiene un costo de \$5,9 la hora, le toma en promedio 2,5 horas en realizar la revisión, llenar las bitácoras, lo cual es realizado a diario por 30 días, esto suma un costo de \$442,5 ($\$5,9 * 2,5 \text{ horas} * 30 \text{ días}$), las irregularidades se pueden presentar por la instalación de aplicaciones, lo cual forma parte de la operativa de la compañía, las revisiones son realizadas tanto en el servidor de producción como en el de contingencia.

Para calcular el costos de implementar la solución propuesta en la organización, se utiliza la metodología propuesta por Sánchez y Tarapuez (Sánchez Zanbrano & Tarapuez Roa, 2010), los cuales hacen uso del concepto de costo total de propiedad, conocido por sus siglas como TCO (*total cost of ownership*), empleando la siguiente fórmula: $\text{TCO} = \text{costo de aprendizaje} + \text{costo de licenciamiento} + \text{costos de hardware} + \text{costos de mantenimiento y soporte}$.

Para medir el costo en *hardware* se ejecuta un proceso de ingeniería inversa, con la finalidad de calcular el tamaño en megabytes sobre los metadatos que describen a los objetos contenidos en el esquema más pesado del *core* bancario. Se encontró que el más pesado es de 12,3 MB, actualmente la base de datos IBM i contiene en promedio 46 esquemas, por lo que se estima que por cada base de datos que se incorpore al sistema se requiere 565,8MB ($12,3 \text{ MB de metadatos} * 46 \text{ esquemas}$). Se estima que en la organización existen 12 gestores de bases de datos que deben ser supervisados, por lo que se requiere de espacio 6825,6MB ($565,8 \text{ MB} * 12$) como mínimo. Adicionalmente, se contempla que la solución requiere 6825,6 MB para el área de pruebas y esa misma cantidad para el almacén de datos y además, respetar la política interna establecida en la compañía de mantener un 150% de

espacio adicional en disco, a partir del tamaño base definido en los gestores de bases de datos, por lo que se estima que la solución utilizará un total de 34128 MB $((6825,6 \text{ MB} * 2) * 150\% + 6825,6 \text{ MB} * 2)$). Para calcular el costo del espacio requerido se utilizó como base el precio de un disco de 1TB con un costo de \$380¹. Un terabyte tiene 1048576 megabytes, por lo que el costo por megabyte es de \$0,00037 $(\$380/1048576 \text{ MB})$, se estima un costo total de almacenamiento de \$12,37 $(34128 \text{ MB} * \$0,00037)$. El gasto es mensual, ya que es necesario comprar almacenamiento para realizar respaldos.

El costo de aprendizaje contempla la elaboración de una publicación en una *wiki*, que actualmente utiliza la compañía, con la finalidad de capacitar a los usuarios, esta tiene un costo de \$17,7, calculada de la siguiente manera $\$5,9 * 3 \text{ horas} = \$17,7$ (salario * horas requeridas).

La solución no tiene costos de licenciamiento, ya que la compañía posee licencias que puede utilizar sin necesidad de adquirir otras adicionales.

Los costos referentes al mantenimiento y soporte se estiman tomando en cuenta el tiempo que un colaborador emplea en realizar las revisiones periódicas al sistema, se calcula de la siguiente manera $\$5,9 * 8 \text{ horas}$ (salario * horas requeridas), para un total de \$47 por mes.

Tabla 2. Flujos de efectivo proyectados (elaboración propia del investigador)

Flujo de efectivo proyectado a 6 meses						
Mes	1	2	3	4	5	6
Inversión inicial	<u>-\$2.124</u>	<u>-\$1.759</u>	<u>-\$1.375</u>	<u>-\$992</u>	<u>-\$609</u>	<u>-\$226</u>
Ganancias						
Ahorros de análisis por computador	\$443	\$443	\$443	\$443	\$443	\$443

¹ 1. Según datos obtenidos de Newegg (Newegg, 2015) y Amazon (Amazon, 2015).

Total de ingresos	<u>\$443</u>	<u>\$443</u>	<u>\$443</u>	<u>\$443</u>	<u>\$443</u>	<u>\$443</u>
Gastos						
Aprendizaje						
Creación de documentación en <i>Wiki</i>	\$18					
Licenciamiento						
Compra de <i>SQL Server</i>	\$0					
Hardware						
Compra de espacio en disco	\$12	\$12	\$12	\$12	\$12	\$12
Mantenimiento y soporte						
Revisión periódica del sistema	\$47	\$47	\$47	\$47	\$47	\$47
Total de gastos	<u>\$77</u>	<u>\$59</u>	<u>\$59</u>	<u>\$59</u>	<u>\$59</u>	<u>\$59</u>
Flujo de efectivo	-\$1.759	-\$1.375	-\$992	-\$609	-\$226	\$157

Según el flujo de efectivo proyectado a partir del sexto mes, se recupera la inversión inicial y se comienza a percibir flujos de efectivo positivos (beneficios económicos), por lo cual se determina que el proyecto es económicamente factible.

1.3.3. Factibilidad operacional

Se requiere capacitar al recurso humano para la utilización de la solución propuesta de forma adecuada, ya que actualmente la compañía no cuenta con un sistema similar. Actualmente se cuenta con un *wiki* que es accesible a todos los miembros de la organización, donde se pueden publicar los manuales necesarios para

que los usuarios puedan capacitarse sobre el funcionamiento del nuevo sistema, por lo que se determina que la solución es operacionalmente factible.

1.4. Planteamiento del problema

1.4.1. Problema general

La empresa desea contar con una amplia gobernabilidad sobre los objetos de bases de datos existentes en toda la organización, de manera tal que le permita clasificar a estos con base en sensibilidad o riesgo, identificar los propietarios, los controles de seguridad y a su vez, determinar en cuáles plataformas y gestores de bases de datos se encuentran.

1.4.2. Subproblemas del proyecto

- Se encuentran bases de datos heterogéneos, con distintos objetivos, diseños y dueños, pero con datos en común en la mayoría de los casos, los cuales se desean gobernar de manera conjunta.
- No existen procesos que permitan integrar los objetos de bases de datos residentes en diferentes plataformas y gestores.
- No se cuenta con una herramienta que permita visualizar información sobre los objetos de bases de datos residentes en diferentes plataformas y gestores, de forma integrada, la cual implemente algún mecanismo de autenticación.

1.5. Objetivos del proyecto

Se utiliza la taxonomía Bloom expuesta por Engelhart *et al.* (Engelhart, Hill, Furst, Krathwohl, & Bloom, 1956) para la definición de los objetivos.

1.5.1. Objetivo general

Diseñar una solución que permita la gobernabilidad sobre los objetos de bases de datos, la cual facilite clasificar a estos con base en sensibilidad o riesgo, identificar sus propietarios, los controles de seguridad implementados y, a su vez, determinar en cuáles plataformas y gestores se encuentran.

1.5.2. Objetivos específicos

- Construir un modelo de datos que permita identificar, de manera centralizada, los objetos de bases de datos existentes en toda la organización, para gobernarse de manera conjunta.
- Crear procesos que permitan integrar los objetos de bases de datos residentes en diferentes plataformas y gestores.
- Proponer una herramienta que permita visualizar información sobre los objetos de bases de datos, residentes en diferentes plataformas y gestores, de forma integrada, la cual implemente algún mecanismo de autenticación.

1.6. Propuesta de solución

Se propone crear un modelo de datos que almacene un catálogo de objetos de bases de datos integrado, de manera tal que permita identificar la ruta en la cual residen y generar un identificador único numérico para cada uno de los objetos, el cual sea independiente del gestor de base de datos y la plataforma a la cual pertenecen.

Se propone que la ruta esté compuesta por los siguientes metadatos: nombre del servidor + nombre de la instancia de base de datos + base de datos + esquema + tipo de objeto + nombre del objeto, de esta manera se puede identificar en plataformas y gestores heterogéneos.

Los objetos que se desea integrar son tablas, columnas, catálogos de bases de datos, disparadores, funciones, índices, procedimientos almacenados, secuencias y vistas, para que los usuarios puedan llevar a cabo clasificaciones de estos.

El identificador único permite establecer relaciones entre el catálogo de objetos integrado y cualquier otra entidad existente en el modelo relacional que haga referencia a este, permitiendo ampliar los atributos que describen a los objetos de bases de datos.

Se demuestra cómo extender los atributos de los objetos de bases de datos para describir quiénes son los propietarios de estos, cuáles controles de seguridad tienen implementados y cuáles son los comentarios de sus propietarios.

Se propone la utilización de un diccionario de datos, ya que permite registrar palabras que tienen asignadas un identificador único, el cual será utilizado para crear asociaciones entre los objetos de bases de datos y los conceptos de este. Lo comentado anteriormente es útil especialmente para relacionar a los objetos del catálogo integrado con niveles de sensibilidad o riesgo. Adicionalmente, se propone establecer una relación recursiva para construir jerarquías de conceptos en el diccionario, por ejemplo el concepto seguridad puede ser padre del concepto cifrado, de igual manera para cualquier otro concepto.

Para integrar los metadatos que describen a los objetos contenidos en cada uno de los gestores, se propone la utilización de las tecnologías de bases de datos federadas, ya que estas facilitan interconectar los diferentes sistemas de bases de datos.

Se propone la utilización de tecnologías de almacenes de datos para generar un *datamart*, enfocado en el tema específico del diccionario de datos empresarial, el cual puede ser accedido por los usuarios que posean los privilegios necesarios en la organización.

El modelo propuesto permite definir acciones, las cuales invocan a aplicaciones específicas en la base de datos. A su vez, las acciones son utilizadas para definir procesos, los cuales reutilizarán las acciones para ser ejecutadas con una prioridad; de esta manera, es posible la reutilización de las acciones y los procesos para cada una de las instancias de base de datos, de forma tal que el sistema pueda crecer de forma modular, sin necesidad de realizar grandes cambios a nivel de procesos. Esta

característica se incorpora para agregar nuevos gestores de bases de datos y configurar los procesos de extracción, transformación y carga, de manera flexible y con el menor impacto a nivel de configuración.

Para la elaboración de este proyecto se propone utilizar un proceso de desarrollo de *software*, ya que este tiene una gran similitud con el método científico, en el cual se plantea un problema a resolver por medio del establecimiento de los objetivos en la recolección de requerimientos, se plantea una hipótesis de cómo resolver el problema por medio del diseño de la solución, se realiza experimentación por medio de la ejecución de un plan de pruebas y finalmente, se obtienen conclusiones por medio de la revisión de la ejecución de las pruebas con base en la funcionalidad esperada, en la cual se determina si el diseño de la solución propuesta cumple con la funcionalidad requerida para resolver el problema planteado.

1.7. Cronograma de desarrollo de la propuesta

El cronograma propuesto se subdivide en las etapas de un proyecto de desarrollo de *software* que contempla análisis, diseño, desarrollo (construcción), pruebas y documentación.

La etapa de análisis contempla la definición de los requerimientos y análisis de herramientas que ofrece el mercado, con base en las características que estas proveen.

La etapa de diseño contempla un conjunto de propuestas, tales como definición de la arquitectura, definición del modelo de datos a nivel lógico y físico, definición de los procesos para la integración, extracción, transformación y carga de los datos, el cómo relacionar los objetos con conceptos del diccionario empresarial, el cómo relacionar los objetos con propietarios, el cómo relacionar los objetos con controles de seguridad y como relacionar los objetos con las observaciones de sus propietarios.

La etapa de desarrollo contempla construcción de los procesos para la integración, extracción, carga y transformación de los datos, así como el modelo de datos definido en la etapa de diseño.

La etapa de pruebas contempla de la creación, ejecución y evaluación de un plan de pruebas.

La etapa de documentación contempla la integración de los documentos del diseño con la finalidad de que este pueda ser analizado en posteriores revisiones del sistema. En la figura 1 se expone la estructura de descomposición del trabajo del presente proyecto.

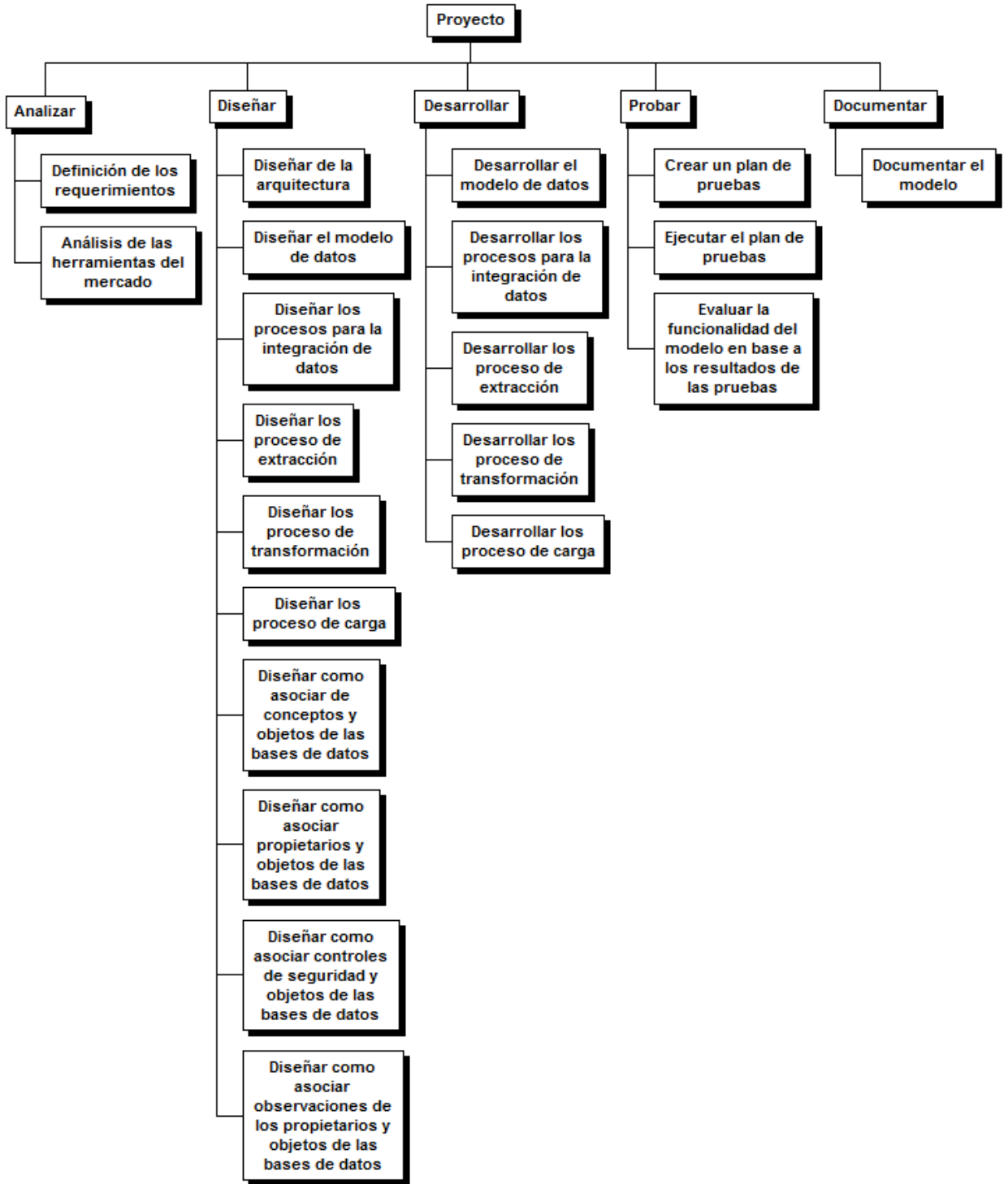


Figura 1: Estructura de descomposición del trabajo (elaboración propia del investigador)

Las fechas expuestas en el cronograma utilizan el formato día/mes/año como se presenta en la Tabla 3.

Tabla 3. Cronograma del proyecto.

Cronograma			
ETAPAS	TAREAS DEL PROYECTO	FECHA DE INICIO	FECHA DE FINALIZACIÓN
ANALIZAR	Definición de requerimientos	01/01/2015	19/01/2015
	Analizar herramientas del mercado	20/01/2015	07/02/2015
DISEÑAR	Diseñar de la arquitectura	08/02/2015	09/02/2015
	Diseñar el modelo de datos	10/02/2015	21/02/2015
	Diseñar los procesos para la integración de datos	22/02/2015	01/03/2015
	Diseñar los proceso de extracción	02/03/2015	09/03/2015
	Diseñar los proceso de transformación	10/03/2015	17/03/2015
	Diseñar los proceso de carga	18/03/2015	25/03/2015
	Diseñar cómo asociar conceptos y objetos de las bases de datos	26/03/2015	02/04/2015
	Diseñar cómo asociar propietarios y objetos de las bases de datos	03/04/2015	10/04/2015
	Diseñar cómo asociar controles de seguridad y objetos de las bases de datos	11/04/2015	18/04/2015
	Diseñar cómo asociar observaciones de los propietarios y objetos de las bases de datos	19/04/2015	26/04/2015

DESARROLLAR	Desarrollar el modelo de datos	27/04/2015	28/04/2015
	Desarrollar los procesos para la integración de datos	29/04/2015	06/05/2015
	Desarrollar los proceso de extracción	07/05/2015	14/05/2015
	Desarrollar los proceso de transformación	15/05/2015	22/05/2015
	Desarrollar los proceso de carga	23/05/2015	24/05/2015
PROBAR	Crear un plan de pruebas	25/05/2015	27/05/2015
	Ejecutar el plan de pruebas	28/05/2015	28/05/2015
	Evaluar la funcionalidad del modelo con base en los resultados de las pruebas	29/05/2015	30/05/2015
DOCUMENTAR	Documentar el modelo	31/05/2015	01/07/2015

1.8. Marco de referencia organizacional y socioeconómico

1.8.1. Reseña histórica

Según información recopilada (Credomatic, 2014), se indica que los inicios del Grupo BAC Credomatic se remontan a más de medio siglo atrás, cuando en 1952 se fundó el Banco de América en Nicaragua. Sin embargo, fue hasta los años setenta cuando se incursionó en el negocio de tarjetas de crédito a través de las empresas Credomatic.

A mediados de los años ochenta, el grupo decidió ingresar en otros mercados de la región, empezando por Costa Rica, con la adquisición de lo que hoy se conoce como Banco BAC San José. Fue en la década de 1990 que se concretó la expansión hacia los otros mercados centroamericanos, fortaleciendo así la presencia del grupo en toda la región, la cual se mantiene hasta el día de hoy.

Ya en el año 2004, el grupo inició sus operaciones de tarjeta de crédito en México y, un año más, tarde se llevó a cabo una alianza estratégica por medio de la cual *GE Consumer Finance* (subsidiaria de *GE Capital Corporation*) adquirió el 49,99% del capital de BAC Credomatic, una sociedad que controlaba indirectamente el 100% de BAC *International Bank*.

Paralelamente y como parte de la estrategia de expansión, se llevó a cabo la adquisición del Banco Mercantil (BAMER) de Honduras, uno de los bancos privados más importantes de ese país, para dar paso a lo que hoy se conoce como BAC Honduras. En el 2007 también se adquirieron Propemi (Programa de Promoción a la Pequeña y Microempresa) en El Salvador y la Corporación Financiera Miravalles en Costa Rica, ambas compañías dirigidas a segmentos específicos de mercado.

A mediados del 2009, la compañía *GE Capital Corporation* aumentó su participación accionaria al 75%, convirtiéndose así en el accionista mayoritario. No obstante, a raíz de un cambio de estrategia a nivel mundial, GE decidió concentrarse más en la actividad industrial (infraestructura, tecnología y salud) y menos en actividades de banca privada y comercial.

Como resultado, en julio del 2010, el Grupo Aval de Colombia, el conglomerado financiero más grande ese país -conformado por el Banco de Bogotá, el Banco de Occidente, el Banco AV Villas, el Banco Popular y el Fondo de Pensiones AP Porvenir-, suscribió un contrato de compraventa de acciones con *GE Consumer Finance* relativo a la adquisición del 100% de las acciones del Grupo BAC Credomatic. En diciembre del 2010, y después de obtener las aprobaciones de las superintendencias de entidades financieras de cada país, el proceso de compra culminó exitosamente.

Cabe resaltar que, a pesar del cambio de control accionario, la estrategia de negocios y la identidad del Grupo BAC Credomatic se mantienen y, más bien, a raíz de la adquisición ha sido posible ofrecer productos de mayor valor agregado a los clientes, compartir experiencias, aprovechar las sinergias y las mejores prácticas de ambas partes y, sobre todo, compartir la visión de negocios, lo que hace que BAC Credomatic siga siendo hoy en día una organización caracterizada por el mejoramiento continuo, la pasión por la excelencia, la innovación y la creatividad.

Como expone BAC Credomatic Network, su cronología histórica se puede resumir en los siguientes eventos:

- 1952: se fundó el Banco de América en Nicaragua.
- Década de 1970: se iniciaron las operaciones de tarjeta de crédito mediante las empresas Credomatic.
- Década de 1980: el grupo incursionó en el negocio de banca en Costa Rica.
- Década de 1990: se obtuvo las licencias bancarias en el resto de los países de América Central y se fortaleció aún más la presencia del grupo en toda la región.
- 2004: inicio de operaciones de tarjeta de crédito en México.
- 2005: alianza estratégica con *GE Consumer Finance* (adquisición del 49,99% de las acciones)
- 2007: adquisición de BAMER (Honduras), Propemi (El Salvador) y la Corporación Financiera Miravalles (Costa Rica).
- 2009: *GE Capital* aumenta su participación de capital al 75%.
- 2010: Grupo Aval de Colombia adquiere el 100% de las acciones del grupo.

1.8.2. Misión

BAC (BAC Credomatic, 2013) indica que su misión es facilitar con excelencia el intercambio y financiamiento de bienes y servicios, a través de sistemas de pago y soluciones financieras innovadoras y rentables que contribuyan a generar riqueza, a crear empleo y a promover el crecimiento económico sostenible y solidario de los mercados donde operan.

1.8.3. Visión

Su visión es ser la organización financiera preferida de las comunidades que se atienden por su liderazgo en medios de pago, por su solidez, confiabilidad, avanzada tecnología y conectividad con personas y empresas, a quienes les simplifican la vida facilitándoles la realización de sus sueños y el logro de sus metas. (BAC Credomatic, 2013)

1.8.4. Organigrama

La estructura organizacional es una adaptación realizada a partir del diseño original con el cual cuenta la compañía, ya que esta es considerada confidencial. En la figura 2 se muestra el organigrama compuesto por cada una de las empresas que conforman al Grupo Financiero, la figura 3 muestra el organigrama de la Dirección Regional de Informática, en la figura 4 se muestran los países en los cuales opera el grupo financiero, donde la Dirección Regional de Informática tiene que velar por la alineación de la estrategia de TI.

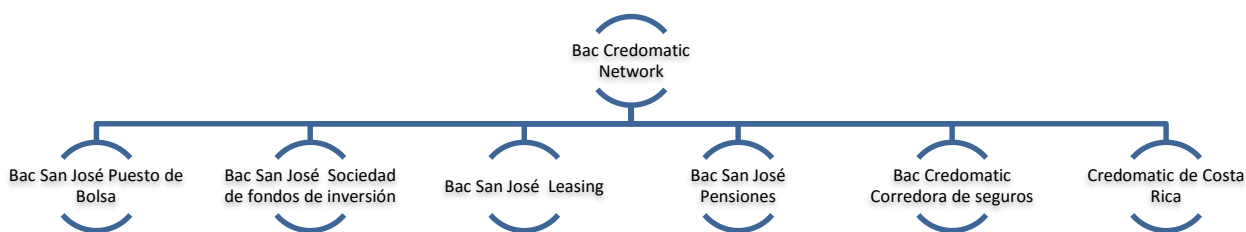


Figura 2: Organigrama de Grupo Financiero BAC Credomatic Network

Fuente: BAC Credomatic, adaptación propia del investigador (BAC Credomatic, 2013b).

BAC San José Puesto de Bolsa: se encarga de la compra y venta de títulos valores, tanto en el mercado primario como secundario.

BAC San José Sociedad de Fondos de Inversión: se encarga de la administración del capital de los inversionistas, sean naturales o jurídicos, que forman parte del fondo de inversión.

BAC San José *Leasing*: se encarga del arrendamiento operativo y financiero de sus clientes.

BAC San José Pensiones: se encarga de la administración del fondo de pensiones, tanto de pensiones obligatorias como voluntarias.

BAC Credomatic Corredora de Seguros: se encarga de la emisión de seguros, tanto personales como de daños.

Credomatic de Costa Rica: se encarga del negocio de tarjetas, tanto emisor como adquirente.

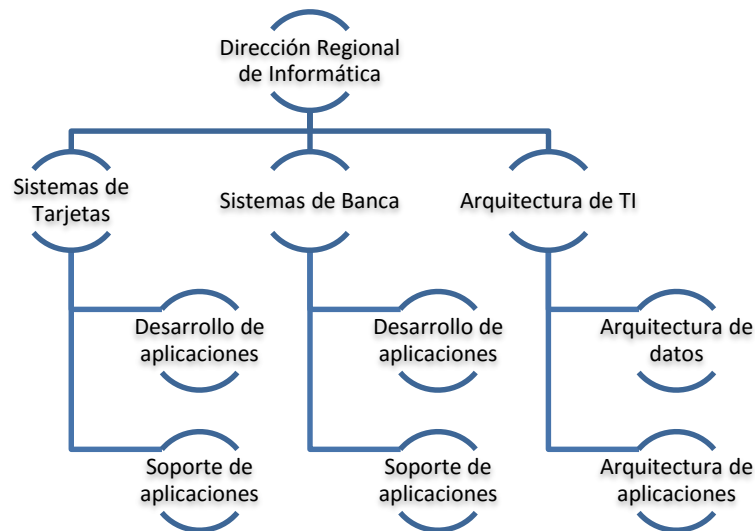


Figura 3: Organigrama de la Dirección Regional de Informática para Centro América.

Fuente: BAC Credomatic, adaptación del investigador a partir de documentación interna (BAC Credomatic, 2013)

Sistemas Tarjetas: se encarga del desarrollo y mantenimiento de las soluciones de *software* orientado al mercado de tarjetas. Este se subdivide en Desarrollo y Soporte.

Sistemas Banca: se encarga del desarrollo y mantenimiento de las soluciones de *software* orientadas al mercado bancario. Este se subdivide en Desarrollo y Soporte.

Arquitectura de TI: ofrece un conjunto de servicios que favorecen el uso de mejores prácticas en el diseño, construcción e implementación, tanto de la base de datos como de componentes y servicios de *software*.

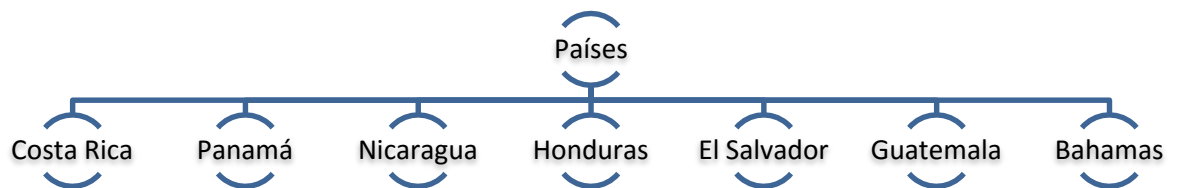


Figura 4: Países en los que opera BAC Credomatic.

Fuente: BAC Credomatic, elaboración propia del investigador (BAC Credomatic, 2013b).

1.8.5. Valores de la organización

Los valores de BAC Credomatic son los siguientes (BAC Credomatic, 2013):

- **Excelencia:** distinguirse por su alto grado de responsabilidad, atendiendo a su propia formación con el fin de hacer un trabajo de calidad superior que lo haga digno de aprecio y estimación en la realización de sus funciones.
- **Integridad:** actuar con rectitud, honestidad, honradez y transparencia, de manera congruente, sin engaños, ni falsedades en la realización de sus funciones.
- **Innovación:** es la búsqueda constante de cambios que generen un valor económico y social.
- **Responsabilidad:** cumplir con oportunidad, eficiencia y eficacia las tareas inherentes a sus funciones, las cuales serán atendidas con diligencia y conocimiento.
- **Respeto:** consideración y tolerancia a las diferencias entre las personas.

1.9. Alcances y limitaciones del proyecto

1.9.1. Alcance

El presente proyecto explorará el uso de los metadatos que describen a los objetos contenidos en las bases de datos en sus diccionarios de datos del sistema, con la finalidad de generar un modelo de datos que permita gobernar los objetos de bases de datos existentes en plataformas y gestores heterogéneos de manera conjunta.

1.9.2. Limitaciones

- A. La investigación genera un diseño y un prototipo, más no su implementación e instalación en la empresa.
- B. La construcción del prototipo se llevará a cabo en el gestor de base de datos de *SQL Server*.
- C. La investigación abarca únicamente los gestores de bases de datos *DB2*, *Oracle* y *SQL Server*, debido a que estos son los utilizados por la organización en la cual se lleva a cabo la investigación.
- D. No se revelará información que la empresa considera confidencial.
- E. No es posible obtener demostraciones de *software* por parte de los fabricantes de las herramientas de administración de metadatos.

1.10. Estado de la cuestión

El estado de la cuestión se desarrolló según lo expuesto por Biolchini *et al.* (Biolchini, Gomes Mian, Cruz Natali, & Horta Travassos, 2005).

1.10.1. Planificación

1.10.1.1. Formulación de la pregunta

Objetivo

Identificar estudios previos, herramientas y modelos de datos que permitan gobernar los objetos contenidos en gestores de bases de datos y plataformas heterogéneas, de manera conjunta.

Problema

Se requiere identificar procesos, herramientas y modelos de datos que permitan gobernar los objetos de bases de datos contenidos en gestores heterogéneos, de manera tal que permita realizar clasificaciones de estos con base en sensibilidad, controles de seguridad implementados y a su vez que permita establecer sus propietarios.

Pregunta

¿Existen estudios previos, herramientas y modelos de datos que permitan gobernar los objetos contenidos en bases de datos heterogéneas?

Palabras clave

Para realizar las búsquedas de los documentos relacionados con el objetivo de investigación, se utilizaron las siguientes palabras: *data dictionary, metadata management, data warehouse, federated database*.

Intervención

Serán analizados estudios previos para identificar procesos, herramientas y modelos de datos que permitan gobernar los objetos

contenidos en gestores de bases de datos de diferente tipo, incluyendo a DB2, *Oracle* y *SQL Server*.

Población

Las poblaciones de documentos a ser analizados corresponden a las publicaciones que hacen referencia a la gobernabilidad de sistemas de información, haciendo énfasis en la administración de los metadatos, los cuales tengan relación con los gestores de bases de datos DB2, *Oracle* y *SQL Server*.

Aplicación

Las áreas de aplicación que se pueden ver beneficiadas de la investigación son las relacionadas con la gobernabilidad de los sistemas de información, ya que permiten:

- La gobernabilidad sobre los objetos de bases de datos, la cual facilite clasificar a estos con base en sensibilidad o riesgo, identificar sus propietarios, los controles de seguridad implementados y, a su vez, determinar en cuáles plataformas y gestores se encuentran.

1.10.1.2. Selección de recursos

Identificación de fuentes

Definición de los criterios de selección

- **Fuentes de métodos de búsqueda:** las búsquedas se realizan por medio de consultas a motores de búsqueda Web y motores de búsqueda de documentos electrónicos, tales como *Adobe PDF*,

Microsoft Word, consultas a expertos y profesores sobre temas específicos.

- **Cadenas de búsqueda:** se utilizarán las siguientes palabras claves:
data dictionary AND metadata managment AND dataware house AND federated database.
- **Lista de recursos:**
 - *Forrester Research*
 - *Data Governace Institute*
 - *IBM Developerworks*
 - *IT Governace Institute*
 - *Informatica - The Data Integration Company.*
 - *ISACA*
 - *MDM Institute*
 - *Microsoft Developer Network*
 - *National Information Standards Organization*
 - *Oracle Technology Network*
 - *PCI Security Standards Council*
 - *SAS Institute*
 - *Technology Research | Gartner Inc.*
 - *U.S Securities and Exchange Commisions*
 - Libros
 - Artículos
 - Documentación suministrada por los colaboradores de la empresa.
 - Acceso a la aplicación que se encuentra instalada en el servidor IBM i para su análisis.
 - Acceso a información vía Web.
 - Experiencia de profesores y especialistas en el tema.

Verificación de referencias

Varios expertos en el tema verificaron las palabras clave especificadas en la sección cadenas de búsqueda con el fin de determinar que son las más apropiadas para obtener buenos resultados, así como los documentos analizados.

1.10.1.3. Selección de estudios

Procedimiento para la selección de estudios

Se utilizaron los siguientes criterios para incluir/excluir estudios:

- Incluir/excluir estudios por título.
- Incluir/excluir estudios por palabras clave.
- Incluir/excluir estudios en función de los contenidos de las conclusiones.
- Incluir/excluir estudios en función de los resúmenes.

1.10.2. Ejecución

Se ejecutan las búsquedas por medio de las palabras claves establecidas en las cadenas de búsqueda.

Ejecución de la selección

Selección de los resultados iniciales

Se realiza una primera iteración para verificar los resultados obtenidos utilizando las cadenas de búsqueda, se concluye que se encuentra documentación necesaria para ser evaluada.

Evaluar la calidad de los estudios

Se valida la credibilidad, relevancia y rigor de los documentos obtenidos en los resultados de búsqueda, se verifica que las fechas de publicación sean recientes y se comprueba que los temas expuestos concuerden con varios autores sobre el mismo tema (validación cruzada).

Extraer los datos de los estudios primarios

Se extraen los documentos considerados más relevantes para el tema en investigación y se incorporan en las referencias de este trabajo.

1.10.3. Análisis de resultados

Abraham *et al.* (Abraham, Henry y S. Sudarshan, 2002) indican que los diccionarios de datos forman parte de los sistemas de base de datos, a partir de ese concepto se procede a analizar la documentación oficial de los proveedores DB2, Oracle, SQL Server para entender qué tipo de información es almacenada en estos diccionarios, y cómo estos pueden ser utilizados para gobernar los objetos contenidos en las bases de datos. Se concluye que se almacena información muy similar, pero de manera no estandarizada, ya que cada motor tiene su propia forma de definir las estructuras de datos que componen a los diccionarios, dicha información se puede integrar en un repositorio de forma estandarizada.

ANSI (ANSI/ISO/IEC, 1999) expone un estándar de cómo organizar los diccionarios de datos en las bases de datos; sin embargo, por medio del análisis de los diccionarios se determina que estos no implementan al pie de la letra el estándar, sino que tienen variaciones que les permite cumplir con el estándar, incluso se llegó a descubrir que los fabricantes tienen varias vistas de los diccionarios para cumplir con diferentes estándares de acceso a bases de datos, como ODBC y JDBC.

Gartner Inc. (Gartner Inc., 2014) expone una comparativa de los principales proveedores de herramientas para la integración de los datos, se

procede a analizar la documentación y se encuentra que los motores de bases *SQL Server*, *Oracle* y *DB2* tienen incorporadas tecnologías de bases de datos federadas, lo cual facilita la comunicación entre bases de datos de diferente tipo para generar un vista unificada de los diferentes diccionarios que poseen diferentes motores de bases de datos.

En una investigación realizada por *Forrester Research* en la cual Peyret y Michele (Peyret y Michele, 2014) exponen acerca de las principales herramientas que facilitan la gobernabilidad de los datos que se encuentran en la industria del *software*, se realiza una revisión de los productos que ofrecen los principales proveedores del mercado, tales como IBM, Informática USA, *Adaptive* y se determina que existen herramientas especializadas para la administración de los metadatos (a diferencia de las herramientas de administración de datos maestros), entre los que se puede mencionar *Informatica Metadata Manager and Business Glossary*, *IBM InfoSphere Information Server* y *Adaptive Metadata Manager*; sin embargo, no se encuentra información concluyente sobre los modelos de datos físicos o lógicos que utilizan esas herramientas para la administración de los metadatos, por lo que se propone el diseño de una solución que exponga un modelo de datos que aporte a la industria una forma de administrar los metadatos.

2. Capítulo II: Marco Teórico

2.1. Diccionario de datos

Abraham *et al.* (Abraham, Henry y S. Sudarshan, 2002) indican que los sistemas de bases de datos contienen un conjunto especial de tablas, denominado diccionario de datos o catálogo del sistema, que almacena datos acerca de la estructura de la propia base de datos. Adicionalmente, los sistemas de bases de datos consultan al diccionario antes de leer o modificar datos. Un diccionario de datos contiene metadatos, es decir, datos acerca de los datos, el esquema de una tabla es un ejemplo de metadatos.

Los lenguajes de definición de datos son un mecanismo utilizado para actualizar el diccionario de datos, por ejemplo, en una instrucción “crear tabla” se debe especificar el esquema y los atributos que la componen, toda esta información es almacenada en el diccionario de datos. Entre los tipos de información que debe guardar el sistema de base de datos son los siguientes:

- Los nombres de las relaciones.
- Los nombres de los atributos de cada relación.
- Los dominios y las longitudes de los atributos.
- Los nombres de las vistas definidas en la base de datos y las definiciones de esas vistas.
- Las restricciones de integridad.
- Los usuarios del sistema.
- Información estadística y descriptiva.

En la figura 5 se puede apreciar como el diccionario de datos forma parte de la arquitectura en los sistemas de bases de datos.

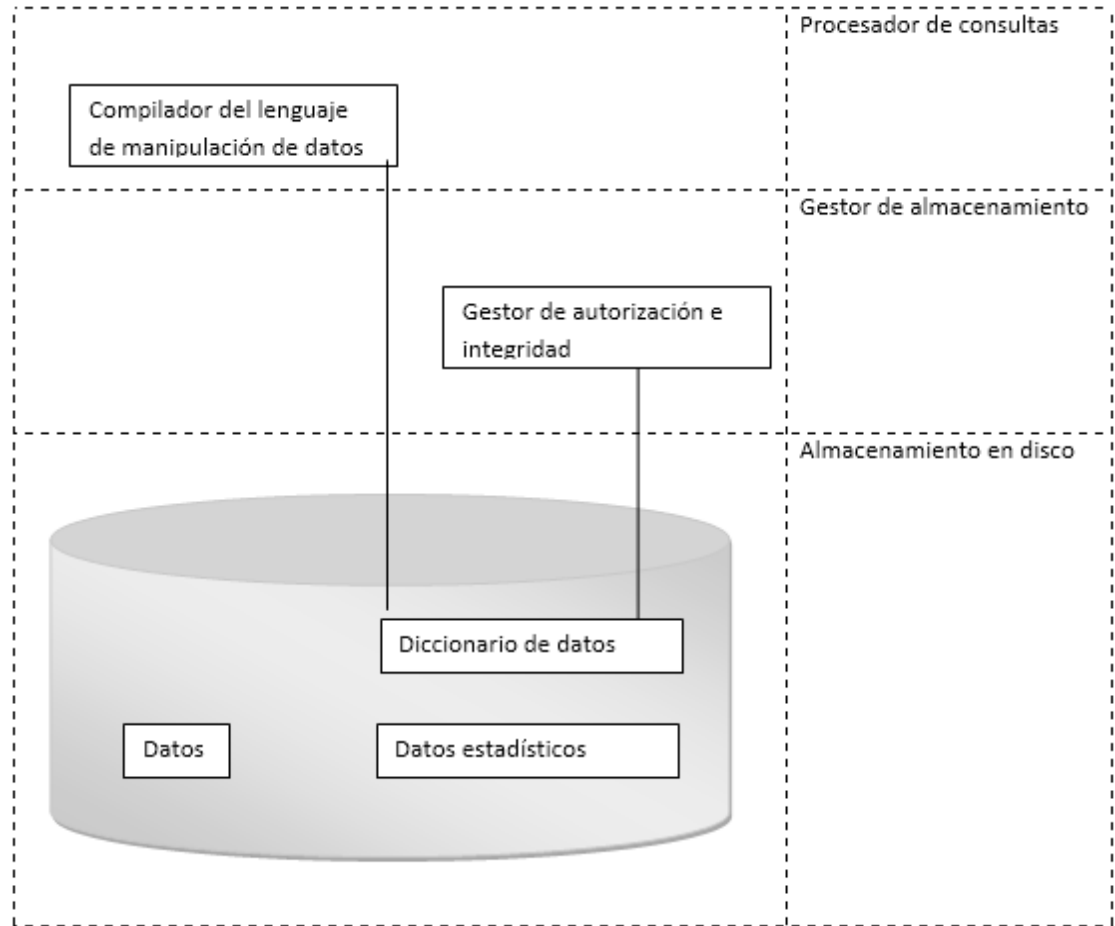


Figura 5 – Diccionario de datos en la arquitectura de bases de datos adaptado de Abraham *et al.* (Abraham, Henry y S. Sudarshan, 2002). Elaboración propia del investigador.

2.1.1. Metadatos

National Information Standards Organization (2015) indica que metadatos es información estructurada que describe, explica, localiza o de otra manera facilita recuperar, usar o administrar un recurso de información. Los metadatos usualmente son llamados datos acerca de los datos o información acerca de la información.

2.1.2. Tipos de metadatos

IBM (2011) indica que los metadatos se pueden clasificar en tres categorías, las cuales se citan a continuación:

Metadatos técnicos: consisten en la descripción técnica de los activos de datos, en esta categoría se pueden incluir las descripciones de tablas y sus diseños, la identificación y los atributos físicos de los orígenes y destinos de los almacenes de datos.

Metadatos de negocio: incluye la definición de términos de negocio y sus descripciones.

Metadatos operacionales: consiste en la información acerca de la ejecución de una aplicación o tarea, incluye información acerca del tiempo de inicio y finalización de los procesos, cantidad de registros procesados y rechazados, adicionalmente datos estadísticos sobre los trabajos. Este tipo de metadatos se utilizan principalmente para monitorear y medir el rendimiento de los procesos.

2.2. Gobernabilidad de los datos

MDM Institute (MDM Institute, 2014) indica que la gobernabilidad de los datos es "la orquestación formal de personas, procesos, y tecnología, que permite a las organizaciones aprovechar los datos como un activo empresarial". *Data Governance Institute (Data Governance Institute, 2015)* indica que "es el ejercicio de la toma de decisiones y la autoridad para los asuntos relacionados con los datos".

2.3. Administración de metadatos

Intricity (2014) indica que la administración de los metadatos permite dar trazabilidad sobre el flujo de los datos en procesos. Expone un ejemplo en donde desde el punto de vista de TI, se puede determinar cuáles reportes utilizan una tabla y desde el punto de vista de negocio se puede determinar de donde provienen los datos en los reportes. A lo descrito se le conoce como análisis de impacto y linaje de los datos, respectivamente.

2.4. Base de datos federada

2.4.1. Definición

Oracle (2002) indica que una base de datos federada es una vista unificada de distintas instancias de bases de datos que son ejecutadas en servidores independientes, los cuales no comparten recursos y se encuentran interconectados por la red.

Los datos se encuentran divididos horizontalmente a través de cada servidor participante (nodo), la única forma de acceder a los datos es por medio de solicitudes entre los nodos que conforman el sistema de federación; en este sentido, si uno de los nodos falla, los datos no estarán disponibles.

Barrios (2009) indica que una definición más reciente confirma que una base de datos federada es un sistema múltiple de base de datos en el cual cada nodo en la federación mantiene su autonomía en los datos y define un conjunto de esquemas de exportación, a través de los cuales se hacen disponibles los datos a otros nodos.

El término federación se refiere a la colección de bases de datos constituyentes que participan en una base de datos federada.

2.4.2. Características

Saltor (2014) indica que un sistema de bases de datos federadas debe cumplir con un conjunto de características, las cuales son: autonomía, heterogeneidad y sistemas distribuidos.

Barrios (2009) indica las definiciones de las características descritas por Saltor (2014), las cuales se listan a continuación:

- Autonomía se refiere a la capacidad de manejar su propio sistema de base de datos de manera separada e independiente.
- Heterogeneidad se refiere a que se debe permitir diferencias entre *hardware*, *software* y sistemas de comunicación.
- Distribución se refiere a que los datos pueden estar ubicados entre múltiples bases de datos.

2.4.3. Diferencia entre bases de datos distribuidas y bases de datos federadas

Barrios (2009) indica que la principal diferencia entre los sistemas de bases de datos distribuidas con respecto a las federadas radica que en federadas intervienen diferentes propietarios independientes que compartirán un esquema conceptual en común, aunque tengan diferentes tipos de fuentes de datos, mientras que en las distribuidas se pretende realizar una fragmentación de los datos en esquemas similares.

En la figura 6 se presenta una base de datos federada, en la figura 7 se presenta una base de datos distribuida.

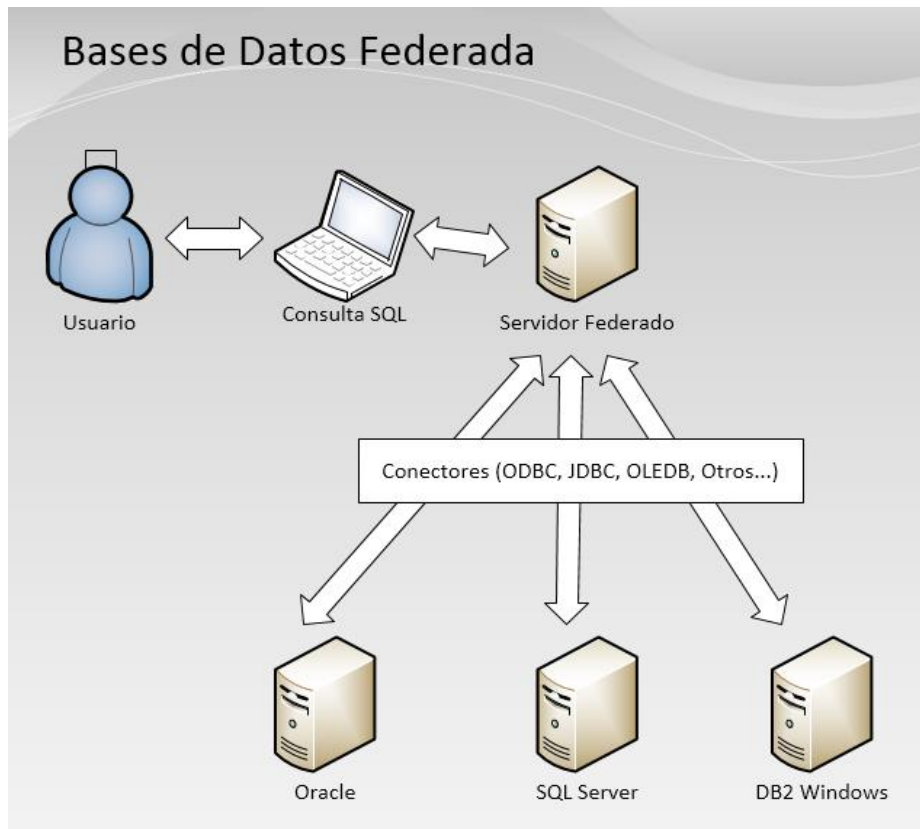


Figura 6: Base de datos federada (elaboración propia del investigador).

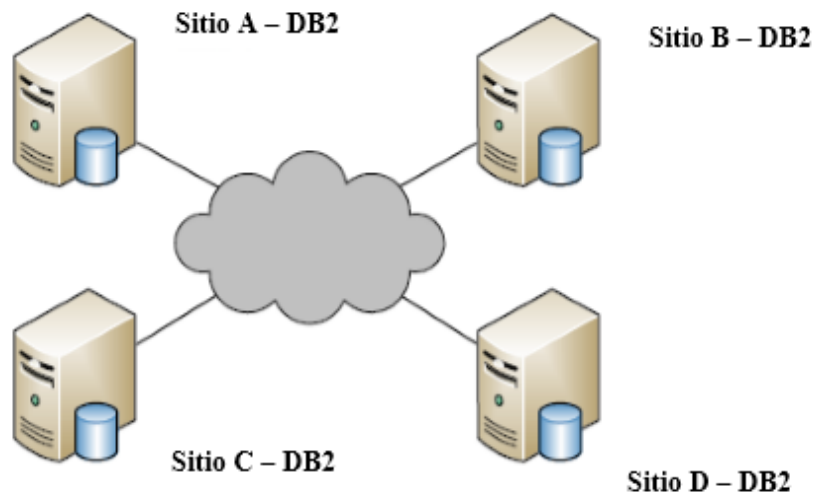


Figura 7: Base de datos distribuida (elaboración propia del investigador).

2.5. Modelamiento de datos

Kriegel (2011) indica que existen tres niveles principales de modelamiento de datos: conceptual, lógico y físico. Cada una de las capas se refiere a un grado de elaboración que culmina con un modelo que puede ser traducido en sentencias SQL y, a su vez, ser implementado en sistemas de bases de datos relacionales.

El modelo conceptual es el nivel más alto de abstracción, se ocupa principalmente de las entidades y sus relaciones sin entrar en detalle de especificar atributos, llaves primarias o foráneas, columnas y tipos de datos.

El modelo lógico es el nivel siguiente de abstracción, es construido con base en el modelo conceptual y se ocupa principalmente de la especificación de los atributos y las llaves primarias o foráneas.

El modelo físico consiste en tomar las ideas abstractas generadas a partir del modelo lógico para la construcción concreta de *scripts* SQL que pueden ser implementados en las bases de datos relacionales.

En la Tabla 4 se presentan las etapas y los elementos que se deben considerar para el desarrollo de los modelos de datos.

Tabla 4. Etapas para el desarrollo de modelos de datos (Kriegel, 2011).

Pasos	Conceptual	Lógico	Físico
Entidades	X		
Relaciones	X	X	
Atributos		X	
Llaves primarias		X	X
Llaves foráneas		X	X
Tablas/vistas			X
Columnas			X
Tipos de datos			X

Para la elaboración de este proyecto se utilizó la etapa del modelamiento de datos descrita por Kriegel (2011).

2.6. Almacén de datos y sus características

Inmon (1992) indica que un almacén de datos es una colección de datos orientada a temas, integrada, variable en el tiempo y no volátil, cuyo propósito es apoyar a la toma de decisiones gerenciales.

Orientado a temas: se almacena información resumida y significativa referente a temas específicos de interés, por ejemplo: servicios, clientes, créditos.

Integrados: integra la información de varios sistemas, proporcionando una visión coherente de la información.

Variante en el tiempo: se toman fotografías de datos en el tiempo para poder comparar los cambios ocurridos entre periodos de tiempo.

No volátiles: la información no se modifica, ni se elimina, solo se almacena para su posterior lectura.

Para este proyecto se utilizó un almacén de datos con la finalidad de almacenar un catálogo de objetos que integra todos los objetos contenidos en las bases de datos heterogéneas que posee la organización, para mostrar a los usuarios una vista unificada que les permita asociarla a palabras claves del diccionario de datos empresarial.

2.7. Procesos de extracción, transformación y carga

El proceso de extracción consiste en la lectura de los datos de las fuentes de información para almacenarlos en el área de pruebas para su posterior procesamiento.

El proceso de transformación consiste en realizar cambios a los datos obtenidos del proceso de extracción con la finalidad de aplicar limpieza de datos, eliminación de datos irrelevantes o combinar datos, las transformaciones pueden ser simples o complejas.

El proceso de carga consiste cargar los datos al almacén de datos, en donde ya se encontrarán disponibles para su utilización por parte de los usuarios finales.

En este proyecto se utilizaron procesos de extracción, transformación y carga.

2.8. Data Mart

Oracle (2007) indica que un mercado de datos es una forma simple de almacén de datos que se centra en un solo tema, tales como ventas, finanzas o mercadeo. Adicionalmente, los mercados de datos son consumidos normalmente por un solo departamento en una organización.

Inmon (1992) indica que existen dos tipos de *Data Marts*:

- Dependiente: se alimentan de un almacén de datos corporativo.
- Independiente: se alimenta de diferentes sistemas transaccionales.

En este proyecto se propone la creación de un mercado de datos para el almacenamiento de un catálogo de objetos integrado, el cual permita complementar al diccionario de datos empresarial.

2.9. Diseño multidimensional

Adamson (Adamson, 2010) indica que los sistemas de información se dividen en dos categorías, una que soporta la ejecución de los procesos de negocio (sistemas operacionales) y otra que soporta el análisis de los procesos de negocio (sistemas analíticos). Afirma que los principios del diseño multidimensional han surgido como una respuesta directa a las necesidades únicas de los sistemas analíticos, explica que el core de cada modelo dimensional es un conjunto de métricas de negocio que capturan cómo se evalúa un proceso y una descripción del contexto de cada medición, por lo que en un diseño multidimensional, las medidas son llamadas hechos y los descriptores de los contextos son llamados dimensiones. En la Tabla 5 se muestra una comparativa entre los sistemas operacionales y los analíticos.

Tabla 5. Comparativa entre los sistemas operacionales y analíticos, adaptado de Adamson (Adamson, 2010).

	Sistemas Operacional	Sistemas Analíticos
Propósito	Ejecución de procesos de negocio.	Medidas de los procesos de negocio.
Principal estilo de interacción	<i>Insert, update, select, delete</i>	<i>select</i>
Patrón de consultas	Predecible y estable	Impredecible y cambiante
Enfoque del tiempo	Actual	Actual e histórico
Optimización del diseño	Concurrencia en la actualización	Consultas de alto rendimiento.
Principio de diseño	Entidad - relación, diseñado en tercera forma normal	Diseño multidimensional (esquema de estrella o cubos)
Es conocido como	Sistemas transaccionales, sistemas de procesamiento en	Almacén de datos o <i>datamart</i>

línea.

Adamson (Adamson, 2010) indica que existen dos tipos de esquemas que pueden ser implementados en las bases de datos relacionales, la estrella y el copo de nieve. El esquema de estrella lleva su nombre debido a su apariencia, en la cual la tabla de hechos está en el centro y es rodeada por las tablas de dimensiones. El esquema de copo de nieve utiliza un modelado de datos normalizado en la definición de las dimensiones. En la figura 8 se muestra un ejemplo de esquema en forma de estrella y en la figura 9 se muestra un ejemplo de esquema copo de nieve.

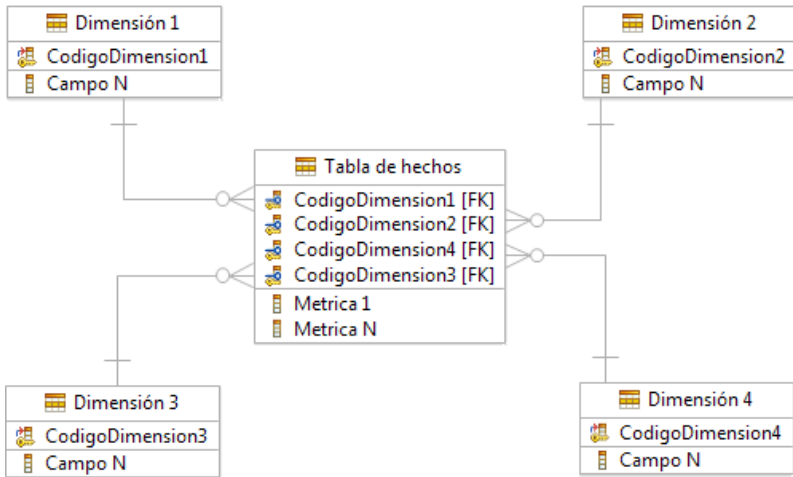


Figura 8. Esquema de estrella.

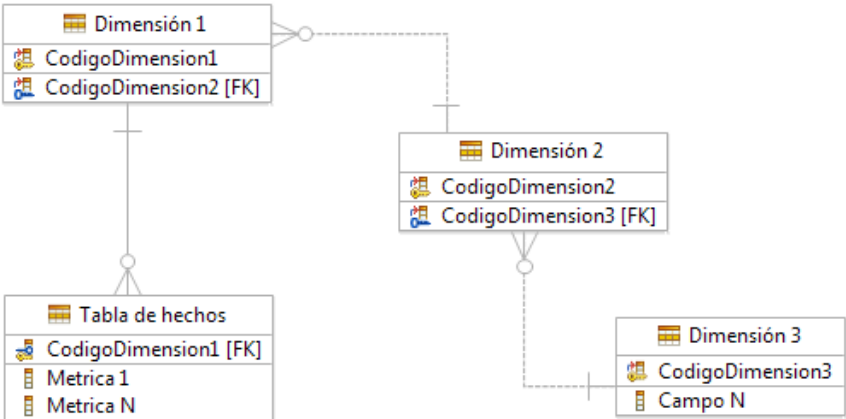


Figura 9. Esquema copo de nieve.

2.10. Cubos multidimensionales

Adamson (Adamson, 2010) indica que las bases de datos multidimensionales almacenan información dimensional en un formato llamado cubo. Esta forma de almacenamiento presenta ventajas sobre las bases de datos relacionales, una de ellas es la velocidad de respuesta ante el usuario. Darío (Darío, 2009) indica que un cubo multidimensional representa o convierte los datos planos que se encuentran en filas y columnas, en una matriz de N dimensiones. En la figura 10 se muestra una representación de cubo multidimensional.

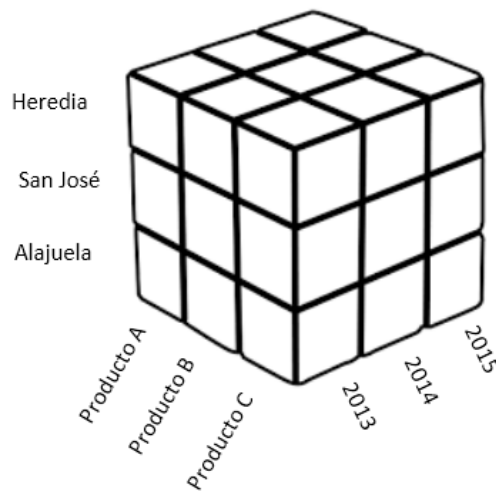


Figura 10. Representación de cubo multidimensional.

2.11. Área de datos de pruebas

Kimball (Kimball, 1998) indica que el área de pruebas es un espacio de trabajo del almacén de datos, en el cual se cargan los datos en bruto provenientes de los sistemas operacionales, cuyo propósito es trabajar los datos (transformarlos), de manera tal que permitan ser cargados en un servidor de presentación, el cual puede ser un sistema relacional o analítico.

2.12. Área de datos operacional

Inmon (Inmon W. H., 2005) indica que los almacenes de datos no pueden ser accedidos en milisegundos, por la naturaleza y el volumen de los datos que estos contienen, por lo que el almacén de datos no garantiza soportar procesamiento de transacciones en línea. Recomienda que en caso de requerir este tipo de procesamiento, se utilice el área de datos operacional, ya que este puede realizar este tipo de procesamiento. Comenta que el área de datos operacional es opcional; sin embargo, puede complementar al almacén de datos. Argumenta que una de las características del almacén de datos es que no está optimizado para realizar operaciones de actualización, esto debido a que fue diseñado para almacenar datos históricos que no cambian (no hay necesidad de actualizarlos). Con base en el tiempo requerido de actualización del almacén de datos operacional, los clasifica de la siguiente manera:

- Clase 1: Se actualiza en un periodo de tiempo de milisegundos.
- Clase 2: Se actualiza en un periodo de tiempo de horas.
- Clase 3: Se actualiza en un periodo de tiempo de días.
- Clase 4: Se actualiza en un periodo de tiempo de meses o años.

2.13. Herencia en el modelo relacional

González (González Alvarado, 2009) indica que la herencia en el modelo relacional, busca afinar una entidad llamada super-entidad en otras más especializadas, llamadas sub-entidades, como estas representan a los mismos objetos del mundo real, deben contar con la misma llave primaria. En la figura 11 se muestra un ejemplo en el cual se realiza una especialización de la tabla personas, en las tablas de clientes y empleados, cada una de estas con nuevos atributos.

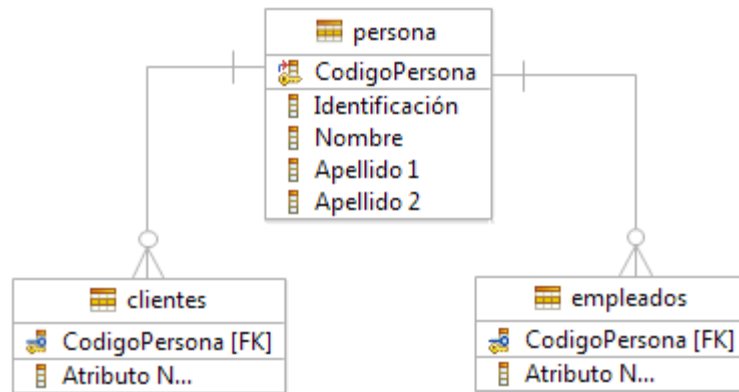


Figura 11. Ejemplo de herencia en el modelo relacional.

2.14. SQL Incorporado

IBM (IBM, 2001) indica que las sentencias de SQL incorporadas son escritas en lenguajes de programación de aplicaciones, tales como C o java; antes de compilar un programa, un precompilador SQL preprocesa estas sentencias. Existen dos tipos de SQL incorporado, el estático y dinámico. En SQL estático, la sentencia se prepara antes de la ejecución del programa y esta persiste después de la ejecución. En SQL dinámico, las sentencias se construyen y preparan en tiempo de ejecución.

3. Capítulo III: Marco Metodológico

3.1. Tipo de investigación

Este trabajo se trata de una investigación evaluativa, ya que se desea evaluar si el modelo propuesto es funcional con base en requerimientos. Hurtado de Barrera (2005) señala que este tipo de investigación tiene como propósito valorar los resultados de una intervención en términos de sus objetivos, que para nuestro estudio en cuestión se basa en los requerimientos del modelo propuesto.

3.2. Alcance de la investigación

Esta investigación tiene un alcance exploratorio, dado que se realizaron pruebas para demostrar que es posible alcanzar los objetivos planteados.

3.3. Enfoque de la investigación

Esta investigación utiliza el enfoque de abordaje alternativo. Desde la dimensión epistemológica se utilizará ingeniería de requerimientos para interpretar las necesidades por parte de los usuarios. Desde la dimensión ontológica, se parte del concepto expresado por Gruber (1993): "lo que existe es exactamente aquello que puede ser representado". La representación ontológica del tipo de implementación que se propone se puede apreciar en la figura 12.

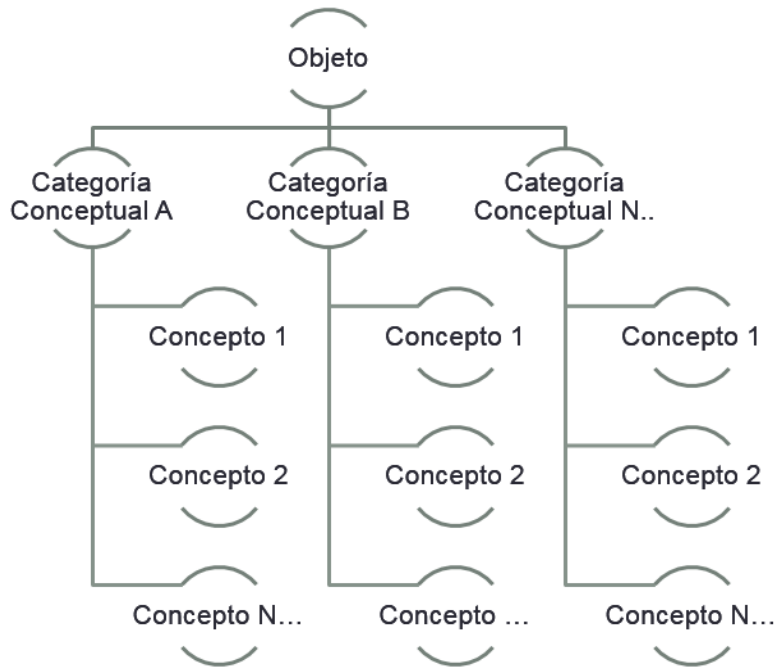


Figura 12. Ontología (elaboración propia del investigador)

Es por ello que se considera que el estudio propuesto tiene una posibilidad clara de representación del conocimiento que se desea producir.

Desde la dimensión axiológica requiere especial tratamiento en cuanto a la funcionalidad del modelo, la cual será evaluada de acuerdo con la escala de valores mostrada en la Tabla 6.

Tabla 6. Criterios de evaluación del modelo de datos.

Funcionalidades a evaluar en el modelo					
ID	Funcionalidad	Valor %	Cantidad total de escenarios de pruebas	Cantidad de escenarios de prueba OK	Porcentaje de cumplimiento de funcionalidad
1	Permite identificar los objetos de	5,55	27	27	100

	diferentes bases de datos de manera integrada.				
2	Permite identificar, de forma única, los conceptos del diccionario.	5,55	4	4	100
3	Permite agrupar conceptos del diccionario en jerarquías.	5,55	2	2	100
4	Permite asociar objetos con conceptos del diccionario.	5,55	1	1	100
5	Permite asociar objetos con propietarios.	5,55	6	6	100
6	Permite asociar objetos con controles.	5,55	3	3	100
7	Permite asociar objetos observación de los propietarios.	5,55	1	1	100
8	Permite consultar información sobre los servidores remotos vinculados.	5,55	3	3	100
9	Permite registrar trazabilidad de los procesos del sistema.	5,55	1	1	100
10	Permite definir las bases de datos a las cuales realizar el proceso de extracción.	5,55	1	1	100
11	Permite definir los objetos participantes en el proceso de extracción.	5,55	1	1	100
12	Permite consultar objetos por conceptos definidos en el diccionario.	5,55	1	1	100
13	Permite consultar objetos por	5,55	2	2	100

	propietarios.				
14	Permite consultar objetos por controles.	5,55	1	1	100
15	Permite consultar objetos por observaciones.	5,55	1	1	100
16	Permite escalabilidad en el versionamiento de gestores de bases de datos.	5,55	1	1	100
17	Permite escalabilidad funcional de forma modular.	5,55	4	4	100
18	Permite utilizar mecanismos de autenticación, para realizar las consultas.	5,55	1	1	100

El valor porcentual de cada funcionalidad se calcula de la siguiente manera: 100 dividido entre la cantidad de funcionalidades por probar, lo que da como resultado $100/18 = 5,55\%$. La fórmula para el cálculo del porcentaje de cumplimiento de funcionalidad es $(\text{Cantidad de escenarios de prueba OK} * 100) / \text{cantidad de escenario de pruebas por requerimiento}$, posteriormente se realiza un promedio simple con la fórmula: $\text{suma de porcentajes de cumplimiento de funcionalidad} / (\text{cantidad de funcionalidades} * 100) * 100$.

3.4. Sujetos y fuentes de información

3.4.1. Fuentes de información

Las principales fuentes de información utilizadas para este proyecto son:

- Documentación oficial de los vendedores de bases de datos IBM, *Oracle* y *Microsoft*.
- Documentación brindada por el área de arquitectura de bases de datos.

- Documentos, libros y artículos que brinde apoyo referente a temas de diccionario de datos, almacenes de datos.
- Reuniones con el patrocinador.
- Reuniones con el personal del área de arquitectura de bases de datos.

3.4.2. Sujetos de información

Los sujetos de información utilizadas para este proyecto son:

- Involucrados en el proyecto.
- Los diferentes motores de bases de datos relacionales que utiliza BAC Credomatic Network (DB2, *SQL Server*, *Oracle*).

3.5. Descripción de instrumentos

En esta investigación se utilizarán los siguientes instrumentos:

- Observación: permite conocer los procesos.
- Grupos de discusión: permite conocer diferentes puntos de vistas de otros profesionales.
- Documento de especificación de requerimientos del sistema (apéndice 1).
- Documento de plan de pruebas (apéndice 5).

4. Capítulo IV: Desarrollo de Prototipo

4.1. Metodología seleccionada para el desarrollo del proyecto

Para el desarrollo del proyecto se utilizó la metodología *rapid warehousing* expuesta por *SAS Institute* (2002), ya que permite el desarrollo de un almacén de datos a corto plazo implementando cinco fases, tal como se puede apreciar en la figura 13.

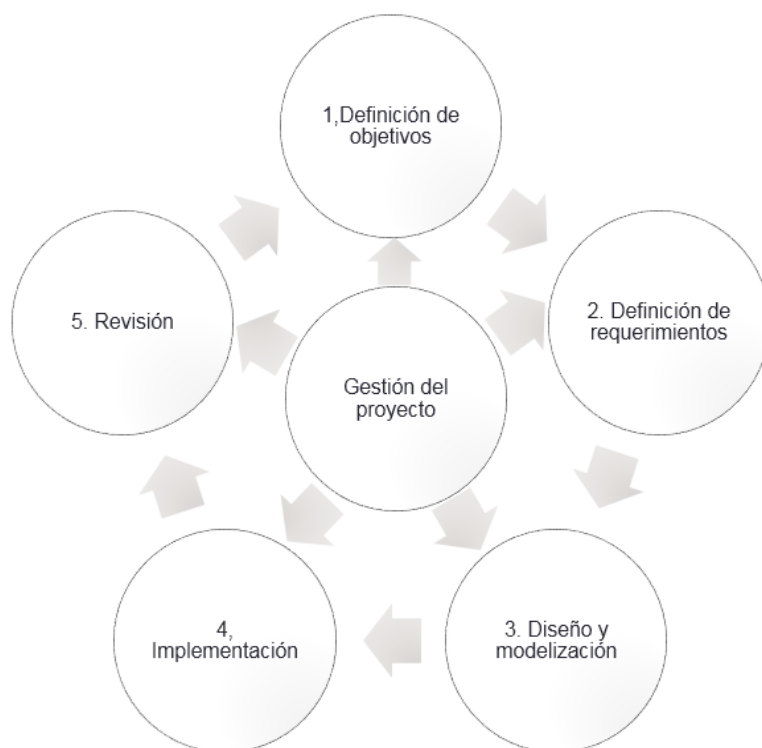


Figura 13. Fases de la metodología *rapid warehousing* (adaptación de *SAS Institute*. Elaboración propia del investigador)

Espinoza (2009) describe cada una de las fases de la metodología, tal como se explica a continuación:

Definición de los objetivos: se definirá el equipo del proyecto, el alcance del sistema y cuáles son sus funciones.

Definición de los requerimientos de información: se realizará el estudio de los sistemas de información existentes, que ayudarán a comprender las carencias actuales y futuras que deben ser resueltas en el diseño del almacén de datos.

Diseño y modelización: se identifican las fuentes de datos y las transformaciones necesarias para, a partir de dichas fuentes, obtener el modelo lógico y físico de datos del almacén de datos.

Implementación: la implementación tiene implícita la extracción y transformación de los datos de los orígenes, la carga de los datos validados en el almacén de datos y la explotación del almacén de datos por medio de diversas técnicas, tales como reportes, consultas y visualización de la información entre otras. Al final de esta etapa, el almacén de datos estará disponible a los usuarios.

Revisión: se definen cuáles son los aspectos a mejorar de manera iterativa para mejorar la funcionalidad que ofrece el almacén de datos.

4.2. Selección de miembros del equipo de trabajo

Después de múltiples reuniones con el patrocinador y las partes interesadas, se definió el equipo de trabajo que participó en el desarrollo del proyecto. Se seleccionaron profesionales de diferentes áreas para contar con diferentes puntos de vista, los cuales abarcan desde las áreas de negocio hasta las diversas áreas técnicas para la posterior definición de los objetivos y requerimientos. Se genera una matriz de roles y responsabilidades (RACI), como se muestra en la Tabla 7.

Tabla 7 – Matriz RACI (elaboración propia del investigador)

Leyenda									
R	Responsable	Ejecuta tareas							
A	Aprobador	Aprueba tareas							
C	Consultado	Consultados para la realización de tareas							
I	Informado	Informado sobre el avance de las tareas							
Entregable/Hito			Patrocinador	Arquitecto de BI	Seguridad de sistemas	Arquitecto de base de datos	Analista de negocio	Desarrollador de aplicaciones	Estudiante
Fase - Planificación									
Carta del proyecto			R						I
Plan de comunicación			A	I	I	I	I	I	R
Crear los equipos de trabajo			R/A	I	I	I	I	I	I
Definición de Objetivos									
Definir los objetivos			C	C	C	C	C	C	R
Revisar los objetivos			A	C	C	C	C	C	I
Aprobar los Objetivos			R/A	I	I	I	I	I	I
Definición de requerimientos									
Definir requerimientos del sistema			C	C	C	C	C	C	R
Revisión de los requerimientos			C	C	C	C	C	C	R
Matriz de trasabilidad de requerimientos			I	I	I	I	I	I	R
Aprobación de los requerimientos			A	R	R	R	R	R	I
Diseño									
Modelado de datos lógicos						C			R
Modelado de datos físicos						C			R
Modelado de arquitectura				C	C	C			R
Modelado del proceso de extracción				C					R
Modelado del proceso de transformación				C					R
Modelado del proceso de carga				C					R
Modelado del proceso de herramientas de visualización							C		R
Modelado de pruebas					C	C	C	C	R
Revisión del diseño			A	C	C	C	C	C	R
Implementación									
Construcción del diseño			C	C	A		C	R	R
Ejecución de pruebas			C	R				C	C
Análisis de los resultados de las pruebas			C	C	A		C	R	R
Aprobación de las pruebas			R	C	C	C	C	C	I
Revisión									
Revisión de pruebas			I	I	I	I	I	I	R
Análisis de los resultados de las pruebas			I	I	I	I	I	I	R

Fuente: basado en ITIL V3 (2014).

Se define un plan de comunicación para dar a conocer al equipo de trabajo, la frecuencia y los temas a tratar en las sesiones. En la Tabla 8 se muestra el plan de comunicación.

Tabla 8. Plan de comunicación (elaboración propia del investigador)

Plan de comunicación			
Participantes	Frecuencia	Foro	Temas a tratar
Equipo de proyecto	Mensual	Estado del proyecto	Progreso de tareas, identificación de problemas.
Patrocinador	Mensual	Sesiones informativas	Progreso en general del proyecto, revisión de objetivos.
Colega de TI	Quincenal	Sesiones técnicas	Progreso del sistema, revisiones técnicas.

A partir de la definición del equipo de trabajo se realizan sesiones posteriores en donde se generan “lluvias de ideas” para comprender la necesidad de la empresa. Estas ideas se fueron refinando hasta convertirse en los objetivos y requerimientos especificados en el documento *“Especificación de Requerimientos del Sistema”*, adjunto en el apéndice 1.

4.3. Definición de objetivos

Para la definición de los objetivos se utiliza el documento de especificación de requerimientos del sistema, que se encuentra en el apéndice 1. El documento cuenta con una sección para la definición de los objetivos, cada uno de los objetivos tiene un identificador único, una fecha de creación, los solicitantes, una descripción de lo que se quiere lograr y, adicionalmente, un comentario.

Se verifica de manera conjunta con el patrocinador, que la definición de los objetivos se encuentre dentro del alcance de este proyecto.

4.4. Definición de requerimientos

Para el proceso de recolección de los requerimientos se utiliza el proceso unificado racional propuesto por Jacobs (2003), en el cual se enumeran los requisitos candidatos, se entiende el contexto del sistema y se capturan los requisitos funcionales.

Para la definición de los requerimientos se utiliza como herramienta el documento “Especificación de Requerimientos del Sistema”, que se encuentra en el apéndice 1.

Los requerimientos se asocian a un objetivo en específico para facilitar la trazabilidad en la relación de estos, son validados por los integrantes del equipo de trabajo y son debidamente aprobados por el patrocinador. En la figura 14 se muestra el diagrama de casos de uso empleado para el proyecto.

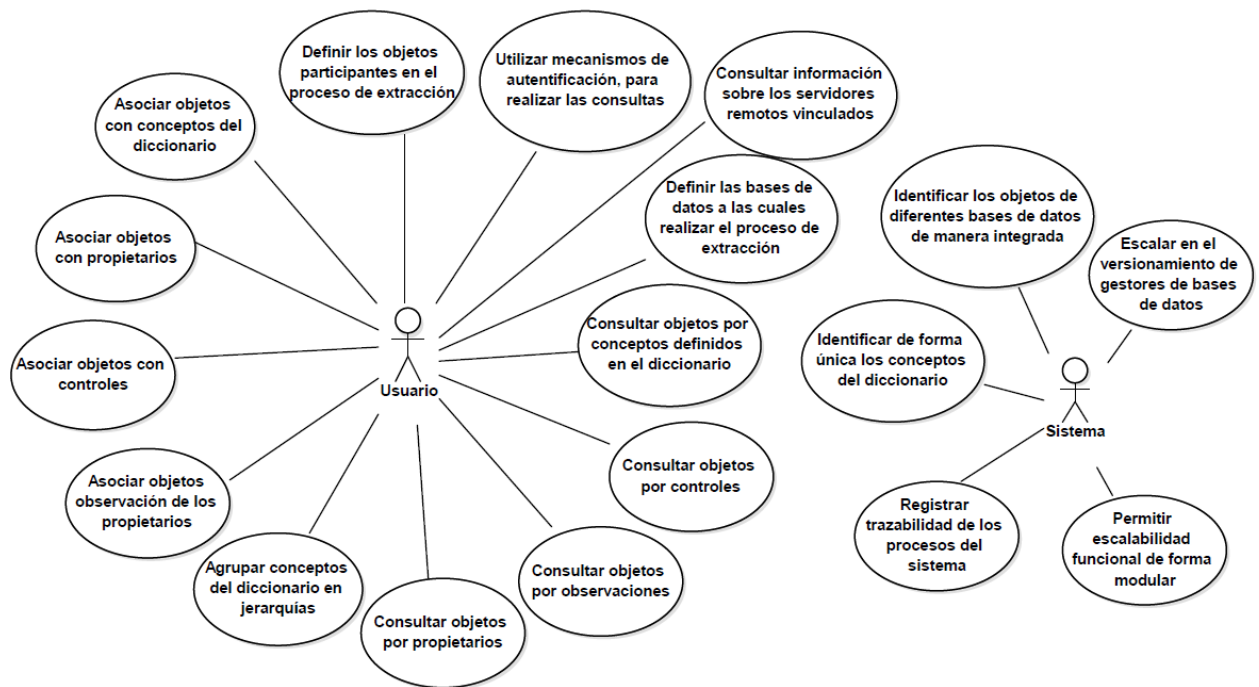


Figura 14. Diagrama de casos de uso (elaboración propia del investigador).

4.5. Diseño y modelización

A continuación, se presentan los componentes que conforman el modelo propuesto.

4.5.1. Modelamiento de la arquitectura

En la figura 15 se muestran los componentes que conforman la arquitectura de la solución propuesta.

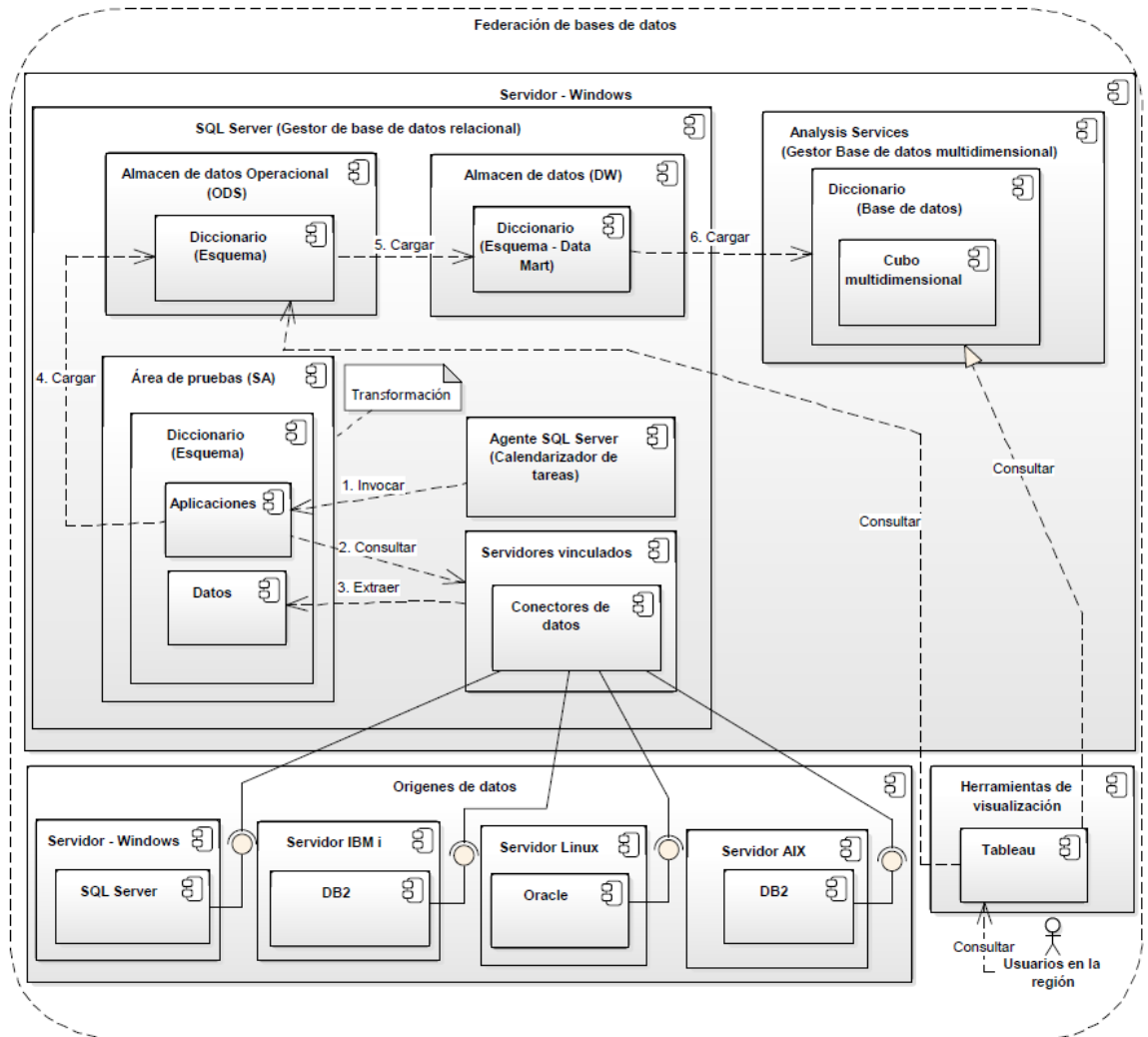


Figura 15. Arquitectura de la solución. Diagrama de componentes.

Los componentes de la arquitectura se detallan a continuación:

- Orígenes de datos: están conformados por los gestores de bases de datos relacionales DB2, Oracle y SQL Server, los cuales están

instalados sobre plataformas heterogéneas, tales como *Windows*, *IBM i*, *Linux* y *AIX*.

- Los servidores vinculados: son gestores de bases de datos que se encuentran interconectados con una instancia de *SQL Server*, estos forman parte de las tecnologías de federación de datos que tiene incorporado este gestor. Esta característica facilita el proceso de comunicación entre bases de datos heterogéneas, a través de los conectores de datos.
- Conector de datos: controlador proporcionado por los fabricantes de los gestores de bases de datos, que permite el acceso a estos. Para el caso en estudio se utilizan controladores ODBC (*Open DataBase Connectivity*), proporcionados por los fabricantes *IBM*, *Oracle* y *Microsoft*.
- Agente *SQL Server*: servicio que permite la ejecución de tareas programadas. Para el caso en estudio se utiliza este componente para calendarizar la ejecución de los procesos de extracción.
- Área de prueba: es un espacio de trabajo, en el cual se realizan extracciones de datos provenientes de los orígenes. Tiene como finalidad evitar cargas de trabajo relacionadas con los procesos de transformación e integración de los datos; en las fuentes, permite controlar y corregir errores que pueden ocasionarse durante los procesos de extracción, transformación y carga de los datos hacia el almacén de datos. La solución propuesta contempla la creación de una base de datos “Diccionario_SA”, el cual contiene un esquema “diccionario”. Bajo este esquema se almacenan aplicaciones y datos relacionados con la solución.
- Almacén de datos operacional (ODS): contiene información transformada e integrada proveniente de los sistemas transaccionales, la cual puede ser actualizada frecuentemente (casi en línea). Se utiliza para poblar el modelo de datos relacional propuesto. En el

caso de los usuarios, realizan relaciones entre objetos de bases de datos con conceptos, propietarios, controles y comentarios. La solución propuesta contempla la creación de una base de datos “Diccionario_ODS”, la cual contiene un esquema “diccionario”, bajo este esquema se almacenan los datos relacionados con el diccionario de datos empresarial.






- Almacén de datos: almacena información de forma histórica e integrada. Se utiliza para poblar el modelo de datos multidimensional propuesto, con los datos provenientes del almacén de datos operacional, correspondiente al tema del diccionario de datos empresarial. La solución propuesta contempla la creación de una base de datos “Diccionario DW”, la cual contiene un esquema “diccionario”. Este esquema es el *data mart*, que contiene información sobre el tema específico del diccionario de datos empresarial.
- *Data mart*: es una parte del almacén de datos que expone un tema en específico; en nuestro caso, un diccionario de datos empresarial almacenado en el esquema diccionario, el cual contiene un modelo de datos multidimensional (estrella).
- *Analysis services*: es un gestor de bases de datos multidimensional, en el cual se crea una base de datos “Diccionario”. Esta contiene y expone a los clientes, los cubos multidimensionales, los cuales son poblados con la información contenida en el *data mart* del diccionario de datos empresarial.
- *Tableau*: herramienta que permite la visualización de los datos contenidos en los cubos multidimensionales expuestos en el gestor *analysis services* o en el almacén de datos operacional.

4.5.2. Modelado de datos

El modelo de datos se descompone en varios diseños, los cuales se enfocan en la solución de requerimientos específicos establecidos en el documento de especificación de requerimientos del sistema (véase apéndice 1). Para el modelado de los datos se utilizó la notación *Crow's Foot*. La simbología empleada en los diagramas se detalla en la Tabla 9. La lista de diseños generados son los siguientes:

- D001-Administración de servidores, plataformas, gestores e instancias.
- D002-Administración de bases de datos y esquemas por instancia.
- D003-Procesos por instancia.
- D004-Registro de colaboradores.
- D005-Propietarios por objeto.
- D006-Comentarios por propietario.
- D007-Conceptos por objeto.
- D008-Controles por objeto.
- D009-Servidores por ubicación geográfica.
- D010-Administradores por servidores.
- D011-Administradores por instancia de gestores de base de datos.

Tabla 9. Simbología empleada en los diagramas de datos.

Símbolo	Descripción
	Tabla
	Llave primaria
	Llave foránea
	Llave primaria y foránea
	Columna

En el apéndice 2 se muestra el diagrama de datos conceptual del modelo relacional, y en el apéndice 3 se muestra el diagrama de datos físico del modelo relacional.

4.5.2.1. Diseño - D001 - Servidores, plataformas, gestores e instancias

4.5.2.1.1. Especificación del diseño

Este diseño permite administrar información específica sobre plataformas, instancias de bases de datos y aplicaciones que utilizan los servidores.

Se contemplan los siguientes requerimientos:

- R-008. Consultar información sobre los servidores remotos vinculados.
- R-016. Escalabilidad en versionamiento de gestores.

4.5.2.1.2. Propuesta del diseño

4.5.2.1.2.1. Modelo conceptual

En la figura 16 se muestra el diseño conceptual, en la Tabla 10 se detallan las entidades, y en la Tabla 11 se detallan las relaciones.

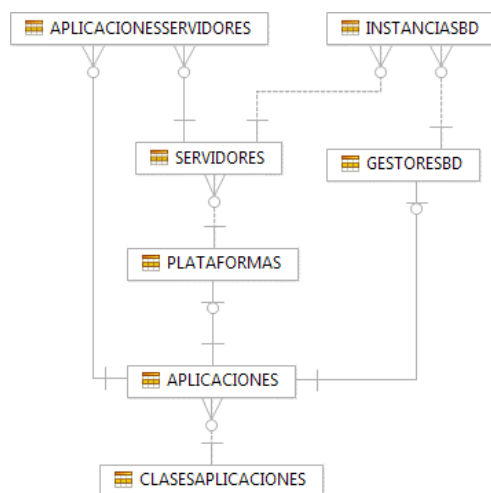


Figura 16. Modelo conceptual del diseño D001. Servidores, plataformas, gestores e instancias.

Tabla 10. Descripción de las entidades del modelo lógico para el diseño D001. Servidores, plataformas, gestores e instancias.

Descripción de las entidades		
ID	Entidad	Descripción
1	clasesAplicaciones	Clases de aplicaciones existentes para categorizar el <i>software</i> . Por ejemplo: gestores de bases de datos, plataformas, antivirus, ofimática.
2	aplicaciones	Programas de computadora, tales como <i>Norton antivirus, Office 2014, Oracle 12G</i> .
3	gestoresBd	Aplicaciones especializadas en la administración de sistemas de bases de datos, tales como <i>DB2, Oracle y SQL Server</i> .
4	plataformas	Aplicaciones utilizadas como sistemas operativos, tales como <i>Linux, Windows, AIX, OS400</i> .
5	servidores	Computadores especializados en brindar servicios, tales como bases de datos, correo electrónico y páginas Web.
6	aplicacionesServidores	Aplicaciones instaladas en un servidor; por ejemplo, un antivirus o <i>software</i> para la realización de respaldos.
7	instanciasBd	Son los servicios que expone un motor de base de datos. Por ejemplo: un computador tiene instalados los gestores <i>DB2 y Oracle</i> . Ambos son servicios independientes, y cada uno de estos servicios es considerado una instancia de bases de datos.

Tabla 11. Descripción de las relaciones entre entidades para el diseño D001. Servidores, plataformas, gestores e instancias.

Descripción de las relaciones			
ID	Entidad	Cardinalidad	Descripción
1	Aplicaciones – clasesAplicaciones	1:N	Una clase de aplicación se utiliza para catalogar cero, una o más aplicaciones.
2	gestoresBd - aplicaciones	1:1	Una aplicación puede registrarse de cero a una vez como gestor de base de datos.
3	Plataformas - aplicaciones	1:1	Una aplicación puede registrarse de cero a una vez como plataforma.
4	Servidores - plataformas	1:N	Una plataforma es utilizada por cero, uno o más servidores.
5	AplicacionesServidores - servidores	1:N	Un servidor utiliza cero, una o más aplicaciones.
6	aplicacionesServidores - aplicaciones	1:N	Una aplicación es utilizada por cero, uno o más servidores.
7	instanciasBd - Servidores	1:N	Un servidor puede tener instalado de cero, una o más instancias de bases de datos.
8	instanciasBd - gestoresBd	1:N	Un gestor de base de datos es utilizado por cero, una o más instancias de bases de datos.

4.5.2.1.2.2. Modelo lógico

En la Tabla 12 se detalla el diseño del modelo lógico.

Tabla 12. Modelo de datos lógico para el diseño D001. Servidores, plataformas, gestores e instancias.

Entidad:	clasesAplicaciones		
Descripción de atributos			
Atributo	Clave primaria	Clave foránea	Descripción
codigoClaseAplicacion	X		Identificador de la clase de aplicación.
Nombre			Nombre de la clase de aplicación. Por ejemplo: el <i>software</i> se puede clasificar bajo los nombres de: gestores de bases de datos, plataformas, antivirus, ofimática.
Descripción			Descripción de la clase de aplicación.
Restricciones			
Descripción	Tipo	Descripción	
clasesAplicaciones_uchk	Unicidad	Los nombres de las clases de aplicaciones deben ser únicos.	
Comentarios adicionales			
Ninguno.			

Entidad:	aplicaciones		
Descripción de atributos			
Atributo	Clave primaria	Clave foránea	Descripción
codigoAplicacion	X		Identificador de la aplicación.
codigoClaseAplicacion		X	Identificador de la clase de aplicación.
Nombre			Nombre de aplicación.

Versión			Versión de la aplicación.
Proveedor			Proveedor de la aplicación. Por ejemplo, IBM, <i>Microsoft</i> .
Restricciones			
Descripción	Tipo	Descripción	
aplicaciones_clase sAplicaciones_fk	clave foránea	El campo codigoClaseAplicacion debe hacer referencia al campo de la tabla clasesAplicaciones.	
aplicaciones_uchk	unicidad	La combinación de los campos nombre y versión deben ser únicos.	
Comentarios adicionales			
Esta tabla permite registrar cualquier tipo de <i>software</i> , incluyendo plataformas (sistemas operativos) y gestores de bases de datos.			

Entidad:	gestoresBd		
Descripción de atributos			
Atributo	Clave primaria	Clave foránea	Descripción
codigoGestordb	X	X	Identificador de aplicación gestora de base de datos.
Restricciones			
Descripción	Tipo	Descripción	
gestoresbd_aplicaciones_FK	Clave foránea	El campo codigoGestorBd debe hacer referencia al campo codigoaplicacion de la tabla aplicaciones.	
Comentarios adicionales			
Ninguno.			

Entidad:	plataformas		
Descripción de atributos			

Atributo	Clave primaria	Clave foránea	Descripción
codigoPlataforma	X	X	Identificador de aplicaciones correspondientes a sistemas operativos.
Restricciones			
Descripción	Tipo	Descripción	
plataformas_aplicaciones_FK	Clave foránea	El campo codigoplataforma debe hacer referencia al campo codigoaplicacion de la tabla aplicaciones.	
Comentarios adicionales			
Un servidor físico o virtualizado posee únicamente una plataforma (sistema operativo).			

Entidad:	servidores		
Descripción de atributos			
Atributo	Clave primaria	Clave foránea	Descripción
codigoServidor	X		Identificador del servidor.
codigoPlataforma		X	Identificador de la plataforma (sistema operativo) que utiliza el servidor.
Nombre			Nombre de servidor.
Descripción			Descripción detallada del servidor.
Dirección red			Dirección en la red del servidor.
Virtualizado			Indica si un servidor es virtual o no. Los posibles valores de este campo son "S" (Sí) o "N" (No).
Restricciones			
Descripción	Tipo	Descripción	
servidores_platafo	Clave	El campo codigoPlataforma debe hacer referencia	

rmas_fk	foránea	al campo codigoPlataforma de la tabla plataformas.
servidores_virtualizado_chk	Comprobación	El campo virtualizado admite los valores “N” (No) o “S” (Sí).
servidores_uchk	Unicidad	El campo nombre debe ser único
Comentarios adicionales		
Los servidores, tanto físicos como virtuales, solo pueden tener asociado un sistema operativo (plataforma), pero admiten la instalación de múltiples instancias de bases de datos.		

Entidad:	aplicacionesServidores		
Descripción de atributos			
Atributo	Clave primaria	Clave foránea	Descripción
codigoServidor	X	X	Identificador de servidor.
codigoAplicación	X	X	Identificador de aplicación.
Restricciones			
Descripción	Tipo	Descripción	
aplicacionesServidores_servidores_fk	Clave foránea	El campo codigoservidor debe hacer referencia al campo codigoservidor de la tabla servidores.	
aplicacionesServidores_aplicaciones_fk	Clave foránea	El campo codigoAplicacion debe hacer referencia al campo codigoAplicacion de la tabla aplicaciones.	
Comentarios adicionales			
Ninguno.			

Entidad:	instanciasBd		
Descripción de atributos			
Atributo	Clave primaria	Clave foránea	Descripción
codigoInstanciaBd	X		Identificador de la instancia de base de datos.
codigoServidor		X	Identificador del servidor.
codigoGestorBd		X	Identificador del gestor de base de datos.
Nombre			Nombre de la instancia.
Descripción			Descripción de la instancia de base de datos.
descubrirCatalogoBd			Indica si el sistema debe realizar un análisis de los catálogos de bases de datos en una instancia de base de datos, con la finalidad de descubrir nuevos catálogos. Los posibles valores de este campo son “S” (Sí) o “N” (No).
Restricciones			
Descripción	Tipo	Descripción	
instanciasBd_servidores_fk	Clave foránea	El campo codigoservidor debe hacer referencia al campo codigoservidor de la tabla servidores.	
instanciasBd_gestoresBd_fk	Clave foránea	El campo codigoGestorBd debe hacer referencia al campo codigoGestorBd de la tabla gestoresBd.	
instanciasBd_uchk	Unicidad	La combinación de los campos codigoServidor, codigoGestorBd y Nombre debe ser única.	
instanciasBd_descubrirCatalogoBd_chk	Comprobación	El campo descubrirCatalogoBd admite los valores “N” (No) o “S” (Sí).	

Comentarios adicionales
Ninguno.

4.5.2.1.2.3. Modelo físico

En la figura 17 se muestra el modelo de datos físico.

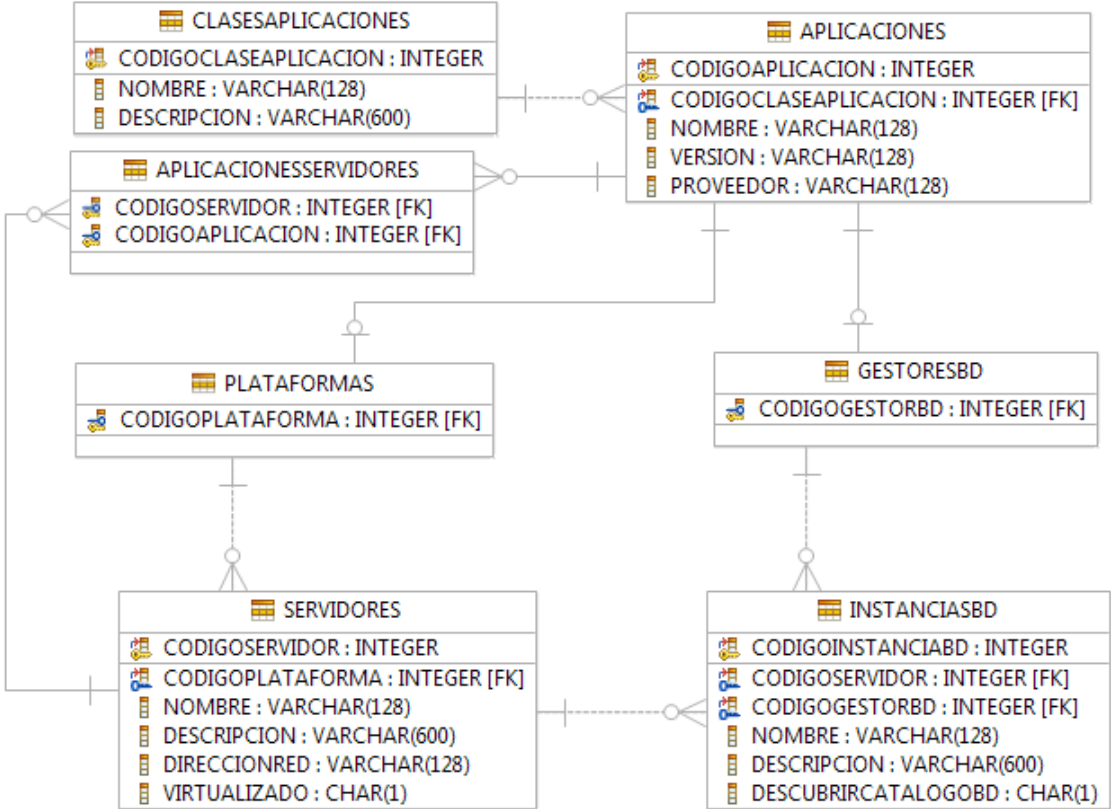


Figura 17. Modelo físico del diseño D001. Servidores, plataformas, gestores e instancias.

4.5.2.1.3. Justificación del diseño

Para este diseño se hace uso del concepto de herencia expuesto por González Alvarado (2009), ya que permite evitar la redundancia de campos en las tablas hijas (especializadas), por lo que se parte de una generalización de la

tabla aplicaciones y se realizan especializaciones en las tablas gestoresBd y plataformas con la siguiente finalidad:

- La especialización de la tabla gestoresBd permite que esta contenga únicamente aplicaciones gestoras de bases de datos, las cuales se pueden referenciar en la tabla InstanciaBd, en el campo codigoGestorBd, ya que si se permite referenciar la tabla de aplicaciones en el campo codigoAplicacion directamente, el diseño admite que cualquier aplicación pueda relacionarse como un gestor de base de datos. Por ejemplo, se podría decir que *Microsoft Word* es el gestor de base de datos para una instancia, lo cual no es correcto.
- La especialización de la tabla plataformas permite que esta contenga únicamente aplicaciones que son plataformas (sistemas operativos), las cuales se pueden referenciar en la tabla servidores en el campo codigoPlataforma, ya que si se permite referenciar la tabla de aplicaciones directamente en el campo codigoAplicacion, el diseño admitiría que cualquier aplicación se pueda relacionar como una plataforma en un servidor. Por ejemplo, se podría decir que *Microsoft Word* es la plataforma (sistema operativo) que utiliza un servidor, lo cual no es correcto.

El modelo cumple con lo establecido en el requerimiento R-008, consultar información sobre los servidores remotos vinculados, debido a que permite:

- Consultar el gestor de bases de datos utilizado por una instancia en un servidor.
- Consultar la plataforma que utiliza un servidor.

El modelo cumple con lo establecido en el requerimiento R-016, escalabilidad en versionamiento de gestores, debido a que permite agregar aplicaciones de nuevos gestores de bases de datos a la tabla, con su versión

correspondiente. Adicionalmente, el modelo propuesto toma en cuenta las siguientes consideraciones:

- Permite consultar las aplicaciones instaladas en un servidor, además de los gestores de bases de datos y las plataformas.
- Limita la asociación de una única plataforma por servidor.
- Permite registrar múltiples instancias de bases de datos en un servidor.
- Permite consultar aplicaciones con base en una categoría (clase de aplicación).
- Restringe en la tabla aplicaciones los campos de Nombre y Versión para que sean únicos.
- Restringe en la tabla clasesAplicaciones el campo Nombre para que sea único.
- En la tabla aplicaciones, toda aplicación debe estar registrada bajo una clase de aplicación.

4.5.2.2. Diseño D002. Administración de bases de datos y esquemas por instancia

4.5.2.2.1. Especificación del diseño

Este diseño permite integrar los objetos de bases de datos contenidos en múltiples servidores, plataformas e instancias. A su vez, permite indicar las bases de datos que se deben incluir o excluir en el proceso de integración.

Este diseño contempla los siguientes requerimientos:

- R-001. Identificar los objetos de diferentes bases de datos de manera integrada.
- R-010. Definir las bases de datos a las cuales se les debe realizar la extracción de datos.
- R-011. Definir objetos participantes.

4.5.2.2.2. Propuesta del diseño

4.5.2.2.2.1. Modelo conceptual

En la figura 18 se muestra el diseño conceptual, en la Tabla 13 se detallan las entidades y en la Tabla 14 se detallan las relaciones.

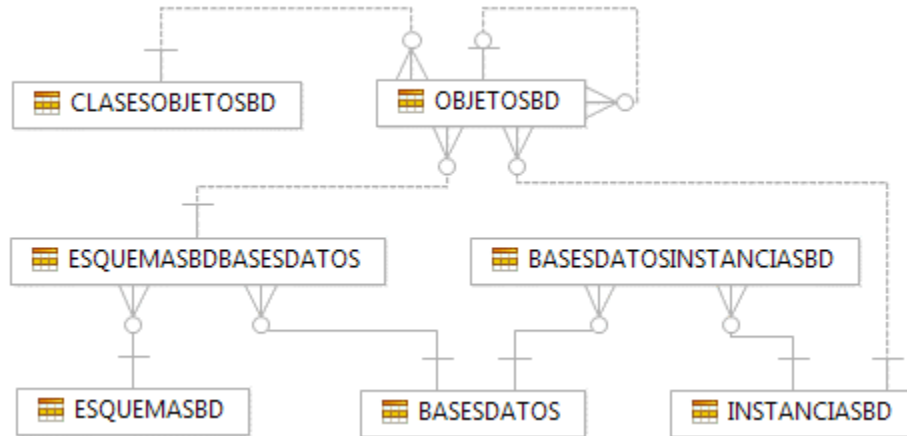


Figura 18. Modelo conceptual del diseño D002. Administración de bases de datos y esquemas por instancia.

Tabla 13. Descripción de las entidades del modelo lógico para el diseño D002. Administración de bases de datos y esquemas por instancia.

Descripción de las entidades		
ID	Entidad	Descripción
1	basesDatos	Bases de datos administradas por un gestor de base de datos. Por ejemplo, las bases de datos <i>Master</i> y <i>Model</i> son administradas por un gestor de base de datos <i>SQL Server</i> .
2	esquemasBd	Esquemas de bases de datos. Por ejemplo, en <i>SQL Server</i> , la base de datos <i>Master</i> contiene el esquema del sistema denominado <i>sys</i> , que posee las vistas del sistema, tales como tablas, funciones, etc.

3	esquemasBdBasesD atos	Esquemas por bases de datos.
4	basesDatosInstancias Bd	Bases de datos por instancia.
5	objetosBd	Son los objetos que se encuentran registrados en los esquemas de bases de datos. Por ejemplo, tablas y funciones.
6	clasesObjetosBd	Son las clases de objetos que utilizan para la integración de estos. Por ejemplo, el objeto tabla se encuentra catalogado en el gestor de base de datos <i>SQL Server</i> con el nombre “ <i>Table</i> ”; sin embargo, para la integración de los objetos se quiere utilizar el nombre de tabla en su lugar.

**Tabla 14. Descripción de las relaciones entre entidades para el diseño D002.
Administración de bases de datos y esquemas por instancia.**

Descripción de las relaciones			
ID	Entidad	Cardinalidad	Descripción
1	objetosBd- instanciasBd	1:N	Una instancia se utiliza para hacer referencia a cero, uno o más objetos.
2	basesDatosInstancia sBd- instanciasBd	1:N	Una instancia es utilizada para administrar cero, una o más bases de datos.
3	basesDatosInstancia sBd- basesDatos	1:N	Una base de datos es administrada por cero, una o más instancias de bases de datos.
4	esquemasBdBasesD atos - basesDatos	1:N	Una base de datos contiene cero, uno o más esquemas.

5	esquemasBdBasesD atos-esquemasBd	1:N	Un esquema pertenece a cero, una o más bases de datos.
6	objetosBd - esquemasBdBasesD atos	1:N	Un esquema correspondiente a una base de datos se utiliza para hacer referencia a cero, uno o más objetos.
7	objetosBd- clasesObjetosBd	1:N	Una clase de objeto de base de datos se utiliza para identificar cero, uno o más objetos.
8	objetosBd – objetosBd	1:N	Un objeto puede ser parte de cero, uno o más objetos.

4.5.2.2.2. Modelo lógico

En la Tabla 15 se detalla el diseño del modelo lógico:

Tabla 15. Modelo de datos lógico para el diseño D002. Administración de bases de datos y esquemas por instancia.

Entidad:	basesDatos		
Descripción de atributos			
Atributo	Clave primaria	Clave foránea	Descripción
codigoBaseDatos	X		Identificador de la base de datos.
Nombre			Ídem.
Descripción			Ídem.
incluirAnálisis			Indica si se deben analizar los objetos contenidos en la base de datos en los procesos de integración de objetos. Los posibles valores para este campo son “S” (Sí) y “N” (No).

Restricciones		
Descripción	Tipo	Descripción
basesDeDatos_inc luirAnálisis_chk	Comprobación	El campo incluirAnálisis admite los valores “S” (Sí) o “N” (No).
Comentarios adicionales		
Ninguno.		

Entidad:	esquemasBd		
Descripción de atributos			
Atributo	Clave primaria	Clave foránea	Descripción
codigoEsquemaBd	X		Identificador del esquema.
Nombre			Ídem.
Descripción			Ídem.
Restricciones			
Descripción	Tipo	Descripción	
N/A	N/A	N/A	
Comentarios adicionales			
Ninguno.			

Entidad:	esquemasBdBasesDatos		
Descripción de atributos			
Atributo	Clave primaria	Clave foránea	Descripción
codigoBaseDatos	X	X	Identificador de base de datos.
codigoEsquemaBd	X	X	Identificador de esquema.

Restricciones		
Descripción	Tipo	Descripción
esquemasBdBases Datos_basesDatos	Clave foránea	El campo codigoBaseDatos debe hacer referencia al campo codigoBaseDatos de la tabla basesDatos.
esquemasBdBases Datos_esquemasB d	Clave foránea	El campo codigoEsquemaBd debe hacer referencia al campo codigoEsquemaBd de la tabla esquemasBd.
Comentarios adicionales		
Ninguno.		

Entidad:	basesDatosInstanciasBd		
Descripción de atributos			
Atributo	Clave primaria	Clave foránea	Descripción
codigoInstanciaBd	X	X	Identificador de instancia.
codigoBaseDatos	X	X	Identificador de base de datos.
Restricciones			
Descripción	Tipo	Descripción	
basesDatosInstanciasBd_instanciasBd	Clave foránea	El campo codigoInstanciaBd debe hacer referencia al campo codigoInstanciaBd de la tabla instanciasBd.	
basesDatosInstanciasBd_basesDatos	Clave foránea	El campo codigoBaseDatos debe hacer referencia al campo codigoBaseDatos de la tabla basesDatos.	
Comentarios adicionales			
Ninguno.			

Entidad:	objetosBd		
Descripción de atributos			
Atributo	Clave primaria	Clave foránea	Descripción
codigoObjetoBd	X		Identificador del objeto.
codigoInstanciaBd		X	Identificador de la instancia.
codigoBaseDatos		X	Identificador de la base de datos.
codigoEsquemaBd		X	Identificador del esquema.
codigoClaseObjetoBd		X	Identificador de la clase de objeto.
Nombre			Nombre del objeto.
Descripción			Descripción del objeto.
nombreEspecifico			Nombre específico en el gestor de base de datos (se utiliza para identificar funciones y procedimientos que tienen el mismo nombre, pero con diferente cantidad de parámetros o el orden de estos).
codigoObjetoBdPadre			Ídem.
Restricciones			
Descripción	Tipo	Descripción	
objetosBd_clasesObjetosBd_fk	Clave foránea	El campo codigoClaseObjetoBd debe hacer referencia al campo codigoClaseObjetoBd de la tabla clasesObjetosBd.	
objetosBd_instanciasBd_fk	Clave foránea	El campo codigoInstanciaBd debe hacer referencia al campo codigoInstanciaBd de la tabla instanciasBd.	
objetosBd_fk	Clave	El campo codigoObjetoBdPadre deben hacer	

	foránea	referencia al campo codigoObjetoBd de la tabla objetosBd.
objetosBd_esquem asBdBasesDatos _fk	Clave foránea	Los campos codigoBaseDatos y codigoEsquemaBd deben hacer referencia a los campos codigoBaseDatos y codigoEsquemaBd de la tabla esquemasBdBasesDatos.
objetosBd_uchk	Unicidad	La combinación de los campos codigoInstanciaBd, codigoBaseDatos, codigoEsquemaBd, codigoClaseObjetoBd, Nombre y nombreEspecífico debe ser único.
Comentarios adicionales		
Ninguno.		

Entidad:	clasesObjetosBd		
Descripción de atributos			
Atributo	Clave primaria	Clave foránea	Descripción
codigoClaseObjetoBd	X		Identificador de la clase de objeto de base de datos.
Nombre			Ídem.
Descripción			Ídem.
Restricciones			
Descripción	Tipo	Descripción	
clasesObjetosBd_uchk	Unicidad	El campo Nombre debe ser único.	
Comentarios adicionales			
Ninguno.			

4.5.2.2.3. Modelo físico

En la figura 19 se muestra el modelo de datos físico.

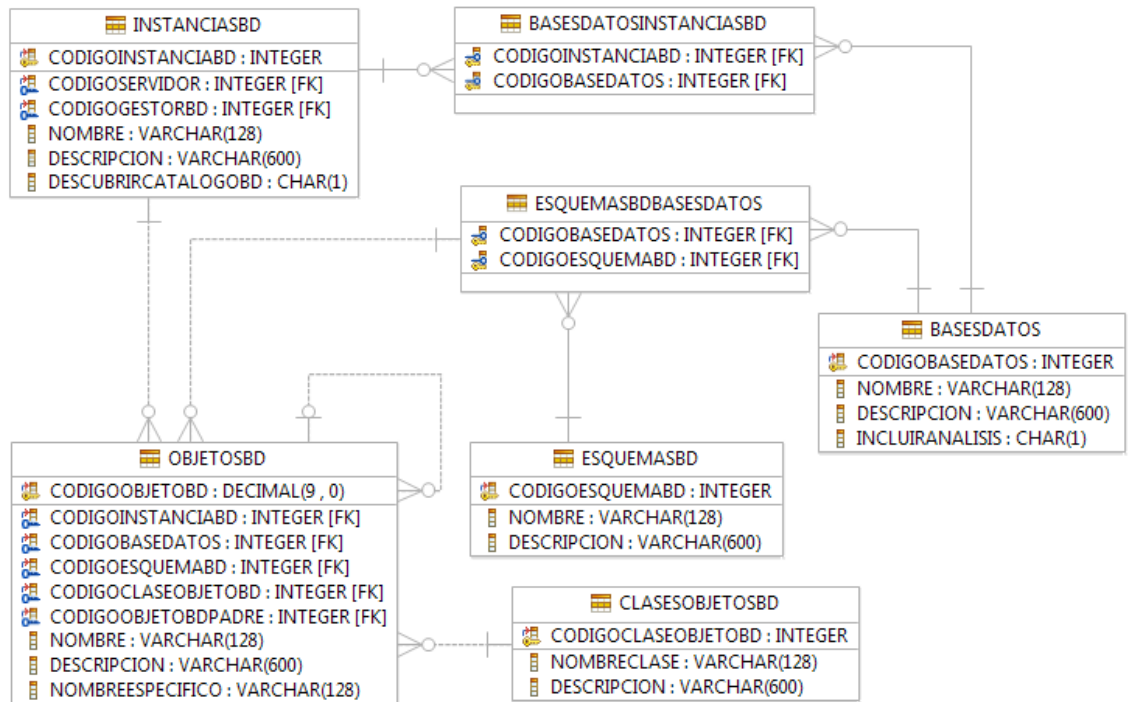


Figura 19. Modelo físico del diseño D002. Administración de bases de datos y esquemas por instancia (elaboración propia del investigador).

4.5.2.2.3. Justificación del diseño

Se parte de la arquitectura que utilizan los gestores de bases de datos DB2, Oracle y SQL Server, tal como se muestra en figura 20.

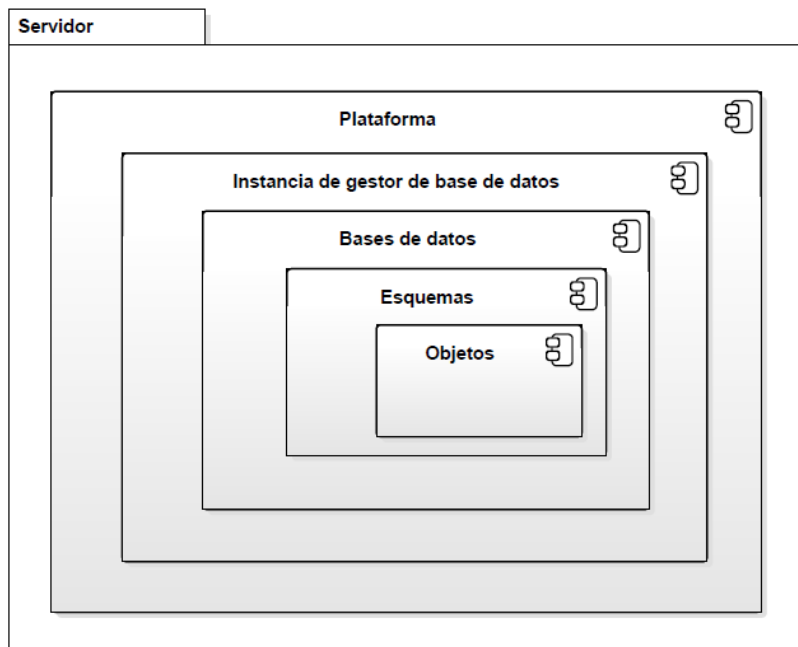


Figura 20. Arquitectura de gestores de bases de datos (elaboración propia del investigador).

Los servidores utilizan una plataforma (sistema operativo) sobre la cual se instalan instancias de gestores de bases de datos. Estas bases, a su vez, incluyen esquemas y estos abarcan a los objetos. Dada la relación de jerarquía planteada anteriormente, se propone utilizarla como una ruta que permita localizar a los objetos de forma única. Es por ello que se incluye en la tabla objetosBd una llave de unicidad conformada por los campos codigoDeInstanciaBd, CódigoDeBaseDeDatos, CódigoDeEsquemaBd y CódigoClaseDeObjetoBd. El código de clase de objeto también forma parte de la ruta, debido a que los gestores de bases de datos permiten nombrar objetos de diferente tipo, en un esquema, bajo el mismo nombre.

Se plantea la utilización de una llave sustituta que permita identificar los objetos de forma única, además de optimizar las consultas, de manera tal que se pregunte por el campo codigoObjetoBd en lugar de los cuatro campos que conforman la llave natural. El identificador único será utilizado para relacionar el objeto con las demás entidades planteadas en el diseño del modelo relacional.

El modelo cumple con lo establecido en el requerimiento R-010, Definir las bases de datos a las cuales realizar la extracción de datos, debido a que permite:

- Registrar las bases de datos correspondientes a una instancia de base de datos.
- Incluir o excluir bases de datos en el proceso de integración, por medio del campo incluirAnálisis de la tabla basesDeDatos.

El modelo cumple con lo establecido en el requerimiento R-011, Definir objetos participantes, debido a que permite definir en el campo incluirAnálisis de la tabla basesDatos, si se desea incluir o excluir la base de datos durante el proceso de extracción.

El modelo cumple con lo establecido en el requerimiento R-001, Identificar los objetos de diferentes bases de datos de manera integrada, debido a que permite registrar los objetos de bases de datos contenidos en múltiples servidores, plataformas e instancias.

Adicionalmente, el modelo propuesto toma en cuenta las siguientes consideraciones:

- Registrar bases de datos por instancia.
- Registrar esquemas por bases de datos.
- Identificar el servidor y el gestor de base de datos que utiliza una instancia.

4.5.2.3. Diseño D003. Procesos por instancia

4.5.2.3.1. Especificación del diseño

Este diseño permite registrar procesos por instancias de bases de datos y facilita dar trazabilidad a las acciones definidas en cada uno de los procesos. El diseño contempla los siguientes requerimientos:

- R-008. Consultar información sobre los servidores remotos vinculados.

- R-009. Registrar trazabilidad de los procesos del sistema.
- R-016. Escalabilidad en versionamiento de gestores.
- R-017. Escalabilidad.

4.5.2.3.2. Propuesta del diseño

4.5.2.3.2.1. Modelo conceptual

En la figura 21 se muestra el diseño conceptual, en la Tabla 16 se detallan las entidades y en la Tabla 17 se detallan las relaciones.

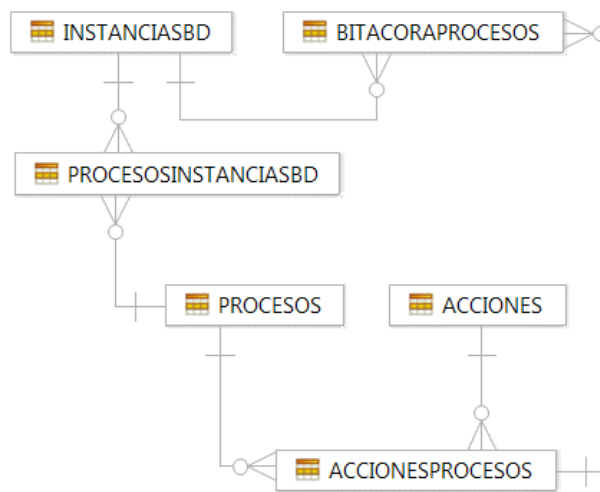


Figura 21. Modelo conceptual del diseño D003. Procesos por instancia (elaboración propia del investigador).

Tabla 16. Descripción de las entidades del modelo lógico para el diseño D003. Procesos por instancia.

Descripción de las entidades		
ID	Entidad	Descripción
1	Procesos	Definiciones de procesos por medio de un nombre único y su descripción. Por ejemplo: se registra un proceso con el nombre "ExtraccionDB2", con la descripción "Extrae los datos del diccionario del sistema de DB2".

2	Acciones	Definiciones de acciones por medio de un nombre y su descripción. Por ejemplo se registra una acción con el nombre "extraerTablaDB2", y con la descripción "Extrae los datos de tablas en DB2"; este, a su vez, invoca a un procedimiento almacenado llamado "usp_extraerTablaDB2", que se encuentra bajo el esquema determinado en la base de datos.
3	accionesProcesos	Define la relación entre procesos y acciones, tomando en cuenta la prioridad (orden) en que estas últimas deben ser ejecutadas. Por ejemplo, el proceso "ExtracionDB2" tiene relacionadas las acciones "usp_extraerTablaDB2" con prioridad 1 y "usp_extraerProcedimientoDB2" con prioridad 2.
4	bitacoraProcesos	Bitácora para almacenar el resultado de la ejecución de las acciones relacionadas con un proceso.
5	procesosInstanciasBd	Define la relación de los procesos a ser ejecutados por instancias de bases de datos, tomando en cuenta la prioridad (orden) en que estos deben ser ejecutados. Por ejemplo: una instancia de <i>Oracle</i> puede ejecutar dos procesos nombrados "ExtraccionOracle" e "IntegraciónOracle", con prioridades 1 y 2 respectivamente.

Tabla 17. Descripción de las relaciones entre entidades para el diseño D003. Procesos por instancia.

Descripción de las relaciones			
ID	Entidad	Cardinalidad	Descripción
1	procesosInstanciasBd-instanciasBd	1:N	Una instancia de base de datos se utiliza cero, una o más veces para relacionarla con un proceso.

2	procesosInstanciasBd-procesos	1:N	Un proceso se utiliza cero, una o más veces para relacionarlo con una instancia de base de datos.
3	accionesProcesos-procesos	1:N	Un proceso utiliza cero, una o más acciones.
4	accionesProcesos-acciones	1:N	Una acción se utiliza por cero, uno o más procesos.
5	bitacoraProcesos-accionesProcesos	1:N	Una bitácora de procesos utiliza cero, una o más acciones por proceso.
6	bitacoraProcesos-instanciasBd	1:N	Una bitácora de procesos utiliza cero, uno o más instancias de bases de datos.

4.5.2.3.2.2. Modelo lógico

En la Tabla 18 se detalla el diseño del modelo lógico:

Tabla 18. Modelo de datos lógico para el diseño D003. Procesos por instancia.

Entidad:	procesosInstanciasBd		
Descripción de atributos			
Atributo	Clave primaria	Clave foránea	Descripción
codigoProceso	X	X	Identificador del proceso.
codigoInstanciaBd	X	X	Identificador de la instancia de base de datos.
Prioridad	X		Indica la prioridad en la cual se deben ejecutar los procesos relacionados con la instancia de base de datos.
Restricciones			
Descripción	Tipo	Descripción	

procesosInstanciasBd_instanciasBd_fk	Clave foránea	El campo codigoInstanciaBd debe hacer referencia al campo codigoInstanciaBd de la tabla instanciasBd.
procesosInstanciasBd_procesos_fk	Clave foránea	El campo codigoProceso debe hacer referencia al campo codigoProceso de la tabla procesos.
Comentarios adicionales		
Ninguno.		

Entidad:	procesos		
Descripción de atributos			
Atributo	Clave primaria	Clave foránea	Descripción
codigoProceso	X		Identificador del proceso.
Nombre			Nombre del proceso.
Descripción			Descripción del proceso.
Restricciones			
Descripción	Tipo	Descripción	
procesos_uchk	Unicidad	El campo nombre debe ser único.	
Comentarios adicionales			
Ninguno.			

Entidad:	Acciones		
Descripción de atributos			
Atributo	Clave primaria	Clave foránea	Descripción
codigoAccion	X		Identificador de la acción.
aplicacionBd			Aplicación de base de datos invocada por la acción. Entiéndase por aplicación a los procedimientos

			almacenados y funciones contenidas en los gestores de bases de datos.
esquemaAplicacionBd			Esquema de base de datos en el cual reside la aplicación.
Nombre			Nombre de la acción.
Descripción			Descripción de la acción.
Restricciones			
Descripción	Tipo	Descripción	
acciones01_uchk	Unicidad	El campo Nombre debe ser único.	
acciones02_uchk	Unicidad	La combinación de los campos aplicacionBd y esquemaAplicacionBd debe ser única.	
Comentarios adicionales			
Se utilizan dos restricciones para evitar acciones con el mismo nombre y evitar acciones con diferentes nombres que invoquen a los mismos programas.			

Entidad:	accionesProcesos		
Descripción de atributos			
Atributo	Clave primaria	Clave foránea	Descripción
codigoAccion	X	X	Identificador de la acción.
codigoProceso	X	X	Identificador del proceso.
Prioridad	X		Indica la prioridad en la cual se deben ejecutar las acciones relacionadas con los procesos.
Restricciones			
Descripción	Tipo	Descripción	
accionesProcesos_procesos_fk	Clave foránea	El campo codigoProceso debe hacer referencia al campo codigoProceso de la tabla procesos.	
accionesProcesos_acciones_fk	Clave foránea	El campo codigoAccion debe hacer referencia al campo codigoAccion de la tabla acciones.	

Comentarios adicionales
Ninguno.

Entidad:	bitacoraProcesos		
Descripción de atributos			
Atributo	Clave primaria	Clave foránea	Descripción
codigoInstanciaBd	X	X	Identificador de la instancia de base de datos.
codigoProceso	X	X	Identificador del proceso.
codigoAccion	X	X	Identificador de la acción.
Prioridad	X	X	Prioridad de la acción.
inicioEjecucion	X		Fecha y hora en que inició la ejecución del proceso.
finalizacionEjecucion			Fecha y hora en que finalizó la ejecución del proceso.
estadoEjecución			Estado de ejecución, los posibles valores para este campo son "P" (Procesado) y "E" (Error).
mensajeError			Mensaje descriptivo del error ocurrido en una acción.
Restricciones			
Descripción	Tipo	Descripción	
bitacoraProcesos_accionesProcesos_fk	Clave foránea	Los campos codigoAccion, codigoProceso y Prioridad deben hacer referencia a los campos codigoAccion, codigoProceso y Prioridad de la tabla accionesProcesos.	
bitacoraProcesos_instanciasBd_fk	Clave foránea	El campo codigoInstanciaBd debe hacer referencia al campo codigoInstanciaBd de la tabla instanciasBd.	

bitacoraProcesos_chk	Comprobación	El campo estadoEjecucion debe permitir únicamente el ingreso de los valores “P” (Procesado) y “E” (Error).
Comentarios adicionales		
Ninguno.		

4.5.2.3.2.3. Modelo físico

En la figura 22 se muestra el modelo de datos físico.

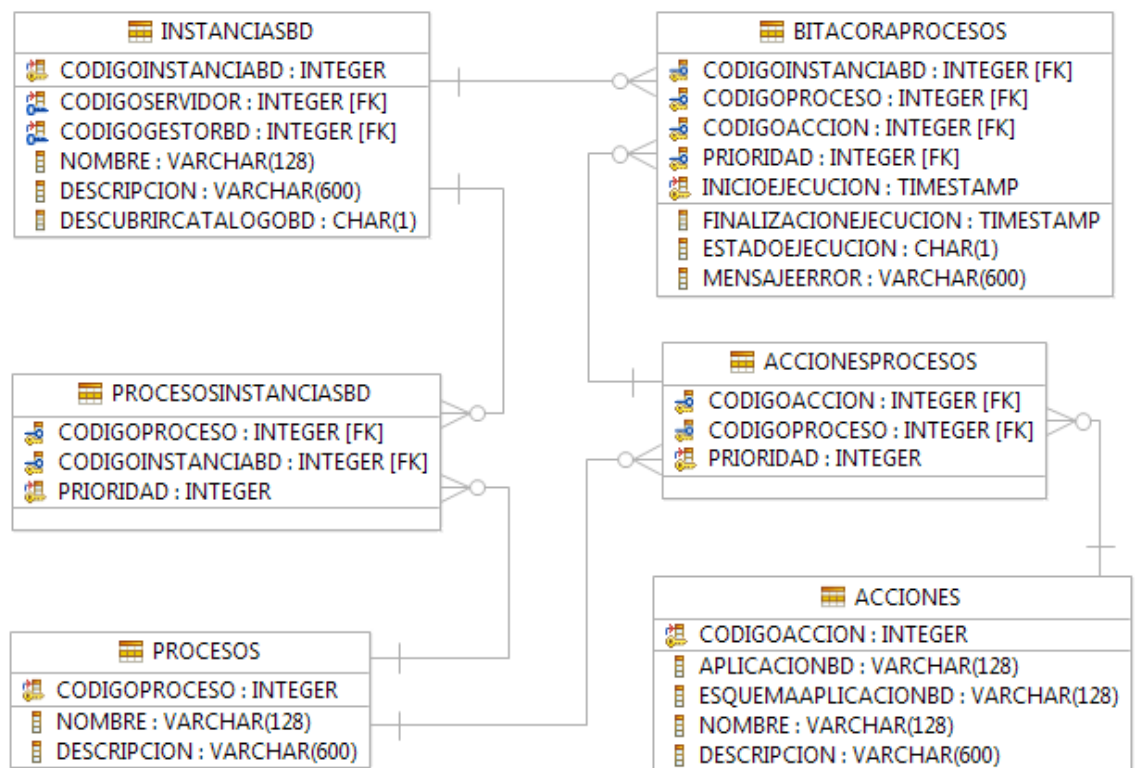


Figura 22.- Modelo físico del diseño D003. Procesos por instancia (elaboración propia del investigador).

4.5.2.3.3. Justificación del diseño

El modelo cumple con lo establecido en el requerimiento R-008, Consultar información sobre los servidores remotos vinculados, debido a que permite:

- Consultar las instancias de bases de datos registrados por servidor.
- Consultar el gestor de base de datos relacionado con una instancia específica.

El modelo cumple con lo establecido en el requerimiento R-009, Registrar trazabilidad de los procesos del sistema, debido a que permite:

- Almacenar en la tabla bitacoraProcesos datos sobre el estado y mensajes de error resultantes de la ejecución de cada una de las acciones realizadas para un proceso en específico, relacionado con una instancia de base de datos.

El modelo cumple con lo establecido en el requerimiento R-016, Escalabilidad en versionamiento de gestores, debido a que permite:

- Registrar en la tabla procesosInstanciasBd procesos reutilizables por instancias de bases de datos. Por ejemplo: se crea un proceso nombrado “ExtraccionDB2”, el cual funciona tanto para DB2 en versión 9.7 como en versión 10.5, solo se requiere parametrizar el proceso de ambas instancias de bases de datos, lo cual implica un cambio mínimo en la configuración.

El modelo cumple con lo establecido en el requerimiento R-017, Escalabilidad, debido a que permite:

- Registrar procesos de forma tal que son reutilizables por múltiples instancias de bases de datos.
- Registrar acciones que son reutilizables por múltiples procesos.
- Registrar en la tabla accionesProcesos un conjunto de acciones que son ejecutadas por un proceso, en una secuencia de prioridad.
- Registrar en la tabla procesosInstanciasBd un conjunto de procesos que son ejecutados por una instancia de base de datos, en una secuencia de prioridad.
- Registrar por acción la aplicación que será invocada desde un esquema de base de datos en específico. Esto permite que el

sistema pueda crecer en funcionalidad de manera modular, ya que, por ejemplo, si se desea agregar una nueva acción en un proceso existente, solo es necesario insertarla en la tabla accionesProcesos. Este cambio se ve reflejado inmediatamente para todas las instancias que consuman el proceso modificado.

Adicionalmente, el modelo propuesto toma en cuenta las siguientes consideraciones:

- El nombre de un proceso es único.
- El nombre de una acción es único.

No se permite el uso de acciones nombradas de diferente manera que ejecuten los mismos programas en un esquema de base de datos, para evitar errores en la entrada de registros por parte de los usuarios.

4.5.2.4. Diseño D004. Registro de colaboradores

4.5.2.4.1. Especificación del diseño

Este diseño permite registrar los colaboradores de la empresa, los cuales pertenecen a un departamento y desempeñan un puesto laboral.

La tabla de colaboradores permite identificar el detalle de las personas propietarias de los objetos de base de datos. Por ejemplo, en una instancia de base de datos DB2 existe una tabla nombrada TAF001, la propietaria es Adriana, quien labora en el Departamento de Operaciones y desempeña el cargo de Supervisora; adicionalmente tiene un número de teléfono para ser contactada en caso de ser necesario, para solicitar permisos con la finalidad de modificar datos en los computadores de producción.

Este diseño contempla los siguientes requerimientos:

- R-005. Asociar los objetos con propietarios.

4.5.2.4.2. Propuesta del diseño

4.5.2.4.2.1. Modelo conceptual

En la figura 23 se muestra el diseño conceptual, en la Tabla 19 se detallan las entidades y en la Tabla 20 se detallan las relaciones.

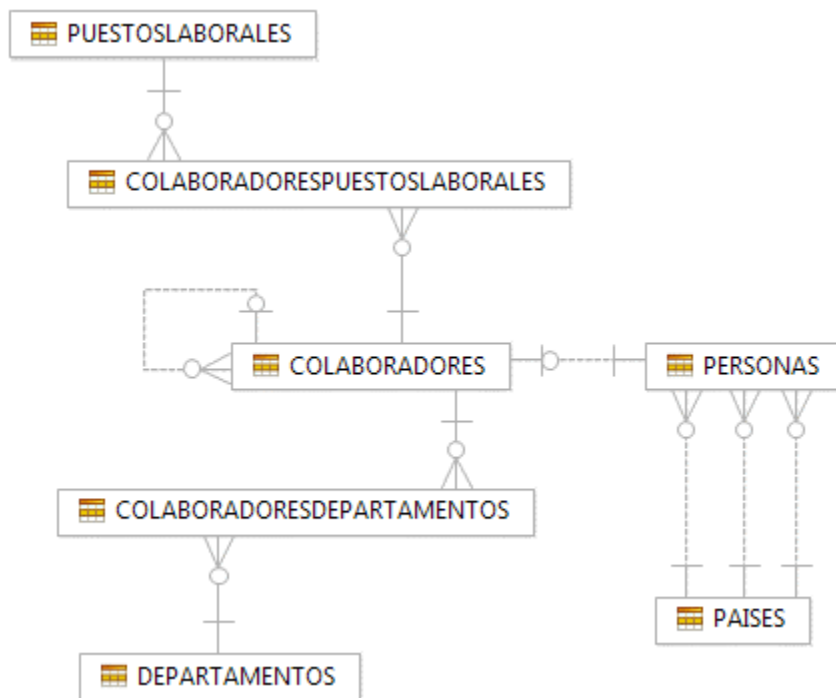


Figura 23. Modelo conceptual del diseño D004. Registro de colaboradores.

Tabla 19. Descripción de las entidades del modelo lógico para el diseño D004. Registro de colaboradores.

Descripción de las entidades		
ID	Entidad	Descripción
1	personas	Individuo de la especie humana.
2	colaboradores	Personas que laboran en la organización.

3	departamentos	Departamentos de la organización. Por ejemplo: Contabilidad, Finanzas, Sistemas.
4	puestosLaborales	Titulo otorgado a un colaborador con base en la función que desempeña en la organización.
5	países	Países del mundo
6	colaboradoresPuestos Laborales	Colaboradores por puesto laboral.
7	colaboradoresDepartamentos	Colaboradores por departamento.

Tabla 20. Descripción de las relaciones entre entidades para el diseño D004. Registro de colaboradores.

Descripción de las relaciones			
ID	Entidad	Cardinalidad	Descripción
1	personas - países	1:N	Los países se utilizan de cero, una o más veces para identificar el país de nacionalidad, residencia y nacimiento de las personas.
2	colaboradores - personas	1:1	Una persona es registrada de cero a una vez como colaborador.
3	colaboradores - colaboradores	1:N	Un colaborador es jefe de cero, uno o más colaboradores.
4	colaboradoresDepartamentos -	1:N	Los colaboradores utilizan cero, uno o más departamentos.

	colaboradores		
5	colaboradoresDepartamentos-departamentos	1:N	Un departamento es utilizado por cero, uno o más colaboradores.
6	colaboradoresPuestosLaborales-colaboradores	1:N	Los colaboradores utilizan cero, uno o más puestos laborales.
7	colaboradoresPuestosLaborales - puestosLaborales	1:N	Un puesto laboral es utilizado por cero, uno o más colaboradores.

4.5.2.4.2.2. Modelo lógico

En la Tabla 21 se detalla el diseño del modelo lógico:

Tabla 21. Modelo de datos lógico para el diseño D004. Registro de colaboradores.

Entidad:	personas		
Descripción de atributos			
Atributo	Clave primaria	Clave foránea	Descripción
codigoPersona	X		Identificador de la persona.
IdentificacionLegal			Identificación legal de la persona, como cédula de identidad, pasaporte o visa.
Nombre			Ídem.
primerApellido			Ídem.
segundoApellido			Ídem.
fechaNacimiento			Ídem.
Edad			Ídem.
Genero			Ídem.

estadoCivil			Ídem.
codigoPaisNacionalidad		X	Identificador del país.
codigoPaisNacimiento		X	Identificador del país.
codigoPaisResidencia		X	Identificador del país.
direccionDomicilio			Identificador del país.
Restricciones			
Descripción	Tipo	Descripción	
personas_uchk	Unicidad	El campo IdentificacionLegal debe ser único.	
personas_genero_chk	Comprobación	El campo “género” admite los valores “M” (Masculino) o “F” (Femenino).	
personas_estadoCivil_chk	Comprobación	El campo estadoCivil admite los valores “S” (Soltero), “C” (Casado), “D” (Divorciado) y “U” (Unión libre).	
personas_paises01_fk	Clave foránea	El campo codigoPaisNacionalidad debe hacer referencia al campo codigoPais de la tabla países.	
personas_paises02_fk	Clave foránea	El campo codigoPaisNacimiento debe hacer referencia al campo codigoPais de la tabla países.	
personas_paises03_fk	Clave foránea	El campo codigoPaisResidencia debe hacer referencia al campo codigoPais de la tabla países.	
Comentarios adicionales			
Ninguno.			

Entidad:	colaboradores		
Descripción de atributos			
Atributo	Clave primaria	Clave foránea	Descripción

codigoColaborador	X		Identificador del colaborador.
codigoJefatura		X	Identificador del colaborador jefe.
codigoPersona		X	Identificador de la persona.
fechaContratacion			Ídem.
fechaDespido			Ídem.
habilitado			Indica si el funcionario se encuentra laborando actualmente. Admite los valores "S" (Sí) o "N" (No).
numerotelefono			Número de teléfono, asignado al colaborador.
correoElectronico			Correo electrónico del colaborador.
Restricciones			
Descripción	Tipo	Descripción	
colaboradores_personas_fk	clave foránea	El campo codigoPersona debe hacer referencia al campo codigoPersona de la tabla personas.	
colaboradores_codigoJefatura_fk	clave foránea	El campo codigoJefatura debe hacer referencia al campo codigoColaborador de la tabla colaboradores.	
colaboradores_unicidad	unicidad	El campo codigoPersona debe ser único.	
colaboradores_habilitado_chk	comprobación	El campo habilitado admite los valores "S" (Sí) o "N" (No).	
Comentarios adicionales			
Ninguno.			

Entidad:	departamentos		
Descripción de atributos			
Atributo	Clave primaria	Clave foránea	Descripción
codigoDepartamento	X		Identificador del departamento.
Nombre			Ídem.
Descripción			Ídem.
Restricciones			
Descripción	Tipo	Descripción	
departamentos_uchk	Unicidad	El campo nombre debe ser único.	
Comentarios adicionales			
Ninguno.			

Entidad:	puestosLaborales		
Descripción de atributos			
Atributo	Clave primaria	Clave foránea	Descripción
codigoPuestoLaboral	X		Identificador del puesto laboral.
Nombre			Ídem.
Descripción			Ídem.
Restricciones			
Descripción	Tipo	Descripción	
puestosLaborales_uchk	Unicidad	El campo Nombre debe ser único.	
Comentarios adicionales			
Ninguno.			

Entidad:	paises		
Descripción de atributos			
Atributo	Clave primaria	Clave foránea	Descripción
codigoPais	X		Identificador del país.
codigoPaisAlfa2			Identificador del país alfanumérico de dos caracteres.
codigoPaisAlfa3			Identificador del país alfanumérico de tres caracteres.
Nombre			Ídem.
Restricciones			
Descripción	Tipo	Descripción	
paises_uchk	Unicidad	La combinación de los campos codigoPaisAlfa2 y codigoPaisAlfa3 debe ser única.	
Comentarios adicionales			
Ninguno.			

Entidad:	colaboradoresDepartamentos		
Descripción de atributos			
Atributo	Clave primaria	Clave foránea	Descripción
codigoColaborador	X	X	Identificador del colaborador.
codigoDepartamento	X	X	Identificador del departamento.
Restricciones			
Descripción	Tipo	Descripción	
colaboradoresDep	Clave	El campo codigoColaborador debe hacer	

artamentos_colaboradores_fk	foránea	referencia al campo codigoColaborador de la tabla colaboradores.
colaboradoresDepartamentos_departamentos_fk	Clave foránea	El campo codigoDepartamento debe hacer referencia al campo codigoDepartamento de la tabla departamentos.
Comentarios adicionales		
Ninguno.		

Entidad:	colaboradoresPuestosLaborales		
Descripción de atributos			
Atributo	Clave primaria	Clave foránea	Descripción
codigoColaborador	X	X	Identificador del colaborador.
codigoPuestoLaboral	X	X	Identificador del puesto laboral.
Restricciones			
Descripción	Tipo	Descripción	
colaboradoresPuestosLaborales_colaboradores_fk	Clave foránea	El campo codigoColaborador debe hacer referencia al campo codigoColaborador de la tabla colaboradores.	
colaboradoresPuestosLaborales_puestosLaborales_fk	Clave foránea	El campo codigoPuestoLaboral debe hacer referencia al campo codigoPuestoLaboral de la tabla puestosLaborales.	
Comentarios adicionales			
Ninguno.			

4.5.2.4.2.3. Modelo físico

En la figura 24 se muestra el modelo de datos físico.

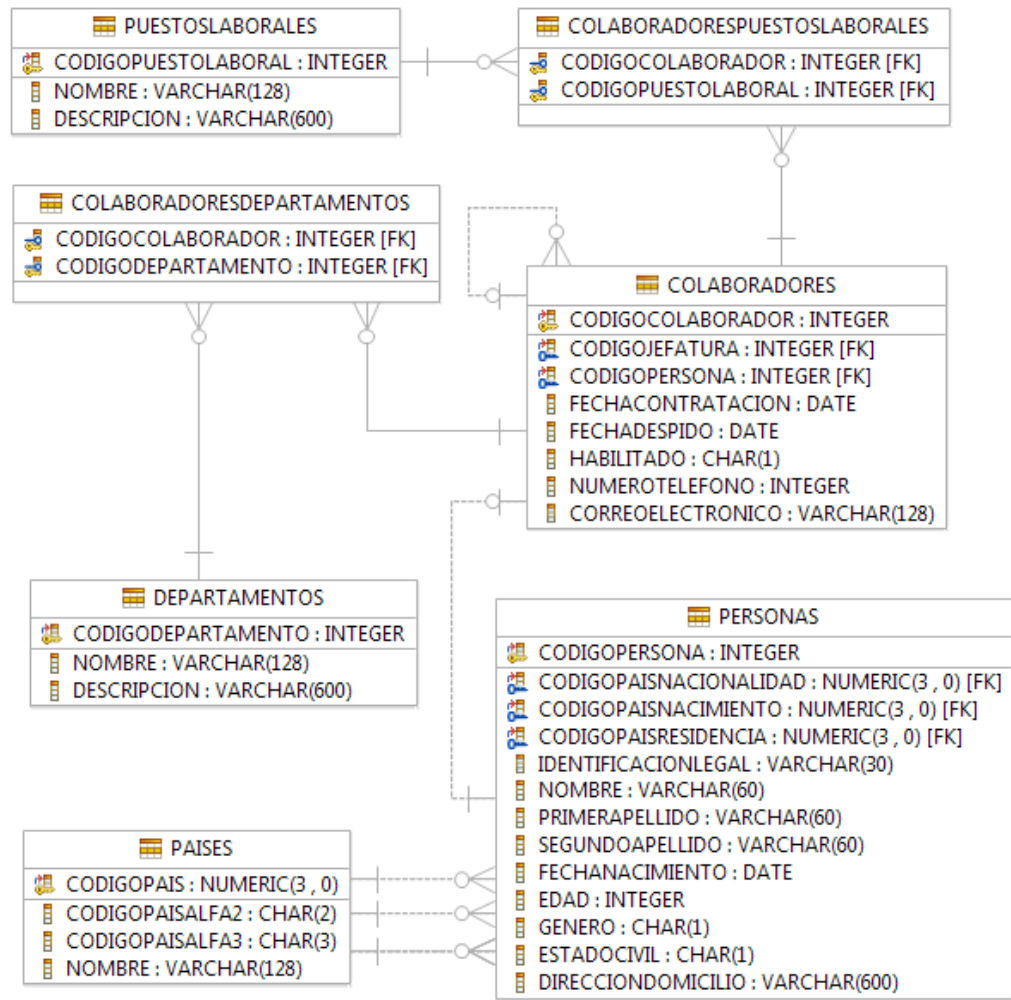


Figura 24. Modelo físico del diseño D004. Registro de colaboradores (elaboración propia del investigador).

4.5.2.4.3. Justificación del diseño

El modelo cumple con lo establecido en el requerimiento R-005, Asociar los objetos con propietarios, debido a que permite consultar el detalle de la persona propietaria de los objetos de bases de datos.

4.5.2.5. Diseño D005. Propietarios por objeto

4.5.2.5.1. Especificación del diseño

Este diseño permite asignar objetos de bases de datos a sus propietarios, los cuales pueden ser titulares o delegados.

Los propietarios titulares son aquellos colaboradores autorizados para otorgar permisos para realizar cambios sobre los objetos. Por ejemplo, un propietario titular puede solicitar la eliminación de datos en tablas.

Los propietarios delegados son colaboradores autorizados por los propietarios titulares para administrar objetos bajo su cargo; sin embargo, no pueden otorgar permisos sobre estos. Por ejemplo, un delegado puede realizar comentarios sobre los objetos para que se clasifiquen como sensibles, pero no puede otorgar permisos para eliminar datos.

Un ejemplo práctico de la utilización de este modelo se presenta cuando se requiere modificar datos debido a errores en la manipulación de estos, dependiendo de las políticas de la compañía, los cambios de datos en los ambientes de producción son restringidos y deben ser verificados por un comité de cambios, el cual requiere identificar quién es la persona propietaria del objeto, para autorizar si un cambio aplica o no.

Este diseño contempla los siguientes requerimientos:

- R-005. Asociar los objetos con propietarios.
- R-013. Consultar objetos por propietarios.

4.5.2.5.2. Propuesta del diseño

4.5.2.5.2.1. Modelo conceptual

En la figura 25 se muestra el diseño conceptual, en la Tabla 22 se detallan las entidades y en la Tabla 23 se detallan las relaciones.

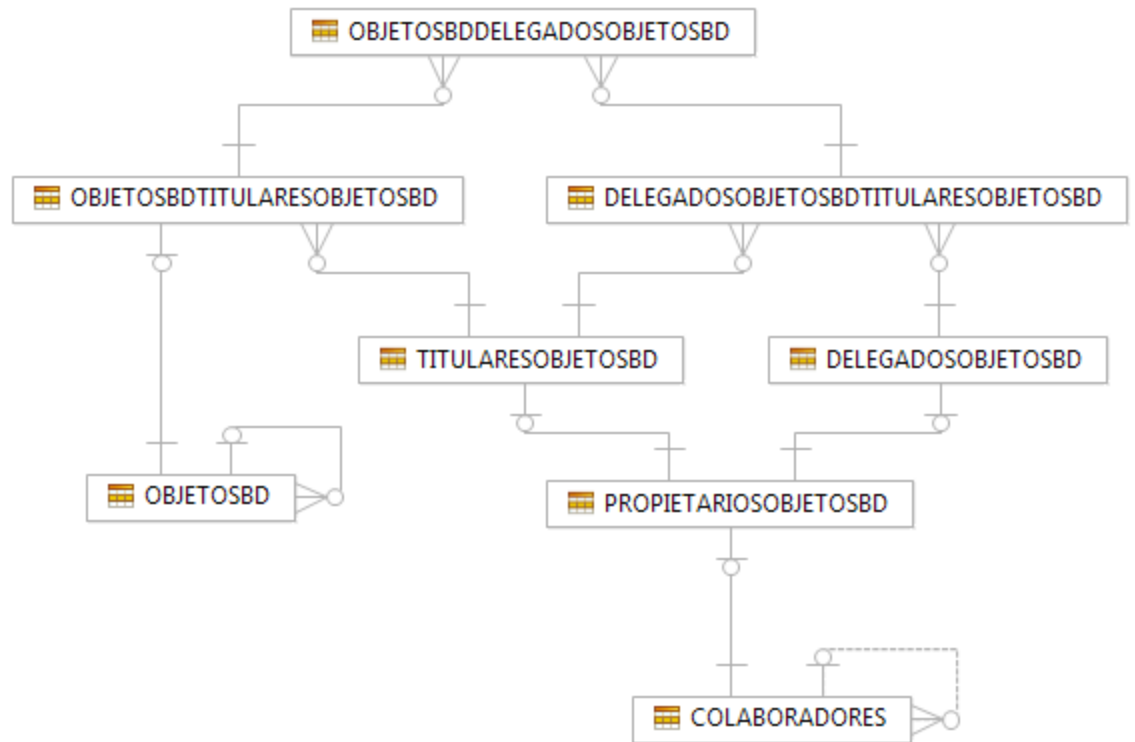


Figura 25. Modelo conceptual del diseño D005. Propietarios por objeto.

Tabla 22. Descripción de las entidades del modelo lógico para el diseño D005.

Propietarios por objeto.

Descripción de las entidades		
ID	Entidad	Descripción
1	propietariosObjetosBd	Colaboradores que custodian los objetos contenidos en los gestores de bases de datos.
2	titularesObjetosBd	Colaboradores autorizados para otorgar permisos para realizar cambios sobre los objetos de bases de datos. Por ejemplo: un propietario titular puede solicitar la eliminación de datos en tablas.
3	delegadosObjetosBd	Colaboradores autorizados por los propietarios titulares para administrar los objetos de bases de datos; sin embargo, no pueden otorgar permisos sobre estos. Por ejemplo, un delegado puede realizar comentarios sobre

		los objetos para que se clasifiquen como sensibles, pero no puede otorgar permisos para cambiar datos o atributos.
4	delegadosObjetosBd TitularesObjetosBd	Colaboradores delegados por un titular para administrar los objetos bajo su cargo.
5	objetosBdTitularesO bjetosBd	Objetos de base de datos asignados a los propietarios titulares.
6	objetosBdDelegados ObjetosBd	Objetos de base de datos asignados a los propietarios delegados.

Tabla 23. Descripción de las relaciones entre entidades para el diseño D005-Propietarios por objeto.

Descripción de las relaciones			
ID	Entidad	Cardinalidad	Descripción
1	objetosTitularesObj etosBd-objetosBd	1:1	Un objeto de base de datos está bajo el cargo de cero a un titular.
2	propietariosObjetos Bd-colaboradores	1:1	Un colaborador se registra como un propietario de objetos de cero a una vez.
3	titularesObjetosBd- propietariosObjetos Bd	1:1	Un propietario de objetos se registra como titular de cero a una vez.
4	delegadosObjetosBd - propietariosObjetos Bd	1:1	Un propietario de objetos se registra como delegado de cero a una vez.
5	objetosTitularesObj etosBd- titularesObjetosBd	1:N	Un propietario titular puede tener a su cargo cero, uno o más objetos.

6	delegadosObjetosBd TitularesObjetosBd- titularesObjetosBd	1:N	Un propietario titular puede tener a su cargo cero, uno o más delegados.
7	delegadosObjetosBd TitularesObjetosBd- delegadosObjetosBd	1:N	Un propietario delegado puede ser delegado de cero, uno o más propietarios titulares.
8	objetosDelegadosOb jetosBd- objetosTitularesObj etosBd	1:N	Los objetos de los titulares son administrados por cero, uno o más delegados.
9	objetosDelegadosOb jetosBd- delegadosObjetosBd TitularesObjetosBd	1:N	Un delegado puede tener a su cargo cero, uno o más objetos de diferentes titulares.

4.5.2.5.2.2. Modelo lógico

En la Tabla 24 se detalla el diseño del modelo lógico:

Tabla 24. Modelo de datos lógico para el diseño D005. Propietarios por objeto.

Entidad:	propietariosObjetosBd		
Descripción de atributos			
Atributo	Clave primaria	Clave foránea	Descripción
codigoPropietario ObjetoBd	X	X	Identificador del propietario de objetos en base de datos.
Restricciones			
Descripción	Tipo	Descripción	
propietariosObjeto sBd_colaboradore	Clave foránea	El campo codigoPropietarioObjetoBd debe hacer referencia al campo codigoColaborador de la tabla	

s_fk		colaboradores.
Comentarios adicionales		
Ninguno.		

Entidad:	titularesObjetosBd		
Descripción de atributos			
Atributo	Clave primaria	Clave foránea	Descripción
codigoTitularObjetoBd	X	X	Identificador del propietario titular de objetos en base de datos.
Restricciones			
Descripción	Tipo	Descripción	
titularesObjetosBd_propietariosObjetosBd_FK	Clave foránea	El campo codigoTitularObjetoBd debe hacer referencia al campo codigoPropietarioObjetoBd de la tabla propietariosObjetosBd.	
Comentarios adicionales			
Ninguno.			

Entidad:	delegadosObjetosBd		
Descripción de atributos			
Atributo	Clave primaria	Clave foránea	Descripción
codigoDelegadoObjetoBd	X	X	Identificador del propietario delegado de objetos en base de datos.
Restricciones			
Descripción	Tipo	Descripción	
delegadosObjetosBd_propietariosObjetosBd_fk	Clave foránea	El campo codigoDelegadoObjetoBd debe hacer referencia al campo codigoPropietarioObjetoBd de la tabla propietariosObjetosBd.	

Comentarios adicionales	
Ninguno.	

Entidad:		delegadosObjetosBdTitularesObjetosBd	
Descripción de atributos			
Atributo	Clave primaria	Clave foránea	Descripción
codigoDelegadoObjetoBd	X	X	Identificador del propietario delegado de objetos en base de datos.
codigoTitularObjetoBd	X	X	Identificador del propietario titular de objetos en base de datos.
Restricciones			
Descripción	Tipo	Descripción	
delegadosObjetosBdTitularesObjetosBd_titularesObjetosBd_fk	Clave foránea	El campo codigoTitularObjetoBd debe hacer referencia al campo codigoTitularObjetoBd de la tabla titularesObjetosBd.	
delegadosObjetosBdTitularesObjetosBd_delegadosObjetosBd_fk	Clave foránea	El campo codigoDelegadoObjetoBd debe hacer referencia al campo codigoDelegadoObjetoBd de la tabla delegadosObjetosBd.	
Comentarios adicionales			
Ninguno.			

Entidad:		objetosBdTitularesObjetosBd	
Descripción de atributos			
Atributo	Clave primaria	Clave foránea	Descripción
codigoTitularObjetoBd	X	X	Identificador del propietario titular de objetos en base de datos.

codigoObjetoBd	X	X	Identificador del objeto de base de datos.
Restricciones			
Descripción	Tipo	Descripción	
objetosTitularesObjetosBd_titularesObjetosBd_fk	Clave foránea	El campo codigoTitularObjetoBd debe hacer referencia al campo codigotitularObjetoBd en la tabla titularesObjetosBd.	
objetosTitularesObjetosBd_objetosBd_fk	Clave foránea	El campo codigoObjetoBd debe hacer referencia al campo codigoObjetoBd en la tabla titularesObjetosBd.	
objetosBdTitularesObjetosBd_codigoObjetoBd_uchk	Unicidad	El campo codigoObjetoBd debe ser único. Este se utiliza para que un objeto solo esté registrado por un propietario titular.	
Comentarios adicionales			
Ninguno.			

Entidad:	objetosBdDelegadosObjetosBd		
Descripción de atributos			
Atributo	Clave primaria	Clave foránea	Descripción
codigoDelegadoObjetoBd	X	X	Identificador del propietario delegado de objetos en base de datos.
codigoTitularObjetoBd	X	X	Identificador del propietario titular de objetos en base de datos.
codigoObjetoBd	X	X	Identificador del objeto de base de datos.
Restricciones			
Descripción	Tipo	Descripción	
objetosDelegadosObjetosBd_delega	Clave foránea	Los campos codigoDelegadoObjetobd y codigoTitularObjetoBd deben hacer referencia a	

dosObjetosBdTitularesObjetosBd_fk		los campos codigoDelegadoObjetobd y codigoTitularObjetoBd en la tabla delegadosObjetosBdTitularesObjetosBd.
objetosDelegadosObjetosBd_objetosTitularesObjetosBd_fk	Clave foránea	Los campos codigoTitularObjetoBd y codigoObjetoBd deben hacer referencia a los campos codigoTitularObjetoBd y codigoObjetoBd en la tabla objetosBdTitularesObjetosBd.
Comentarios adicionales		
Ninguno.		

4.5.2.5.2.3. Modelo físico

En la figura 26 se muestra el modelo de datos físico.

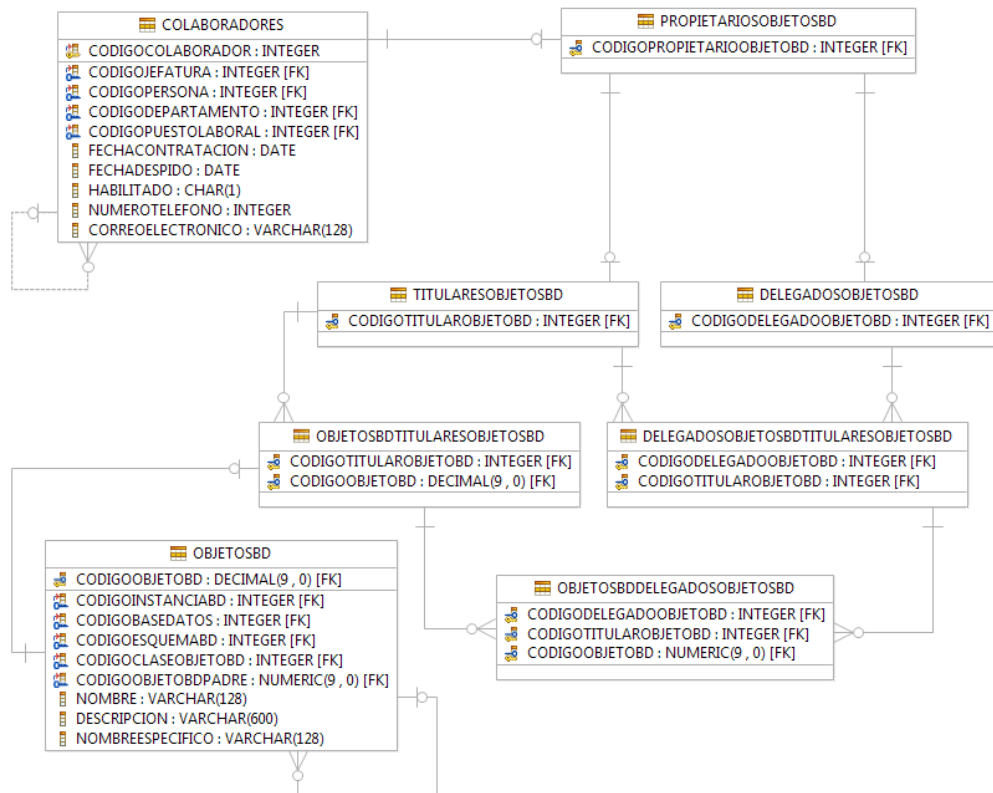


Figura 26. Modelo físico del diseño D005. Propietarios por objeto (elaboración propia del investigador).

4.5.2.5.3. Justificación del diseño

En este diseño se hace uso del concepto de herencia expuesto por González Alvarado (2009). Se parte de una generalización de la tabla colaboradores y se realiza una especialización en la tabla propietariosObjetosBd, que a su vez se especializa en las entidades titularesObjetosBd y delegadosObjetosBd, con la finalidad de crear relaciones entre los propietarios titulares y los propietarios delegados en la tabla delegadosObjetosBdTitularesObjetosBd.

El modelo cumple con lo establecido en el requerimiento R-013, Consultar objetos por propietarios, debido a que permite:

- Consultar los objetos asignados a un propietario titular.
- Consultar los objetos asignados a un propietario delegado.

El modelo cumple con lo establecido en el requerimiento R-005, Asociar los objetos con propietarios, debido a que permite:

- Limitar que un objeto de base de datos se asocie únicamente a un propietario titular, lo cual se logra por medio de la restricción de unicidad sobre el campo códigoObjetoBd en la tabla objetosBdTitularesObjetosBd.
- Un propietario puede tener bajo su responsabilidad más de un objeto.
- Un propietario puede tener uno o más delegados.

El modelo limita de forma tal que los propietarios delegados solo puedan tener relacionados objetos que correspondan a los propietarios titulares que los delegaron. Para lograr esta característica se emplea la combinación de las llaves foráneas sobre la tabla objetosDelegadosObjetosBd, como se muestra figura 27, de forma tal que el objeto tenga que estar registrado como un objeto de un titular en la tabla objetosBdTitularesObjetosBd, y la relación entre el propietario el titular esté definida en la tabla delegadosObjetosBdTitularesObjetosBd.

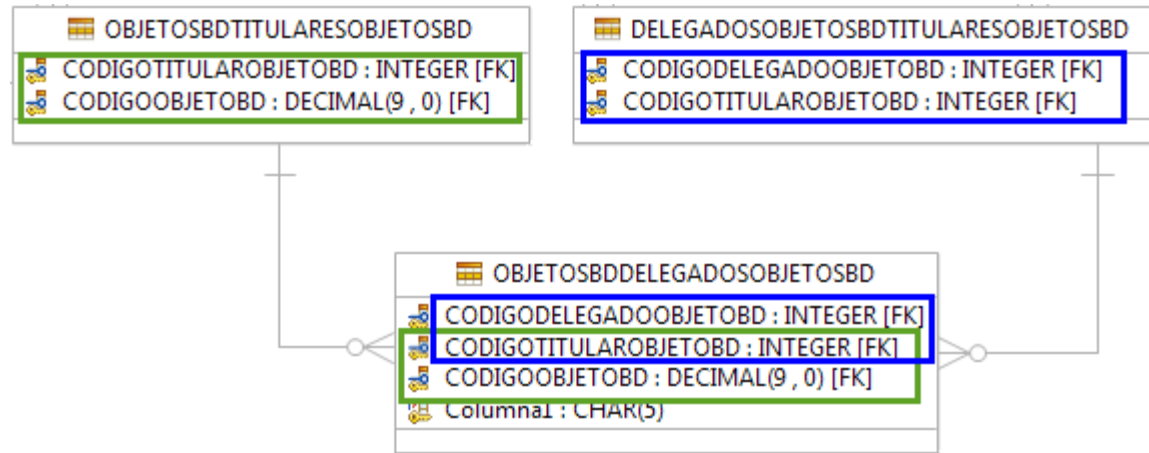


Figura 27. Restricción sobre los objetos de los delegados.

Adicionalmente, el modelo propuesto toma en cuenta las siguientes consideraciones:

- Permite consultar los propietarios delegados por titular.
- Permite consultar los propietarios titulares a los cuales corresponde un delegado.

4.5.2.6. Diseño D006. Comentarios por propietario

4.5.2.6.1. Especificación del diseño

Este diseño permite a los propietarios de los objetos de bases de datos, realizar comentarios sobre estos, para que posteriormente un aprobador realice la revisión correspondiente, incluya sus observaciones y cambie el estado del comentario; por ejemplo, de pendiente a aprobado.

Por medio de los comentarios, los aprobadores realizarán cambios en el diccionario de datos empresarial, de esta forma se promueve el trabajo colaborativo entre las áreas de T.I. y negocio. Por ejemplo, un usuario puede comentar que un campo es sensible, para que posteriormente sea revisado por personal de seguridad y lo tomen en consideración.

Este diseño contempla los siguientes requerimientos:

- R-007. Asociar los objetos con observaciones de sus propietarios.
- R-015. Consultar las observaciones de los objetos.

4.5.2.6.2. Propuesta del diseño

4.5.2.6.2.1. Modelo conceptual

En la figura 28 se muestra el diseño conceptual, en la Tabla 25 se detallan las entidades y en la Tabla 26 se detallan las relaciones.

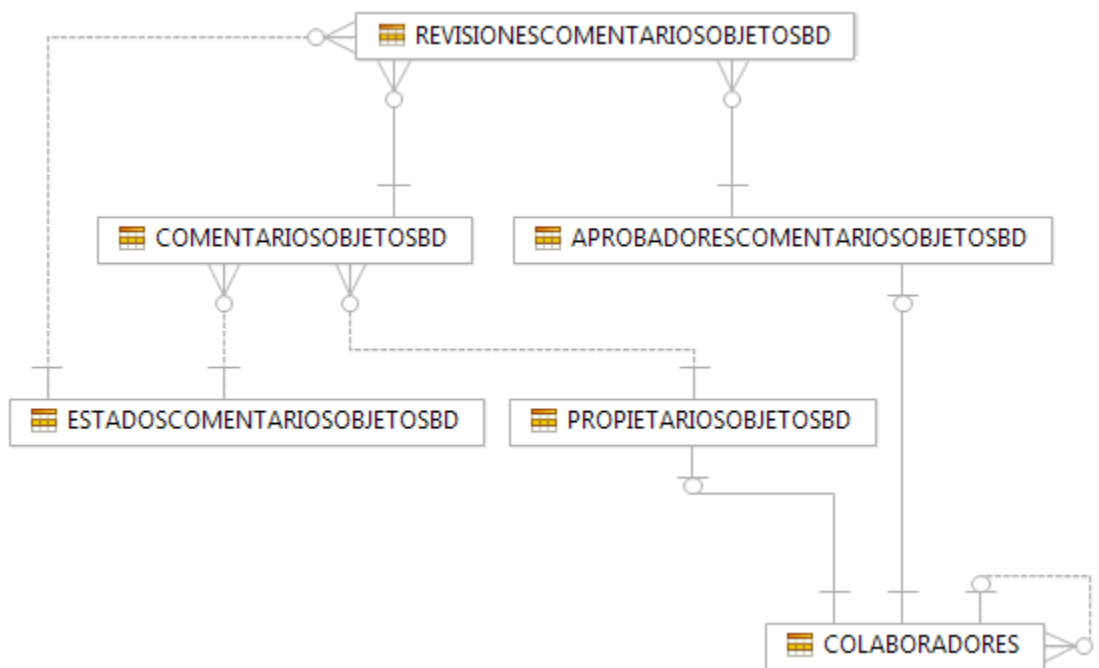


Figura 28. Modelo conceptual del diseño D006. Comentarios por propietario.

Tabla 25. Descripción de las entidades del modelo lógico para el diseño D006. Comentarios por propietario.

Descripción de las entidades		
ID	Entidad	Descripción

1	aprobadoresComentariosObjetosBd	Colaboradores autorizados para aprobar o denegar los comentarios realizados por los propietarios de los objetos.
2	comentariosObjetosBd	Comentarios realizados a los objetos que están a cargo de los propietarios de objetos de base de datos (pueden ser tanto titulares como delegados).
3	estadosComentariosObjetosBd	Estados sobre los comentarios realizados por los propietarios de los objetos de bases de datos. Por ejemplo, un comentario puede tener el estado de aprobado o denegado.
4	revisionesComentariosObjetosBd	Observaciones de los aprobadores de comentarios sobre los comentarios realizados por los propietarios de los objetos. Por ejemplo, José, que es un propietario, ingresa un comentario sobre la tabla clientes, solicitando que sea clasificada como sensible; Jorge, quien es el aprobador del comentario, ingresa una respuesta en la cual detalla que fue incluido.

Tabla 26. Descripción de las relaciones entre entidades para el diseño D006. Comentarios por propietario.

Descripción de las relaciones			
ID	Entidad	Cardinalidad	Descripción
1	codigoAprobadorComentarioObjetoBd-colaboradores	1:1	Un colaborador puede estar registrado de cero a una vez como aprobador de comentarios sobre objetos de base de datos.
2	comentariosObjetosBd-propietariosObjetosBd	1:N	Un propietario realiza cero, uno o más comentarios sobre los objetos a su cargo.

3	revisionesComentariosObjetosBd-aprobadoresComentariosObjetosBd	1:N	Un aprobador de comentarios sobre los objetos de bases de datos realiza cero, una o más revisiones.
4	revisionesComentariosObjetosBd-comentariosObjetosBd	1:N	Un comentario realizado a un objeto de base de datos puede tener cero, una o más revisiones.
5	comentariosObjetosBd-estadosComentariosObjetosBd	1:N	Los estados de los comentarios son utilizados cero, una o más veces en los comentarios de los objetos.
6	revisionesComentariosObjetosBd-estadosComentariosObjetosBd	1:N	Los estados de los comentarios son utilizados cero, una o más veces en las revisiones de los comentarios.

4.5.2.6.2.2. Modelo lógico

En la Tabla 27 se detalla el diseño del modelo lógico:

Tabla 27. Modelo de datos lógico para el diseño D006. Comentarios por propietario.

Entidad:	aprobadoresComentariosObjetosBd		
Descripción de atributos			
Atributo	Clave primaria	Clave foránea	Descripción
codigoAprobador ComentarioObjeto Bd	X	X	Identificador del aprobador de los comentarios sobre los objetos de bases de datos.
Habilitado			Indica si el aprobador de comentarios se encuentra habilitado. Los posibles

			valores para este campo son “S” (Sí) y “N” (No).
Restricciones			
Descripción	Tipo	Descripción	
aprobadorescomentariosObjetosBd_habilitado_chk	Comprobación	El campo habilitado admite los valores “N” (No) y “S” (Sí).	
codigoAprobadorComentarioObjetoBd_colaboradores_fk	Clave foránea	El campo codigoAprobadorComentarioObjetoBd debe hacer referencia al campo codigoColaborador en la tabla colaboradores.	
Comentarios adicionales			
Ninguno.			

Entidad:	comentariosObjetosBd		
Descripción de atributos			
Atributo	Clave primaria	Clave foránea	Descripción
codigoComentarioObjetoBd	X		Identificador del comentario.
codigoObjetoBd		X	Identificador del objeto de bases de datos.
codigoPropietarioObjetoBd		X	Identificador del propietario del objeto.
codigoEstadoComentarioObjetoBd		X	Identificador del estado del comentario.
comentario			Comentario realizado por los propietarios de los objetos.
fechaInclusion			Fecha de inclusión del comentario.
fechaModificacio			Fecha de modificación del registro del

n			comentario. Por ejemplo, cuando se aprueba, cambia el estado de pendiente de revisar a aprobado.
Restricciones			
Descripción	Tipo	Descripción	
comentariosObjetosBd_objetosBd_fk	clave foránea	El campo codigoObjetoBd debe hacer referencia al campo codigoObjetoBd en la tabla objetosBd.	
comentariosObjetosBd_propietariosObjetosBd_fk	clave foránea	El campo codigoPropietarioObjetoBd debe hacer referencia al campo codigoPropietarioObjetoBd en la tabla propietariosObjetosBd.	
comentariosObjetosBd_estadosComentariosObjetosBd_fk	clave foránea	El campo codigoEstadoComentarioObjetoBd debe hacer referencia al campo codigoEstadoComentarioObjetoBd en la tabla estadosComentariosObjetosBd.	
Comentarios adicionales			
Ninguno.			

Entidad:	estadosComentariosObjetosBd		
Descripción de atributos			
Atributo	Clave primaria	Clave foránea	Descripción
codigoEstadoComentarioObjetoBd	X		Identificador del estado del comentario.
nombre			Nombre del comentario, por ejemplo: “Aprobado”, “Rechazado”, “En revisión”.
Descripción			Ídem
Restricciones			
Descripción	Tipo	Descripción	

estadosComentariosObjetosBd_nombre_uchk	Unicidad	El campo Nombre debe ser único.
Comentarios adicionales		
Ninguno.		

Entidad:	revisionesComentariosObjetosBd		
Descripción de atributos			
Atributo	Clave primaria	Clave foránea	Descripción
codigoAprobadorComentarioObjetoBd	X	X	Identificador del aprobador de los comentarios sobre los objetos de bases de datos.
codigoComentarioObjetoBd	X	X	Identificador del comentario.
fechaModificacion	X		Fecha de modificación del comentario.
codigoEstadoComentarioObjetoBd		X	Identificador del estado del comentario.
comentario			Ídem
Restricciones			
Descripción	Tipo	Descripción	
revisionesComentariosObjetosBd_aprobadorescomentariosObjetosBd_fk	Clave foránea	El campo codigoAprobadorComentarioObjetoBd debe hacer referencia al campo codigoAprobadorComentarioObjetoBd en la tabla aprobadoresComentariosObjetosBd.	
revisionesComentariosObjetosBd_comentariosObjeto	Clave foránea	El campo codigoComentarioObjetoBd debe hacer referencia al campo codigoComentarioObjetoBd en la tabla comentariosObjetosBd.	

sBd_fk		
revisionesComentariosObjetosBd_estadosComentariosObjetosBd_fk	Clave foránea	El campo codigoEstadoComentarioObjetoBd debe hacer referencia al campo codigoEstadoComentarioObjetoBd en la tabla estadosComentariosObjetosBd.
Comentarios adicionales		
Ninguno.		

4.5.2.6.2.3. Modelo físico

En la figura 29 se muestra el modelo de datos físico.

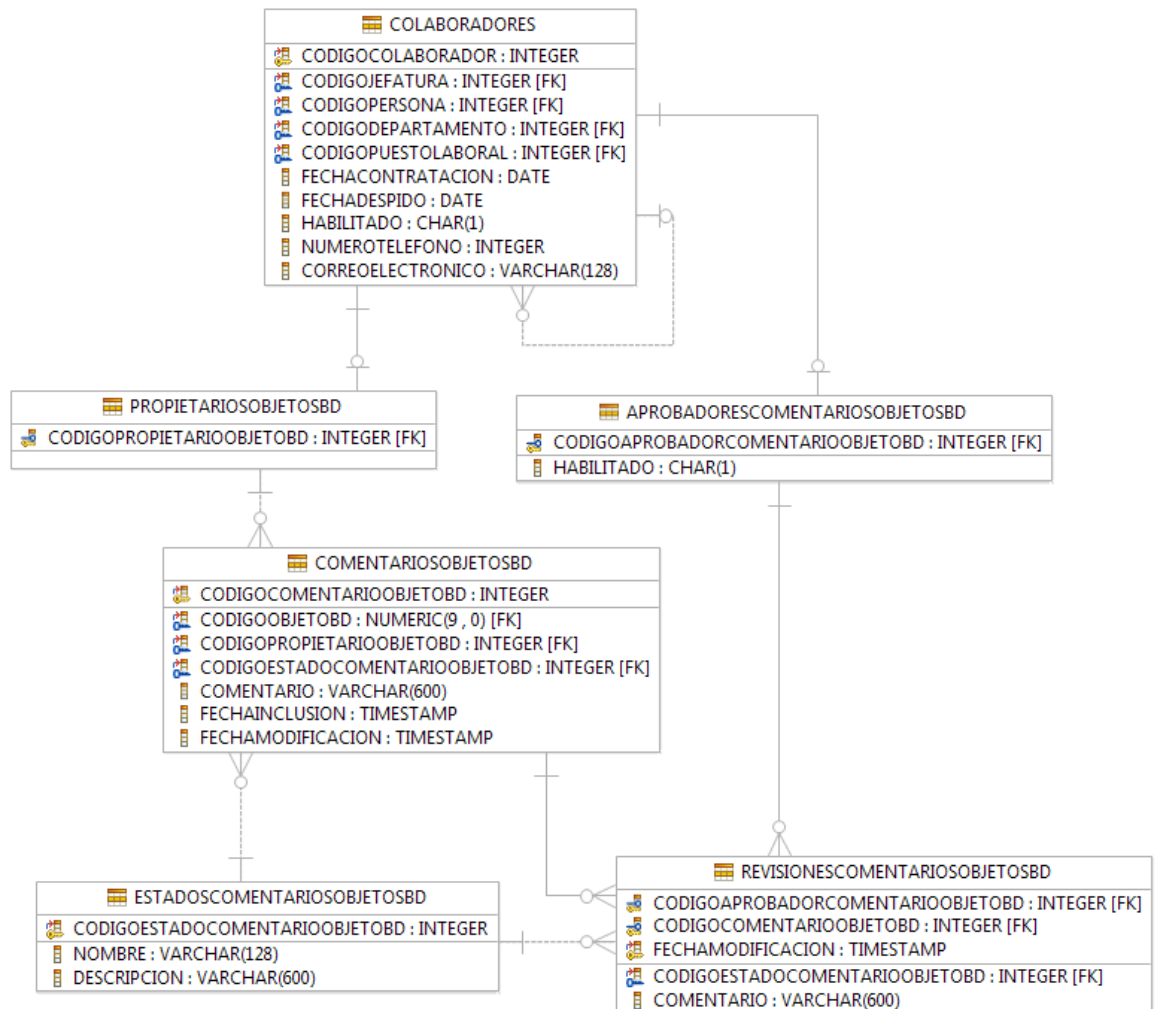


Figura 29. Modelo físico del diseño D006. Comentarios por propietario (elaboración propia del investigador).

4.5.2.6.3. Justificación del diseño

El modelo cumple con lo establecido en el requerimiento R-007, Asociar los objetos con observaciones de sus propietarios, debido a que permite incluir las observaciones realizadas por los propietarios.

El modelo cumple con lo establecido en el requerimiento R-015, Consultar las observaciones de los objetos, debido a que permite consultar las observaciones realizadas por los propietarios. Adicionalmente, el modelo propuesto toma en cuenta las siguientes consideraciones:

- Permite que un aprobador incluya sus observaciones en los comentarios realizados por los propietarios en la tabla revisionesComentariosObjetosBd.

4.5.2.7. Diseño D007. Conceptos por objeto

4.5.2.7.1. Especificación del diseño

Este diseño permite administrar los objetos de bases de datos con base en los conceptos registrados en el diccionario de datos empresarial. Contempla los siguientes requerimientos:

- R-002. Conceptos del diccionario.
- R-003. Agrupar conceptos del diccionario en jerarquías.
- R-004. Asociar objetos con conceptos del diccionario de datos.
- R-012. Consultar objetos por conceptos del diccionario.

4.5.2.7.2. Propuesta del diseño

4.5.2.7.2.1. Modelo conceptual

En la figura 30 se muestra el diseño conceptual, en la Tabla 28 se detallan las entidades y en la Tabla 29 se detallan las relaciones.

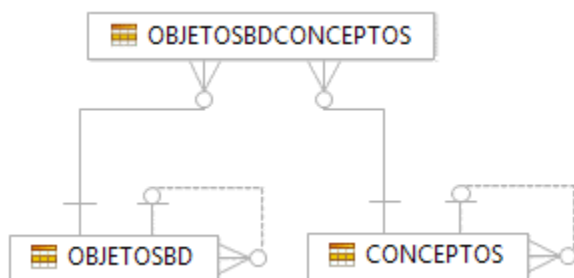


Figura 30. Modelo conceptual del diseño D007. Conceptos por objeto.

Tabla 28. Descripción de las entidades del modelo lógico para el diseño D007. Conceptos por objeto.

Descripción de las entidades		
ID	Entidad	Descripción
1	Conceptos	Conceptos del diccionario de datos empresarial.
2	objetosBdConceptos	Objetos relacionados con un concepto del diccionario de datos empresarial.

Tabla 29. Descripción de las relaciones entre entidades para el diseño D007. Conceptos por objeto.

Descripción de las relaciones			
ID	Entidad	Cardinalidad	Descripción
1	conceptos - conceptos	1:N	Un concepto puede ser padre de cero, uno o más conceptos.

2	objetosBdConceptos - conceptos	1:N	Un concepto es utilizado por cero, una o más objetos de bases de datos.
3	objetosBdConceptos - objetosBd	1:N	Un objeto de base de datos utiliza cero, uno o más conceptos.

4.5.2.7.2.2. Modelo lógico

En la tabla 30 se detalla el diseño del modelo lógico:

Tabla 30. Modelo de datos lógico para el diseño D007. Conceptos por objeto.

Entidad:			
Descripción de atributos			
Atributo	Clave primaria	Clave foránea	Descripción
codigoConcepto	X		Identificador del concepto.
codigoConceptoPadre		X	Identificador del concepto al cual pertenece el concepto (puede ser nulo).
palabraClave			Palabra utilizada para buscar conceptos.
Descripción			Ídem
Restricciones			
Descripción	Tipo	Descripción	
conceptos_conceptos_fk	Clave foránea	El campo codigoConceptoPadre debe hacer referencia al campo codigoConcepto de la tabla conceptos.	
conceptos_uchk	Unicidad	El campo palabraClave debe ser único.	
Comentarios adicionales			
Ninguno.			

Entidad:	objetosBdConceptos		
Descripción de atributos			
Atributo	Clave primaria	Clave foránea	Descripción
codigoConcepto	X	X	Identificador del concepto.
codigoObjetoBd	X	X	Identificador del objeto.
Restricciones			
Descripción	Tipo	Descripción	
objetosBdConceptos_conceptos_fk	Clave foránea	El campo codigoConcepto debe hacer referencia al campo codigoConcepto de la tabla conceptos.	
objetosBdConceptos_objetosBd_fk	Clave foránea	El campo codigoObjetoBd debe hacer referencia al campo codigoObjetoBd de la tabla objetosBd.	
Comentarios adicionales			
Ninguno.			

4.5.2.7.2.3. Modelo físico

En la figura 31 se muestra el modelo de datos físico.

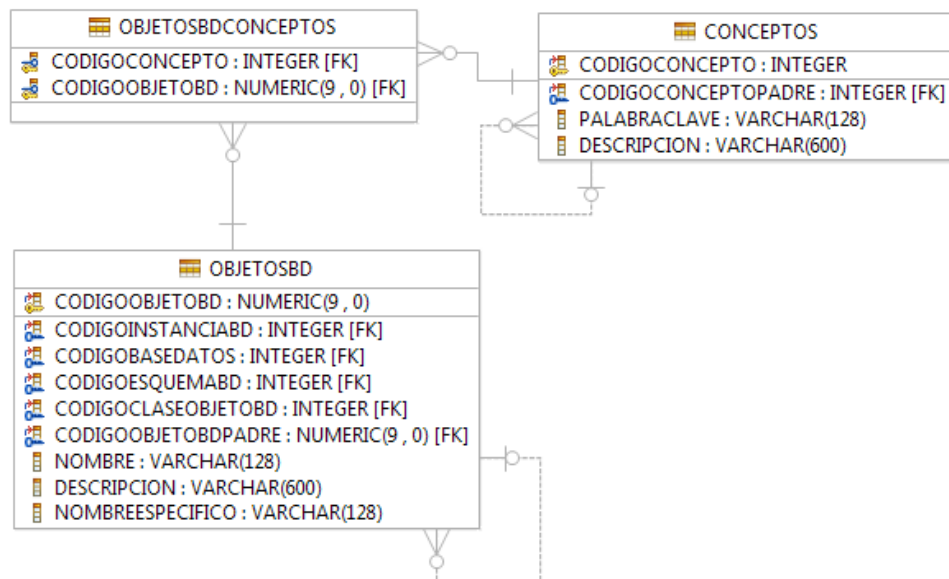


Figura 31. Modelo físico del diseño D007. Conceptos por objeto (elaboración propia del investigador).

4.5.2.7.3. Justificación del diseño

El diseño permite relacionar múltiples objetos con múltiples conceptos del diccionario empresarial. Se puede presentar el caso en el cual se asocia un concepto a múltiples objetos que se encuentran en sistemas de bases de datos de diferente tipo, con un nombre distinto, como se muestra en la figura 32. En la figura 33 se muestra cómo múltiples conceptos pueden estar relacionados con un mismo objeto.

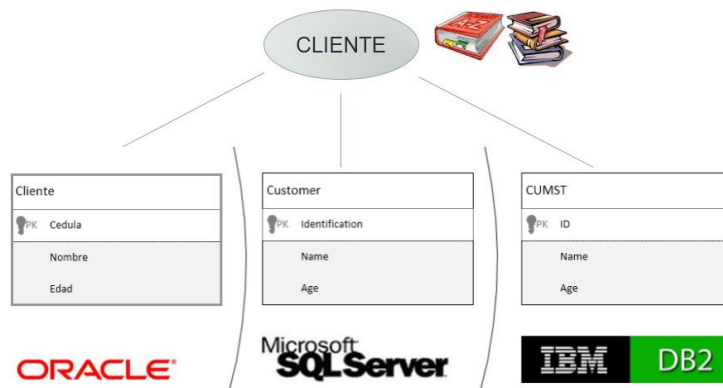


Figura 32. Relación de un concepto con múltiples objetos (elaboración propia del investigador).

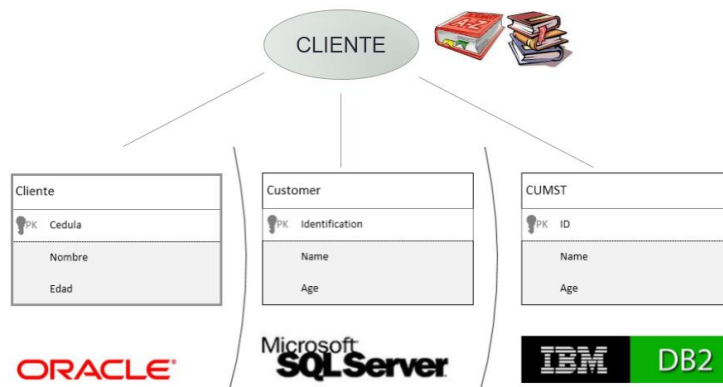


Figura 33. Relación de múltiples conceptos con un objeto (elaboración propia del investigador).

Para agrupar conceptos de forma jerárquica se incorpora una relación recursiva en la tabla conceptos, de manera tal que un concepto pueda tener asociado un padre y así establecer relaciones jerárquicas (agrupaciones). Una

ventaja de este tipo de relaciones es que permite la reorganización de los conceptos sin afectar las relaciones externas realizadas previamente a la tabla conceptos, ya que solo es requerido alterar el campo codigoConceptoPadre de dicha tabla. En la figura 34 se muestra un ejemplo de jerarquía del lado izquierdo y su reorganización del lado derecho para el nodo de seguridad.



Figura 34. Agrupación de conceptos en jerarquías (elaboración propia del investigador).

En figura 35 se muestran los cambios necesarios en los datos para llevar a cabo una reorganización del concepto seguridad.

Antes				Después			
Código concepto	Código concepto padre	Palabra clave	Descripción	Código concepto	Código concepto padre	Palabra clave	Descripción
1	NULL	Diccionario	...	1	NULL	Diccionario	...
2	1	Aplicaciones	...	2	1	Aplicaciones	...
3	2	Préstamo	...	3	2	Préstamo	...
4	2	Riesgo	...	4	2	Riesgo	...
5	2	Seguros	...	5	2	Seguros	...
6	1	Seguridad	...	6	2	Seguridad	...
7	6	Cifrado	...	7	6	Cifrado	...
8	6	Sensible	...	8	6	Sensible	...
9	6	Confidencial	...	9	6	Confidencial	...

Figura 35. Relación de múltiples conceptos con un objeto (elaboración propia del investigador).

El modelo cumple con lo establecido en el requerimiento R-002, Conceptos del diccionario, debido a que permite definir conceptos en el diccionario empresarial que serán utilizados para clasificar los objetos de bases de datos.

El modelo cumple con lo establecido en el requerimiento R-003, Agrupar conceptos del diccionario en jerarquías, debido a que permite asignar a un concepto, un concepto padre. Por medio de esta relación es posible la construcción de jerarquías de conceptos.

El modelo cumple con lo establecido en el requerimiento R-004, Asociar objetos con conceptos del diccionario de datos, debido a que permite asociar los objetos registrados en la tabla objetosBd con los conceptos registrados en la tabla conceptos, por medio de la tabla objetosBdConceptos.

El modelo cumple con lo establecido en el requerimiento R-012, Consultar objetos por conceptos del diccionario, debido a que permite consultar todos los objetos que pertenecen a un concepto determinado, sin importar el gestor de base de datos o la plataforma en la cual se encuentran almacenados.

4.5.2.8. Diseño D008. Controles por objeto

4.5.2.8.1. Especificación del diseño

Este diseño permite registrar controles que deben ser aplicados sobre los objetos de bases de datos. Los controles tienen una clasificación; por ejemplo, existen controles de seguridad y respaldos.

Un ejemplo práctico sobre su uso se presenta cuando se realizan auditorías, en las cuales se desea verificar el cumplimiento de controles establecidos a los objetos de bases de datos.

Este diseño contempla los siguientes requerimientos:

- R-006. Asociar los objetos con controles.
- R-014. Consultar objetos por controles.

4.5.2.8.2. Propuesta del diseño

4.5.2.8.2.1. Modelo conceptual

En la figura 36 se muestra el diseño conceptual, en la tabla 31 se detallan las entidades y en la Tabla 32 se detallan las relaciones.

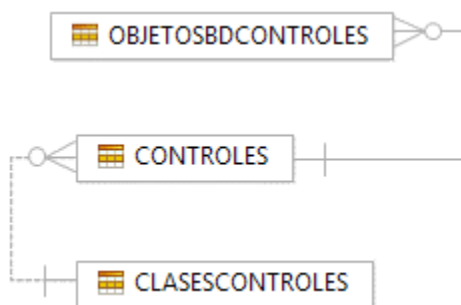


Figura 36. Modelo conceptual del diseño D008. Controles por objeto.

Tabla 31. Descripción de las entidades del modelo lógico para el diseño D008. Controles por objeto.

Descripción de las entidades		
ID	Entidad	Descripción
1	Controles	Ídem.
2	clasesControles	Clases de controles.
3	objetosBdControles	Controles asignados a los objetos de bases de datos.

Tabla 32. Descripción de las relaciones entre entidades para el diseño D008. Controles por objeto.

Descripción de las relaciones			
ID	Entidad	Cardinalidad	Descripción

1	controles - clasesControles	1:N	Una clase de control es utilizado por cero, uno o más controles.
2	objetosControles- controles	1:N	Un control es implementado en cero, uno o más objetos de bases de datos.

4.5.2.8.2.2. Modelo lógico

En la Tabla 33 se detalla el diseño del modelo lógico:

Tabla 33. Modelo de datos lógico para el diseño D008. Controles por objeto.

Entidad:	controles		
Descripción de atributos			
Atributo	Clave primaria	Clave foránea	Descripción
codigoControl	X		Identificador del control.
codigoClaseControl		X	Identificador de la clase de control.
Nombre			Ídem.
Descripción			Ídem.
Restricciones			
Descripción	Tipo	Descripción	
controles_uchk	Unicidad	El campo Nombre debe ser único.	
controles_clasesControl_fk	Clave foránea	El campo codigoClaseControl debe hacer referencia al campo codigoClaseControl de la tabla clasesControles.	
Comentarios adicionales			
Ninguno.			

Entidad:	clasesControles
Descripción de atributos	

Atributo	Clave primaria	Clave foránea	Descripción
codigoClaseControl	X		Identificador de la clase de control.
nombre			Ídem.
descripcion			Ídem.
Restricciones			
Descripción	Tipo	Descripción	
clasesControles_unicidad	Unicidad	El campo Nombre debe ser único.	
Comentarios adicionales			
Ninguno.			

Entidad:	objetosBdControles		
Descripción de atributos			
Atributo	Clave primaria	Clave foránea	Descripción
codigoControl	X	X	Identificador del control.
codigoObjetoBd	X	X	Identificador de la clase de control.
Restricciones			
Descripción	Tipo	Descripción	
objetosControles_unicidad	Clave foránea	El campo codigoControl debe hacer referencia al campo codigoControl de la tabla controles.	
objetosControles_unicidad	Clave foránea	El campo codigoObjetoBd debe hacer referencia al campo codigoObjetoBd de la tabla objetosBd.	
Comentarios adicionales			
Ninguno.			

4.5.2.8.2.3. Modelo físico

En la figura 37 se muestra el modelo de datos físico.

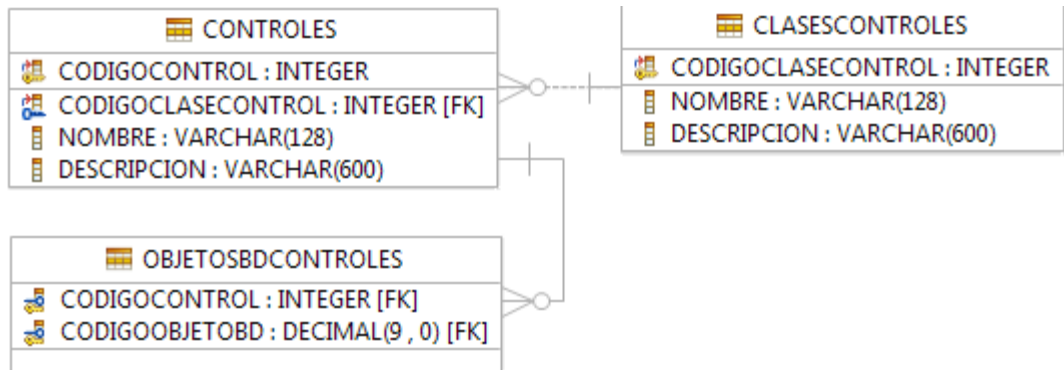


Figura 37. Modelo físico del diseño D008. Controles por objeto (elaboración propia del investigador).

4.5.2.8.3. Justificación del diseño

El modelo cumple con lo establecido en el requerimiento R-006, Asociar los objetos con controles, debido a que permite asociar múltiples controles a múltiples objetos.

El modelo cumple con lo establecido en el requerimiento R-014, Consultar objetos por controles, debido a que permite:

- Consultar todos los controles asignados a un objeto en particular.
- Consultar todos los objetos a los cuales tienen asignado un control específico.
- Consultar los controles bajo una categoría (clase de control).

4.5.2.9. Diseño D009. Servidores por ubicación geográfica

4.5.2.9.1. Especificación del diseño

Este diseño permite administrar información sobre la ubicación geográfica de los servidores, contempla el siguiente requerimiento:

- R-008. Consultar información sobre los servidores remotos vinculados.

4.5.2.9.2. Propuesta del diseño

4.5.2.9.2.1. Modelo conceptual

En la figura 38 se muestra el diseño conceptual, en la Tabla 34 se detallan las entidades y en la Tabla 35 se detallan las relaciones.

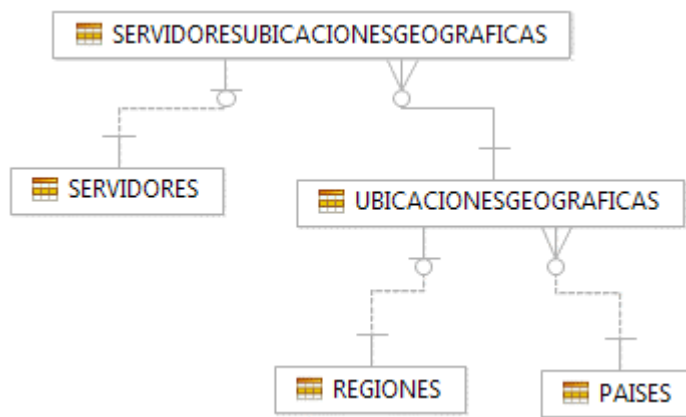


Figura 38. Modelo conceptual del diseño D009. Servidores por ubicación geográfica.

Tabla 34. Descripción de las entidades del modelo lógico para el diseño D009. Servidores por ubicación geográfica.

Descripción de las entidades		
ID	Entidad	Descripción
1	Regiones	Regiones del mundo.
2	ubicacionesGeograficas	Ídem.
3	ubicacionesGeograficasServidoresUbicaciones	Ubicaciones geográficas de los servidores.

	onesGeograficas	
--	-----------------	--

Tabla 35. Descripción de las relaciones entre entidades para el diseño D009.

Servidores por ubicación geográfica.

Descripción de las relaciones			
ID	Entidad	Cardinalidad	Descripción
1	ubicacionesGeograficas- regiones	1:1	Las regiones se utilizan de cero a una vez para definir una ubicación geográfica.
2	ubicacionesGeograficas- paises	1:N	Los países se utilizan cero, una o más veces para definir una ubicación geográfica.
3	servidoresUbicacionesGeograficas - servidores	1:1	Un servidor se registra de cero a una vez en una ubicación geográfica.

4.5.2.9.2.2. Modelo lógico

En la Tabla 36 se detalla el diseño del modelo lógico:

Tabla 36. Modelo de datos lógico para el diseño D009. Servidores por ubicación geográfica.

Entidad:	regiones		
Descripción de atributos			
Atributo	Clave primaria	Clave foránea	Descripción
codigoRegion	X		Identificador de la región.
codigoRegionAlfa			Identificador de la región con dos caracteres.
nombre			Ídem.

Restricciones		
Descripción	Tipo	Descripción
regiones_uchk	Unicidad	El campo codigoRegionAlfa debe ser único.
Comentarios adicionales		
Ninguno.		

Entidad:	ubicacionesGeograficas		
Descripción de atributos			
Atributo	Clave primaria	Clave foránea	Descripción
codigoUbicacion Geografica	X		Identificador de la ubicación geográfica.
codigoPais		X	Identificador del país.
codigoRegion		X	Identificador de la región.
Restricciones			
Descripción	Tipo	Descripción	
ubicacionesGeograficas_paises_fk	Clave foránea	El campo codigoPais debe hacer referencia al campo codigoPais de la tabla paises.	
ubicacionesGeograficas_regiones_fk	Clave foránea	El campo codigoRegion debe hacer referencia al campo codigoRegion de la tabla regiones.	
ubicacionesGeograficas01_uchk	Unicidad	La combinación de los campos codigoPais y codigoRegion debe ser única.	
ubicacionesGeograficas02_uchk	Unicidad	El campo codigoRegion debe ser único.	
Comentarios adicionales			
Ninguno.			

Entidad:	servidoresUbicacionesGeograficas
-----------------	----------------------------------

Descripción de atributos			
Atributo	Clave primaria	Clave foránea	Descripción
codigoServidor	X	X	Identificador del servidor.
codigoUbicacion Geografica	X	X	Identificador de la ubicación geográfica.
Restricciones			
Descripción	Tipo	Descripción	
servidoresUbicacionesGeograficas_servidores_fk	Clave foránea	El campo codigoServidor debe hacer referencia al campo codigoServidor de la tabla servidores.	
servidoresUbicacionesGeograficas_ubicacionesGeograficas_fk	Clave foránea	El campo codigoUbicacionGeografica debe hacer referencia al campo codigoUbicacionGeografica de la tabla ubicacionesGeograficas.	
Comentarios adicionales			
Ninguno.			

4.5.2.9.2.3. Modelo físico

En la figura 39 se muestra el modelo de datos físico.

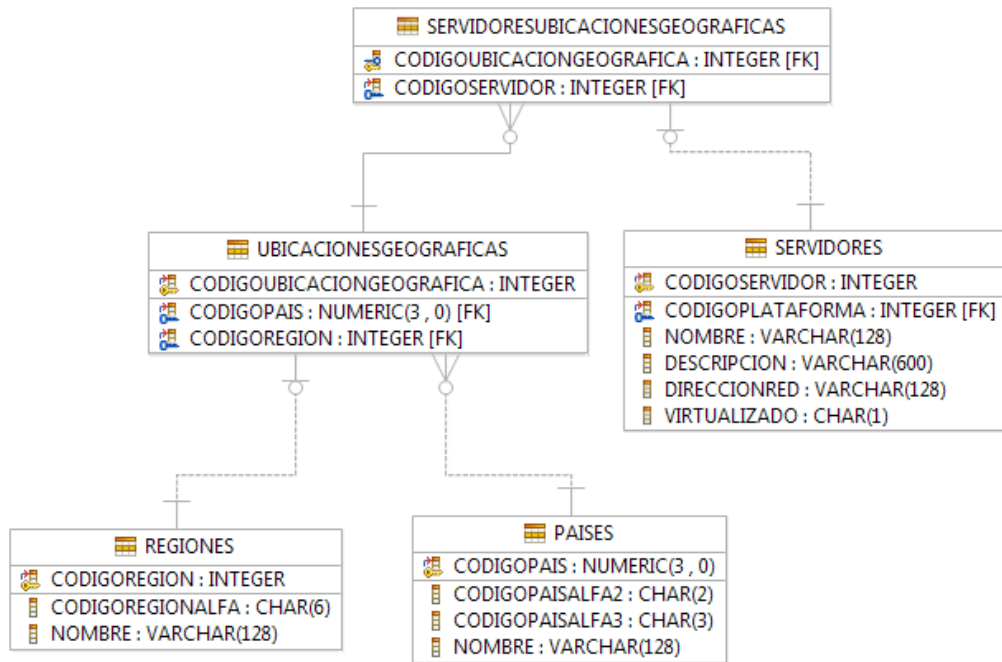


Figura 39. Modelo físico del diseño D009. Servidores por ubicación geográfica (elaboración propia del investigador).

4.5.2.9.3. Justificación del diseño

El modelo cumple con lo establecido en el requerimiento R-008, Consultar información sobre los servidores remotos vinculados, debido a que permite consultar la ubicación geográfica en la cual se encuentran físicamente los servidores de la empresa.

Adicionalmente, el modelo propuesto toma en cuenta las siguientes consideraciones:

- Limita la tabla ubicacionesGeograficas a ingresar un código de región únicamente una vez, ya que las regiones son únicas por país.
- Se limita a que un servidor pueda estar registrado en solo una ubicación geográfica.

4.5.2.10. Diseño D010. Administradores por servidores

4.5.2.10.1. Especificación del diseño

Este diseño permite administrar información sobre los administradores de los servidores. Contempla el siguiente requerimiento:

- R-008. Consultar información sobre los servidores remotos vinculados.

4.5.2.10.2. Propuesta del diseño

4.5.2.10.2.1. Modelo conceptual

En la figura 40 se muestra el diseño conceptual, en la Tabla 37 se detallan las entidades y en la Tabla 38 se detallan las relaciones.

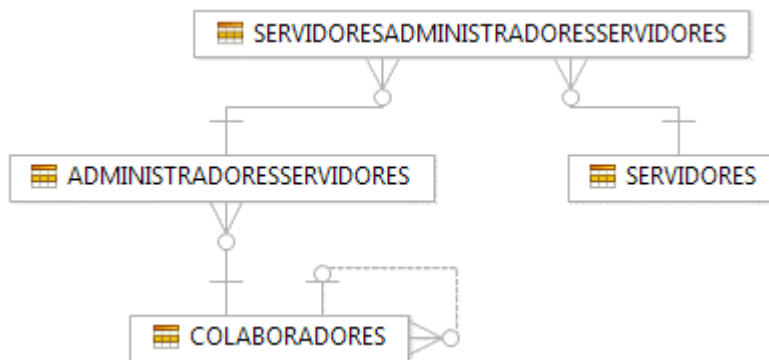


Figura 40. Modelo conceptual del diseño D010. Administradores por servidores.

Tabla 37. Descripción de las entidades del modelo lógico para el diseño D010. Administradores por servidores.

Descripción de las entidades		
ID	Entidad	Descripción
1	administradoresServidores	Son los colaboradores que tienen a su cargo servidores en la compañía.

2	servidoresAdministradoresServidores	Son los servidores que tiene a cargo un administrador de servidores.
---	-------------------------------------	--

Tabla 38. Descripción de las relaciones entre entidades para el diseño D010. Administradores por servidores.

Descripción de las relaciones			
ID	Entidad	Cardinalidad	Descripción
1	administradoresServidores - servidoresAdministradoresServidores	1:N	Un administrador de servidores tiene a su cargo cero, uno o más servidores.
2	servidores - servidoresAdministradoresServidores	1:N	Un servidor está a cargo de cero, uno o más administradores.
3	colaboradores-administradoresServidores	1:1	Un colaborador se registra de cero a una vez como administrador de servidor.

4.5.2.10.2.2. Modelo lógico

En la Tabla 39 se detalla el diseño del modelo lógico:

Tabla 39. Modelo de datos lógico para el diseño D010. Administradores por servidores.

Entidad:	administradoresServidores		
Descripción de atributos			
Atributo	Clave primaria	Clave foránea	Descripción
codigoAdministradorServidor	X		Identificador del administrador del servidor.
Habilitado			Indica si el funcionario actualmente

			tiene el cargo de administrador para el servidor, admite los valores “S” (Sí) o “N” (No).
Restricciones			
Descripción	Tipo	Descripción	
administradoresServidores_colaboradores_fk	Clave foránea	El campo codigoAdministradorServidor debe hacer referencia al campo codigoColaborador de la tabla colaboradores.	
administradoresServidores_habilitado_chk	Comprobación	El campo habilitado admite los valores “S” (Sí) o “N” (No).	
Comentarios adicionales			
Ninguno.			

Entidad:	servidoresAdministradoresServidores		
Descripción de atributos			
Atributo	Clave primaria	Clave foránea	Descripción
codigoServidor	X	X	Identificador del código del servidor.
codigoAdministradorServidor	X	X	Identificador del administrador del servidor.
Restricciones			
Descripción	Tipo	Descripción	
servidoresAdministradoresServidores_administradoresServidores_fk	Clave foránea	El campo codigoAdministradorServidor debe hacer referencia al campo codigoAdministradorServidor de la tabla administradoresServidores	
servidoresAdministradoresServidores_codigoServidor	Clave foránea	El campo codigoServidor debe hacer referencia al campo codigoServidor de la tabla servidores.	

s_servidores_fk		
Comentarios adicionales		
Ninguno.		

4.5.2.10.2.3. Modelo físico

En la figura 41 se muestra el modelo de datos físico.

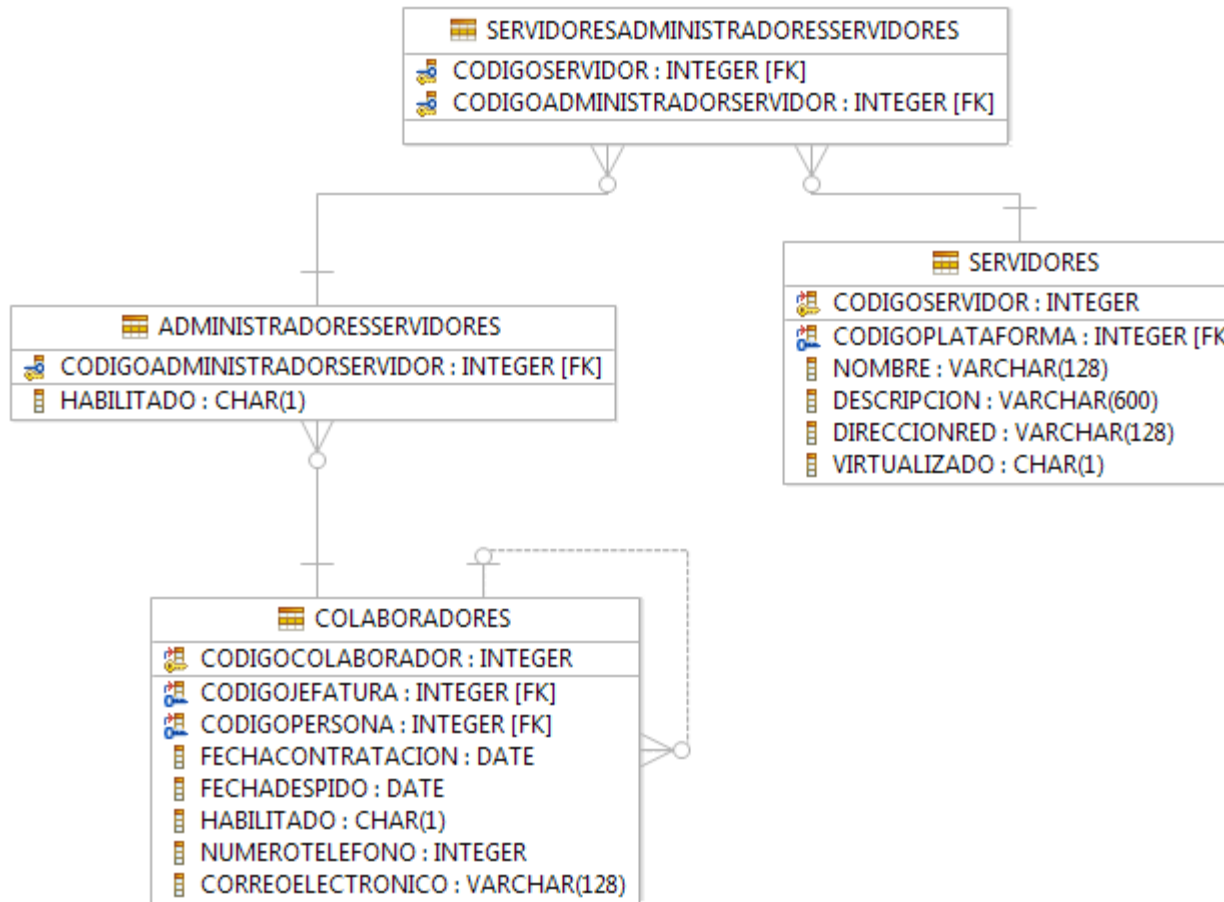


Figura 41. Modelo físico del diseño D010. Administradores por servidores (elaboración propia del investigador).

4.5.2.10.3. Justificación del diseño

El modelo cumple con lo establecido en el requerimiento R-008, Consultar información sobre los servidores remotos vinculados, debido a que

permite consultar sobre los administradores de servidores que se encuentra a cargo de estos.

4.5.2.11. Diseño D011. Administradores de bases de datos por instancia.

4.5.2.11.1. Especificación del diseño

Este diseño permite administrar información sobre los administradores de las instancias de bases de datos. Contempla el siguiente requerimiento:

- R-008. Consultar información sobre los servidores remotos vinculados.

4.5.2.11.2. Propuesta del diseño

4.5.2.11.2.1. Modelo conceptual

En la figura 42 se muestra el diseño conceptual, en la Tabla 40 se detallan las entidades y en la Tabla 41 se detallan las relaciones.

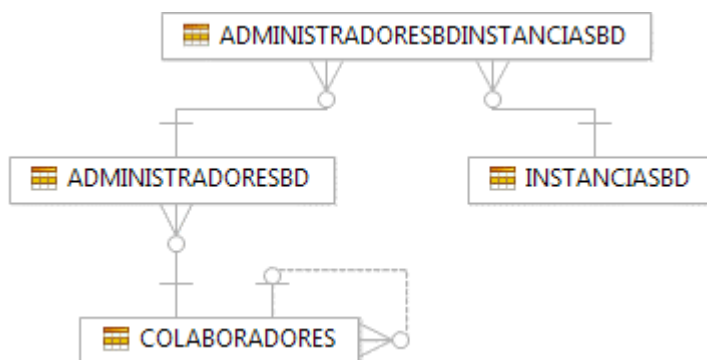


Figura 42. Modelo conceptual del diseño D011. Administradores de bases de datos por instancia.

**Tabla 40. Descripción de las entidades del modelo lógico para el diseño D011.
Administradores de bases de datos por instancia.**

Descripción de las entidades		
ID	Entidad	Descripción
1	AdministradoresBd	Administradores de bases de datos.
2	AdministradoresBdInstanciasBd	Contiene la relación entre los administradores de bases de datos por instancia.

**Tabla 41. Descripción de las relaciones entre entidades para el diseño D011.
Administradores de bases de datos por instancia.**

Descripción de las relaciones			
ID	Entidad	Cardinalidad	Descripción
1	InstanciasBd - AdministradoresBdInstanciasBd	1:N	Una instancia de base de datos está a cargo de cero, uno o más administradores de bases de datos.
2	AdministradoresBd - AdministradoresBdInstanciasBd	1:N	Un administrador de bases de datos tiene a su cargo cero, una o más instancias.
3	colaboradores-administradoresBd	1:1	Un colaborador se registra de cero a una vez como administrador de base de datos.

4.5.2.11.2.2. Modelo lógico

En la Tabla 42 se detalla el diseño del modelo lógico:

Tabla 42. Modelo de datos lógico para el diseño D011. Administradores de bases de datos por instancia.

Entidad:	administradoresBd		
Descripción de atributos			
Atributo	Clave primaria	Clave foránea	Descripción
codigoAdministradorBd	X		Identificador del administrador de base de datos.
Habilitado			Indica si el funcionario actualmente tiene el cargo de administrador de base de datos. Admite los valores “S” (Sí) o “N” (No).
Restricciones			
Descripción	Tipo	Descripción	
administradoresBd_colaboradores_fk	Clave foránea	El campo codigoAdministradorBd debe hacer referencia al campo codigoColaborador de la tabla colaboradores.	
administradoresBd_habilitado_chk	Comprobación	El campo habilitado admite los valores “S” (Sí) o “N” (No).	
Comentarios adicionales			
Ninguno.			

Entidad:	administradoresBdInstanciasBd		
Descripción de atributos			
Atributo	Clave primaria	Clave foránea	Descripción
codigoInstanciaBd	X	X	Identificador de la instancia de base de datos.

codigoAdministradorBd	X	X	Identificador del administrador de base de datos.
Restricciones			
Descripción	Tipo	Descripción	
administradoresBdInstanciasBd_administradoresBd_fk	Clave foránea	El campo codigoAdministradorBd debe hacer referencia al campo codigoAdministradorBd de la tabla administradoresBd.	
administradoresBdInstanciasBd_instanciasBd_fk	Clave foránea	El campo codigoInstanciaBd debe hacer referencia al campo codigoInstanciaBd de la tabla instanciasBd.	
Comentarios adicionales			
Ninguno.			

4.5.2.11.2.3. Modelo físico

En la figura 43 se muestra el modelo de datos físico.

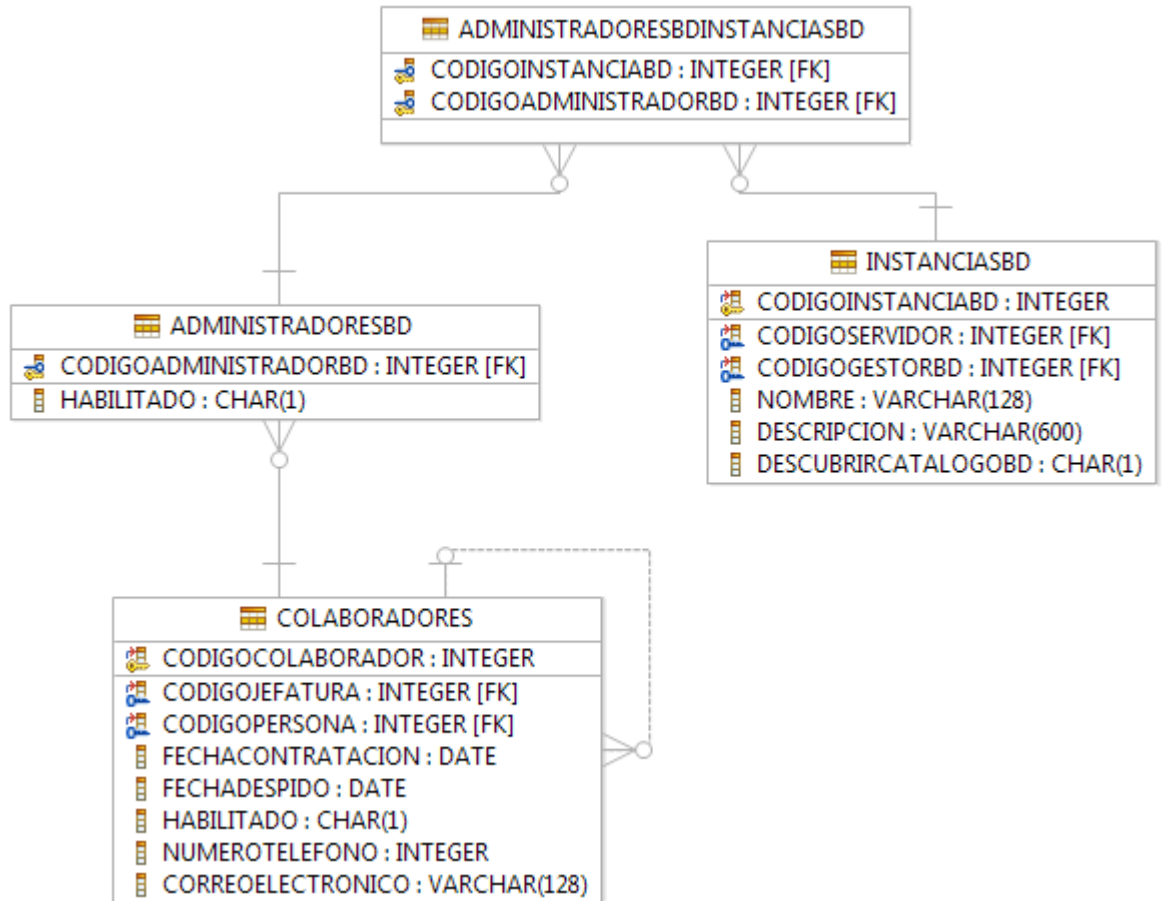


Figura 43. Modelo físico del diseño D011. Administradores de bases de datos por instancia (elaboración propia del investigador).

4.5.2.11.3. Justificación del diseño

El modelo cumple con lo establecido en el requerimiento R-008, Consultar información sobre los servidores remotos vinculados, debido a que permite consultar los administradores de bases de datos que se encuentran a cargo de las instancias de bases de datos.

4.5.3. Modelamiento de procesos para la integración de datos

4.5.3.1. Proceso de extracción

Se utiliza la característica de base de datos federada incorporada en el gestor de base de datos *SQL Server* para realizar consultas distribuidas en los orígenes. Se utilizó *OpenQuery*, ya que permite emplear el dialecto nativo de cada uno de los gestores con la finalidad de extraer los metadatos que describen a los objetos contenidos en las bases de datos, debido a que esta facilita la comunicación entre los computadores. Los fabricantes de las bases de datos proveen los conectores necesarios (ODBC, JDBC, OLEDB) para el uso de dicha tecnología.

Es necesario vincular las bases de datos que forman parte de la federación, para esto se requiere configurar nombres únicos de los servidores a los cuales se desea acceder de manera remota. En la figura 44 se muestra cómo quedan registrados los servidores vinculados con una base de datos de tipo *SQL Server*.

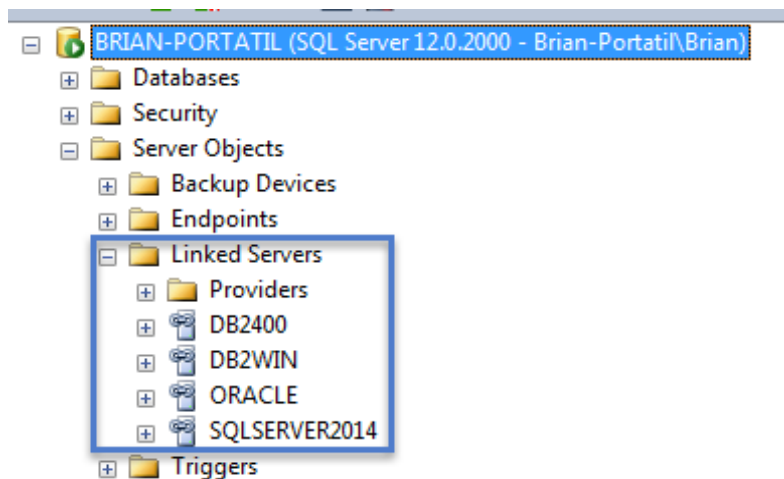


Figura 44. Servidores vinculados en *SQL Server*.

El proceso de extracción se ilustra en la figura 45.

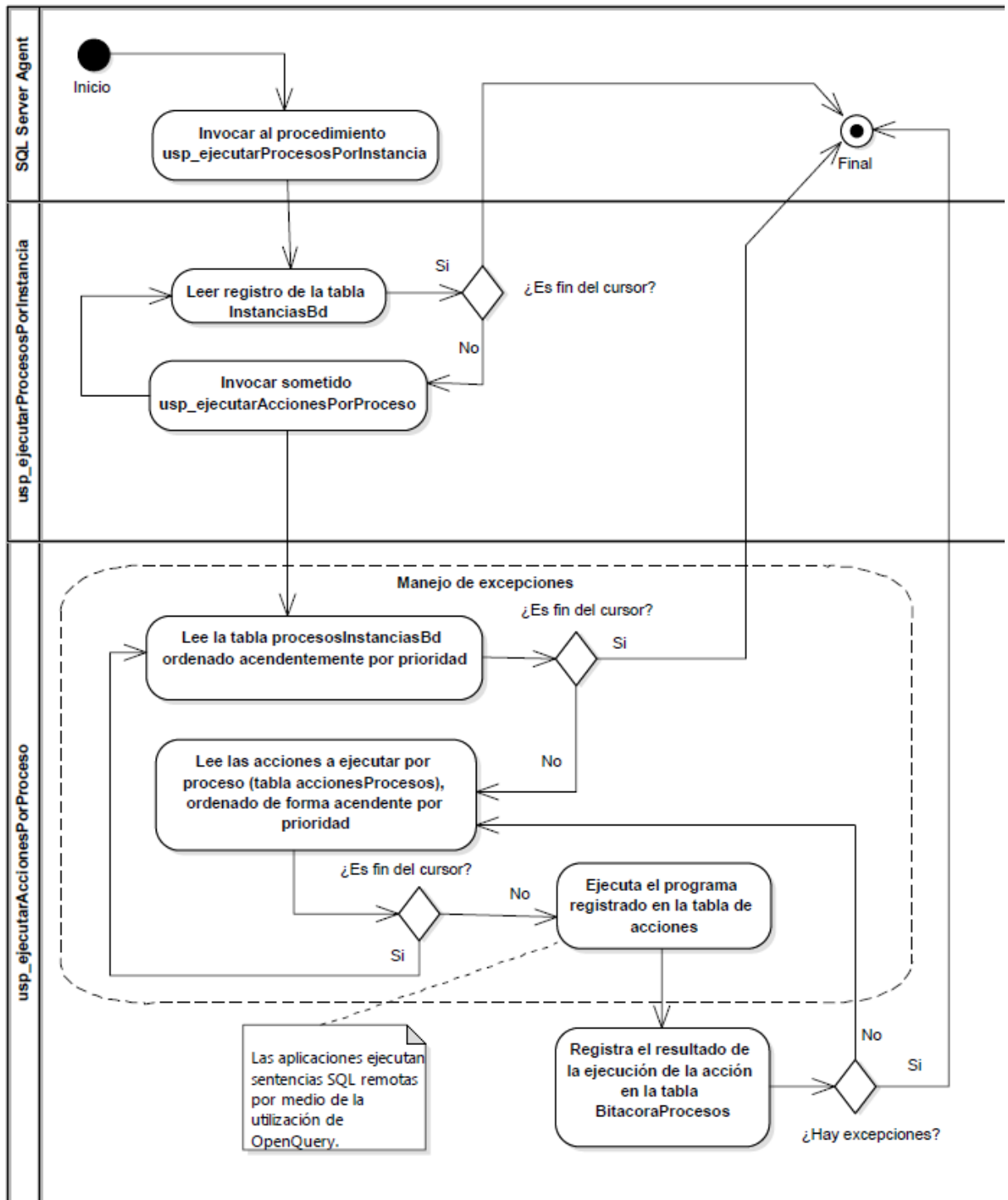


Figura 45. Diagrama de actividad del proceso de extracción.

El proceso de extracción inicia cuando se ejecuta la tarea programada en el componente *SQL Server Agent*. Este invoca al procedimiento almacenado

usp_ejecutarProcesosPorInstancia, el cual recorre, por medio de un cursor, los registros de la tabla instanciasBd. Por cada uno de estos invoca al procedimiento almacenado usp_ejecutarAccionesPorProceso (haciendo uso de *SQL* dinámico), que recorre los procesos que deben ser ejecutados por instancia ordenados por prioridad, de manera ascendente, en la tabla procesosInstanciasBd. Una vez identificado el proceso, busca la lista de acciones por realizar para el proceso en específico, ordenado por prioridad de manera ascendente en la tabla accionesProcesos; posteriormente se ejecuta la aplicación registrada en la acción y se guarda el resultado de la ejecución de esta en la tabla bitacoraProcesos.

El procedimiento almacenado usp_ejecutarAccionesPorProceso tiene bajo su responsabilidad el monitoreo de las excepciones que se puedan presentar en la ejecución de las acciones para un proceso. En caso que se presente una excepción, la registra en la tabla bitacoraProcesos y no continúa con la ejecución de las acciones posteriores.

Las aplicaciones definidas en las acciones hacen uso de *openQuery* para ejecutar consultas distribuidas en los gestores de bases de datos heterogéneos. Estas utilizan instrucciones *SQL* con el dialecto nativo de cada gestor. Se construye una aplicación por cada objeto de base de datos con el que se desea trabajar (realiza consultas específicas al diccionario de datos del gestor). Se decide diseñarlo de esta manera porque registrar en una tabla instrucciones *SQL* implica un riesgo para la seguridad de los sistemas, por lo que si se requiere agregar funcionalidad, es preferible que se creen y publiquen de forma controlada los procedimientos almacenados necesarios y, de esta manera, no comprometer la seguridad del sistema.

Se decide no utilizar herramientas como *Integration services* o similares, debido a que lo que se quiere integrar son gestores de bases de datos y no otro tipo de fuentes; además, la solución propuesta permite que el modelo se pueda implementar sin necesidad de adquirir ese tipo de herramientas, lo que facilita su portabilidad a otras plataformas.

Para la incorporación de las características de descubrimiento automático de base de datos e inclusión o exclusión de bases de datos en el mapeo de los objetos, se utiliza el modelo de datos que se muestra en la figura 46.

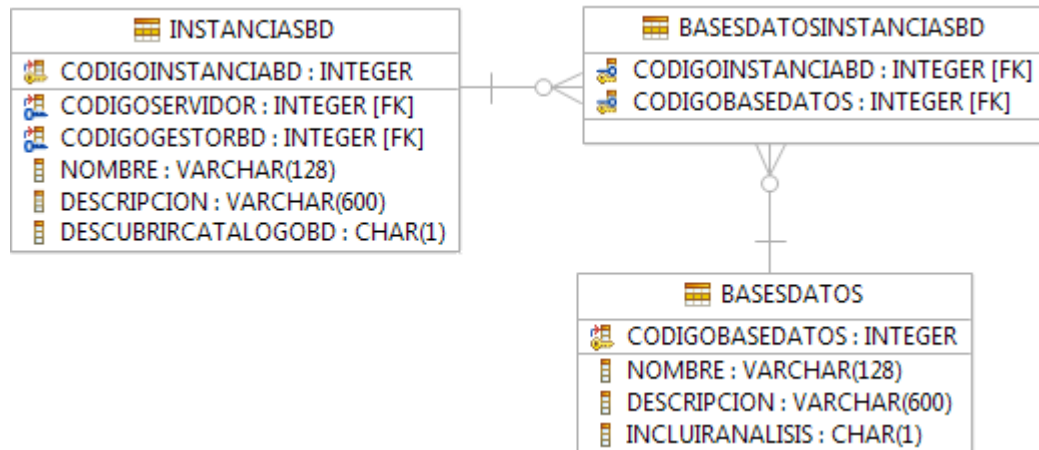


Figura 46. Extracto del modelo físico del diseño D002. Administración de bases de datos y esquemas por instancia.

Se incorpora la característica de descubrir automáticamente las bases de datos contenidas en un gestor. Para ello se agregó en la tabla `instanciasBd` el campo `descubrirCatalogoBd` (cuando el valor es “S”); este es leído por un procedimiento almacenado que forma parte de una acción en el proceso de extracción, el cual lee del gestor origen, el catálogo de bases de datos que contiene. Si encuentra alguna diferencia con las bases de datos registradas por instancia en la tabla `basesDeDatosInstanciasBd`, la agrega de forma automática en la tabla `basesDatos`, con el campo `incluirAnálisis` con “S”. En caso de existir, no la modifica, y en caso de eliminación, el sistema no lo elimina, esta debe ser manual.

Se incorpora la característica para incluir o excluir el mapeo de los objetos por base de datos. Para ello se agregó el campo `incluirAnálisis` en la tabla `basesDatos`, el cual puede tener el valor de “S” (Sí) o “N” (No). El análisis consiste en la revisión de los metadatos que describen a los objetos contenidos en los gestores de base de datos para una base de datos, con los cuales se poblará la tabla `objetosBd` (véase figura 19).

Un ejemplo sobre la utilización de los campos `descubrirCatalogoBd` e `incluirAnalisis` consiste en descubrir todas las bases de datos de un gestor remoto y excluir las bases de datos que forman parte del sistema, como es el caso de las bases de datos *master*, *model* y *tempdb*, para una instancia de *SQL Server*. En la figura 47 se muestra, de manera gráfica, la visualización de los catálogos de bases de datos en *SQL Server*.

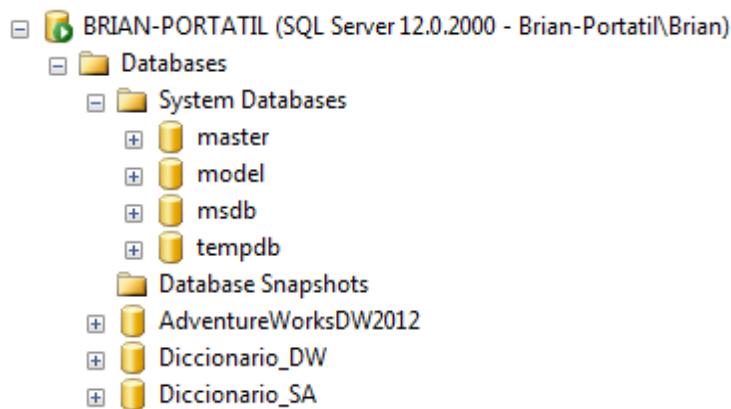


Figura 47. Ejemplo de catálogos de bases de datos en *SQL Server*.

Las consultas realizadas en las bases de datos se especifican en el apéndice 4 y son el resultado del análisis de los catálogos del sistema. Para su elaboración se utilizó como referencia lo especificado por IBM (2002), *Microsoft* (2014), *Oracle* (2014b) y ANSI/ISO/IEC (1999).

4.5.3.2. Proceso de transformación

El proceso de transformación se lleva a cabo en el momento en que los programas registrados en la tabla `acciones` incluyen en la tabla `objetosBd` la referencia del campo `codigoClaseObjetoBd`, debido a que el sistema debe transformar los códigos nativos de las clases de objetos establecidos en los gestores de bases de datos de origen, y utilizar en su lugar, los nombres configurados en la tabla `clasesObjetosBd` del modelo propuesto, con la finalidad de estandarizar el nombre de las clases de objetos. Adicionalmente, por cada inserción en la tabla `objetosBd` se genera un identificador único por objeto (es un campo autogenerado).

Un ejemplo práctico en donde el sistema realiza transformación de las clases de objetos, es cuando el sistema lee la tabla `sysobjects`, en un gestor de *SQL Server*, la cual utiliza en la columna `xtype` nomenclaturas para identificar a los tipos de objetos. Un caso en concreto es que para las funciones utiliza la nomenclatura `FN`; sin embargo, en la tabla `clasesObjetosBd` se especifica el nombre función, el sistema almacenará en el campo `clasesObjetosBd` de la tabla `objetosBd` la referencia a la descripción de “función” en lugar de su código nativo “FN”.

4.5.3.3. Proceso de carga

El proceso de carga consta de las siguientes etapas:

- Carga del área de pruebas al almacén de datos operacional.
- Carga del almacén de datos operacional al almacén de datos.
- Carga del almacén de datos a los cubos multidimensionales.

La carga del área de pruebas (SA) hacia el almacén de datos operacional (ODS) consiste en registrar los objetos que se encuentren en el área de pruebas, que no han sido registrados previamente en el almacén de datos operacional. De esta forma, en el almacén de datos operacional, los objetos tienen un identificador único.

La carga del almacén de datos operacional (ODS) hacia el almacén de datos (DW) consiste en tomar los datos que este contiene en una fecha determinada y cargar el modelo de datos multidimensional, el cual permite guardar información de forma histórica y que a su vez facilita a los usuarios el descubrimiento de patrones interesantes, a través de las dimensiones y métricas que este posee.

La carga del almacén de datos (DW) hacia los cubos multidimensionales consiste en procesar los datos en el gestor de base de datos multidimensional *analysis services*, el cual expone la información a los usuarios autorizados.

4.5.4. Diseño multidimensional

4.5.4.1. Modelo conceptual

La conceptualización del modelo multidimensional, parte de la definición de las tablas de dimensiones y hechos, como se muestra en la Tabla 43.

Tabla 43. Conceptualización del modelo multidimensional, dimensiones y hechos.

Tablas de dimensión
Dimensión de tiempo fechas_dim
Dimensión geográfica regiones_dim paises_dim
Otras dimensiones objetosBd_dim clasesObjetosBd_dim controles_dim conceptos_dim comentarios_dim propietariosTitulares_dim propietariosDelegados_dim esquemasBd_dim basesDatos_dim instanciasBd_dim servidores_dim
Tabla de hechos - catalogoObjetosBd_th
Métricas Cantidad de objetos

A continuación se detalla la especificación de cada una de las dimensiones y la tabla de hechos:

La dimensión fechas_dim permite filtrar los datos contenidos en el modelo por períodos de tiempo, este permite ver los cambios realizados de forma histórica; por ejemplo, se pueden seleccionar los conceptos relacionados con un objeto en enero y luego en febrero, para determinar los cambios a través del tiempo.

La dimensión regiones_dim permite filtrar los datos contenidos en el modelo por región geográfica; por ejemplo, un usuario puede seleccionar los datos que corresponden a la región de San José.

La dimensión paises_dim permite filtrar los datos contenidos en el modelo por país; por ejemplo, un usuario puede seleccionar los datos que corresponden a Costa Rica y, de esta manera, restringir el volumen de datos con los cuales desea trabajar.

La dimensión objetosBd_dim permite consultar el detalle de los objetos contenidos en las bases de datos heterogéneas de forma integrada; por ejemplo, el usuario puede visualizar el nombre de todos los objetos del tipo tabla que se encuentran relacionados con el concepto de “dato cifrado” en la dimensión de conceptos (conceptos_dim), con la finalidad de identificar en cuáles servidores se encuentran.

La dimensión clasesObjetosBd_dim permite filtrar los objetos de bases de datos que corresponden a un tipo en particular; por ejemplo, se pueden filtrar los objetos de clase tabla o procedimiento almacenado.

La dimensión controles_dim permite filtrar los objetos de bases de datos que tienen asignados un control en particular; por ejemplo, se pueden filtrar los objetos que tienen relacionado el control de “respaldo diario”.

La dimensión conceptos_dim permite filtrar los objetos de bases de datos que tienen asignados un concepto en particular; por ejemplo, se pueden filtrar los objetos que tienen relacionado el concepto “pago de planillas”.

La dimensión comentarios_dim permite consultar los comentarios realizados a un objeto.

La dimensión propietariosTitulares_dim permite identificar el propietario titular de un objeto de base de datos en particular.

La dimensión propietariosDelegados_dim permite identificar los propietarios delegados de un objeto de base de datos en particular.

La dimensión esquemasBd_dim permite filtrar los objetos por esquemas de bases de datos.

La dimensión basesDatos_dim permite filtrar los objetos por bases de datos.

La dimensión instanciasBd_dim permite filtrar los objetos por instancias de bases de datos.

La dimensión servidores_dim permite filtrar los objetos de bases de datos por servidor, plataforma y gestor de base de datos.

La tabla de hechos catalogoObjetosBd_th hace referencia a las tablas de dimensiones descritas anteriormente, contiene de métrica el campo de cantidad; por ejemplo, esta permite medir la cantidad de objetos relacionados con un concepto.

El modelo de datos conceptual correspondiente al modelo multidimensional se adjunta en el apéndice 6, y en el apéndice 7 se muestra el diagrama de datos físico.

4.5.4.2. Modelo físico de datos

Las tablas de dimensiones se nombran utilizando el sufijo "_dim", en el caso de las tablas de hechos se utiliza "_th", las demás tablas en las cuales no se utilizan los sufijos son auxiliares. Los campos con el prefijo "ref" corresponden a las llaves naturales, las demás llaves utilizadas corresponden a llaves sustitutas. A continuación se exponen cada uno de los modelos de datos físicos utilizados para construir el modelo multidimensional, en el cual se emplea el esquema de “estrella” y “copo de nieve”. El modelo de datos físico correspondiente al modelo multidimensional se adjunta en el apéndice 7.

4.5.4.2.1.1. Dimensiones regiones y países

Las dimensiones regiones y países proveen un mecanismo para filtrar los objetos de bases de datos por país y región, conforman un “copo de nieve”. En la figura 48 se muestra el modelo físico de datos.

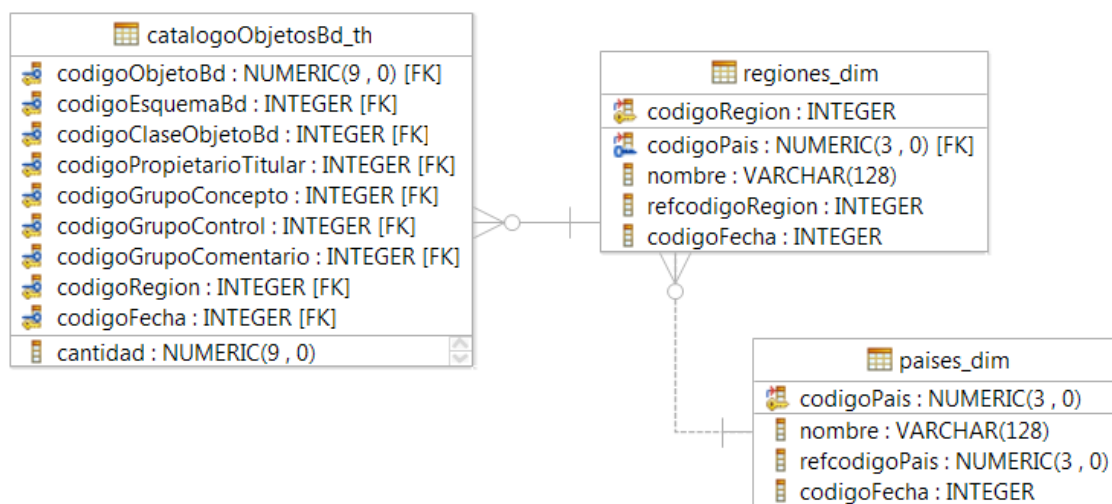


Figura 48. Modelo de datos físico para las dimensiones regiones y países.

4.5.4.2.1.2. Dimensión conceptos

Esta dimensión permite buscar los objetos de bases de datos por sus conceptos asociados. Se incorpora una relación de tipo padre-hijo (recursiva) en los campos codigoConcepto y codigoConceptoPadre de la tabla conceptos_dim, que facilita la construcción de jerarquías de forma automática; por medio de estas es posible seleccionar un concepto padre con todos los conceptos hijos relacionados. Un ejemplo práctico de la utilidad de esta característica se presenta cuando se requiere consultar todos los objetos relacionados con un concepto padre “información sensible”, este a su vez, contiene a los conceptos hijos “información sensibilidad alta” e “información sensibilidad media”, con solo el hecho de seleccionar en la jerarquía al concepto padre, la consulta realizada al modelo multidimensional devolverá toda la información asociada al concepto padre y a sus hijos. Este diseño contempla el uso de relaciones de muchos a

muchos, ya que un objeto de base de datos puede estar relacionado con más de un concepto, para ello se incorporan una tabla de intersección (gruposConceptos) y una tabla puente (miembrosGruposConceptos). En la figura 49 se muestra el modelo físico de datos.

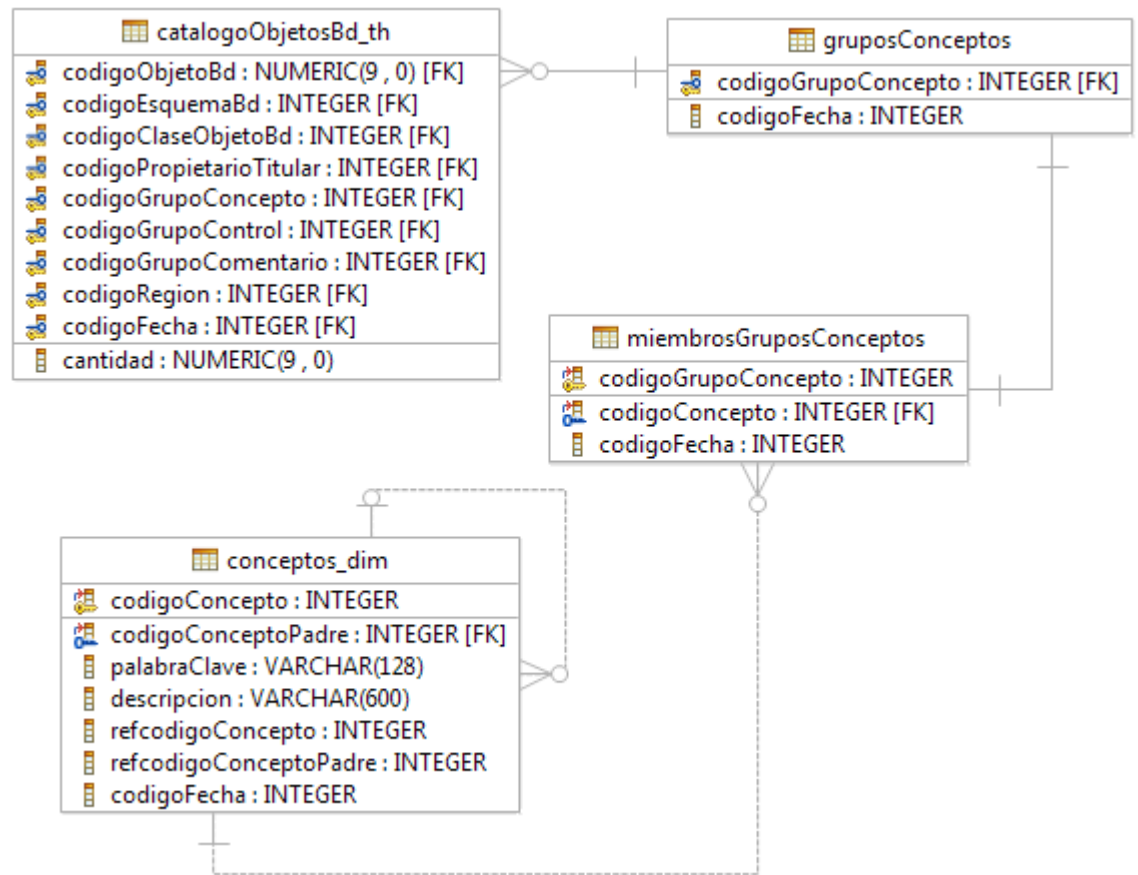


Figura 49. Modelo de datos físico para la dimensión conceptos.

4.5.4.2.1.3. Dimensión fechas

Esta dimensión permite realizar búsquedas de información de forma histórica. Para registrar información histórica se requiere administrar la versión de los registros en cada una de las entidades del diseño multidimensional (con la finalidad de tratar adecuadamente las dimensiones lentamente cambiantes de

tipo 2), para ello se incorpora en todas las entidades el campo codigoFecha, el cual tiene dos funciones, una es para identificar la versión de los registros y la otra es para identificar los registros cargados en una fecha en particular en cada una de las tablas. El campo codigoFecha debe contener el valor que se utiliza en la tabla fechas para el campo que lleva el mismo nombre, de esta forma es posible alinear el versionamiento de los registros de todo el modelo multidimensional. En la figura 50 se muestra el modelo físico de datos.

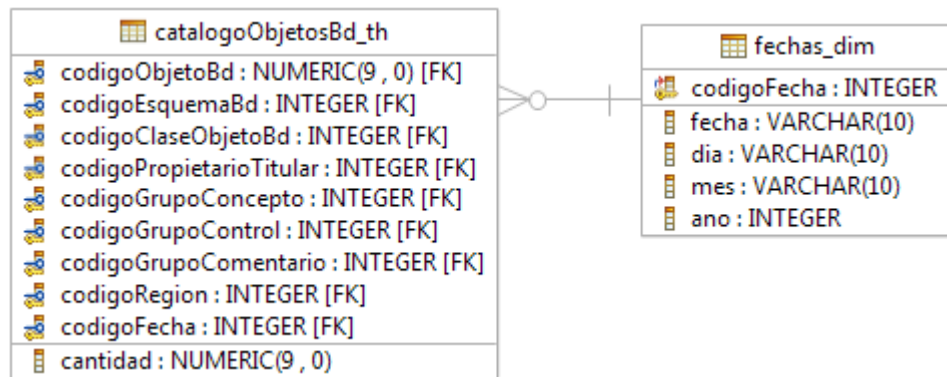


Figura 50. Modelo de datos físico para la dimensión fechas.

4.5.4.2.1.4. Dimensión controles

Esta dimensión permite buscar los objetos de bases de datos por sus controles asociados. Este diseño contempla el uso de relaciones de muchos a muchos, ya que un objeto de base de datos puede estar relacionado con más de un control, para ello se incorporan una tabla de intersección (gruposControles) y una tabla puente (miembrosGruposControles). En la figura 51 se muestra el modelo físico de datos.

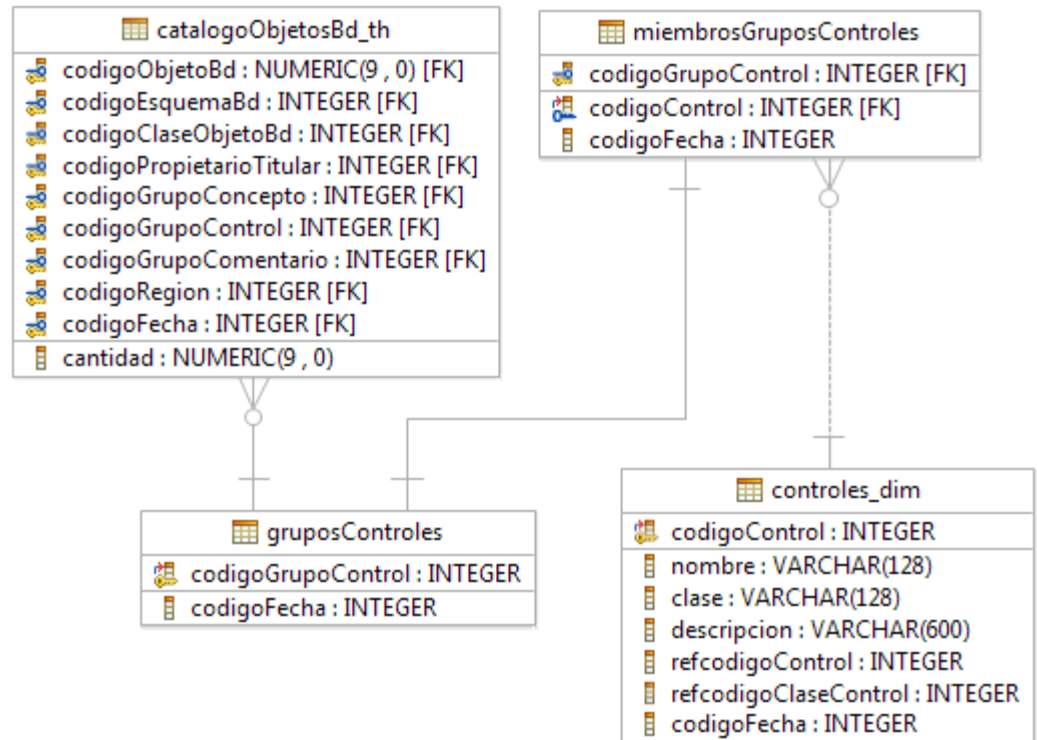


Figura 51. Modelo de datos físico para la dimensión controles.

4.5.4.2.1.5. Dimensión comentarios

Esta dimensión permite consultar los comentarios asociados con un objeto de base de datos. Este diseño contempla el uso de relaciones de muchos a muchos, ya que un objeto de base de datos puede estar relacionado con más de un comentario, para ello se incorporan una tabla de intersección (gruposComentarios) y una tabla puente (miembrosGruposComentarios). En la figura 52 se muestra el modelo físico de datos.

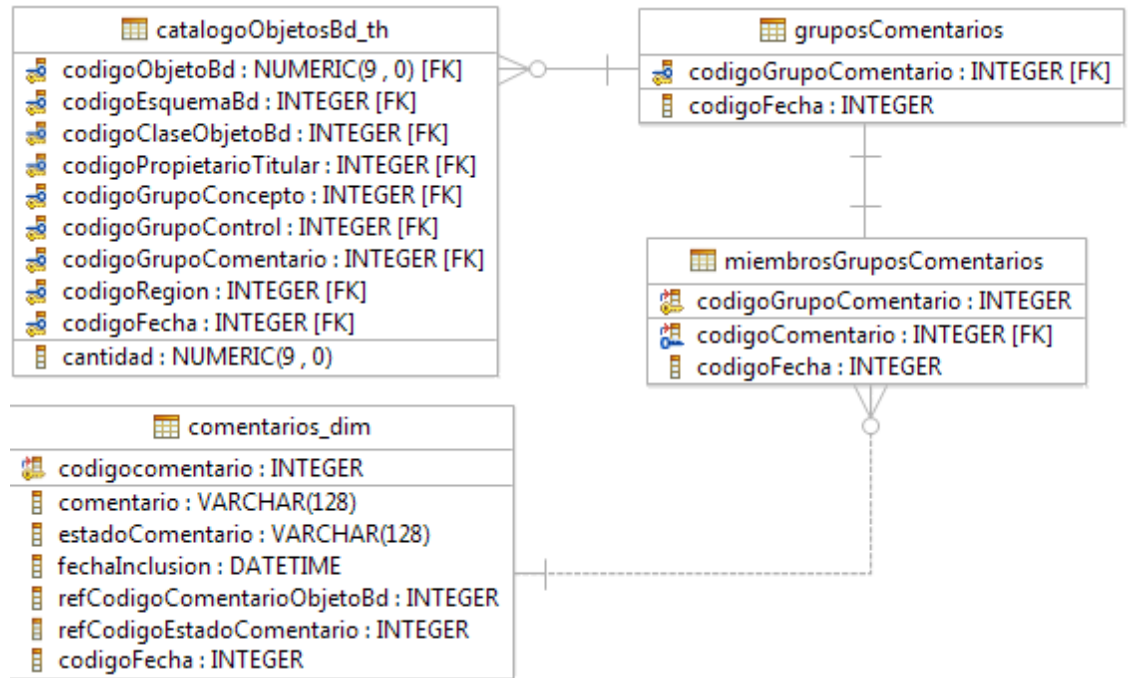


Figura 52. Modelo de datos físico para la dimensión comentarios.

4.5.4.2.1.6. Dimensiones propietarios titulares y propietarios delegados

Esta dimensión permite consultar los propietarios de los objetos de bases de datos, sean titulares o delegados. Este diseño contempla el uso de relaciones de muchos a muchos, ya que un objeto de base de datos puede estar relacionado con más de un propietario delegado, para ello se incorporan una tabla de intersección (gruposPropietariosDelegados) y una tabla puente (miembrosGruposPropietariosDelegados). En la figura 53 se muestra el modelo físico de datos.

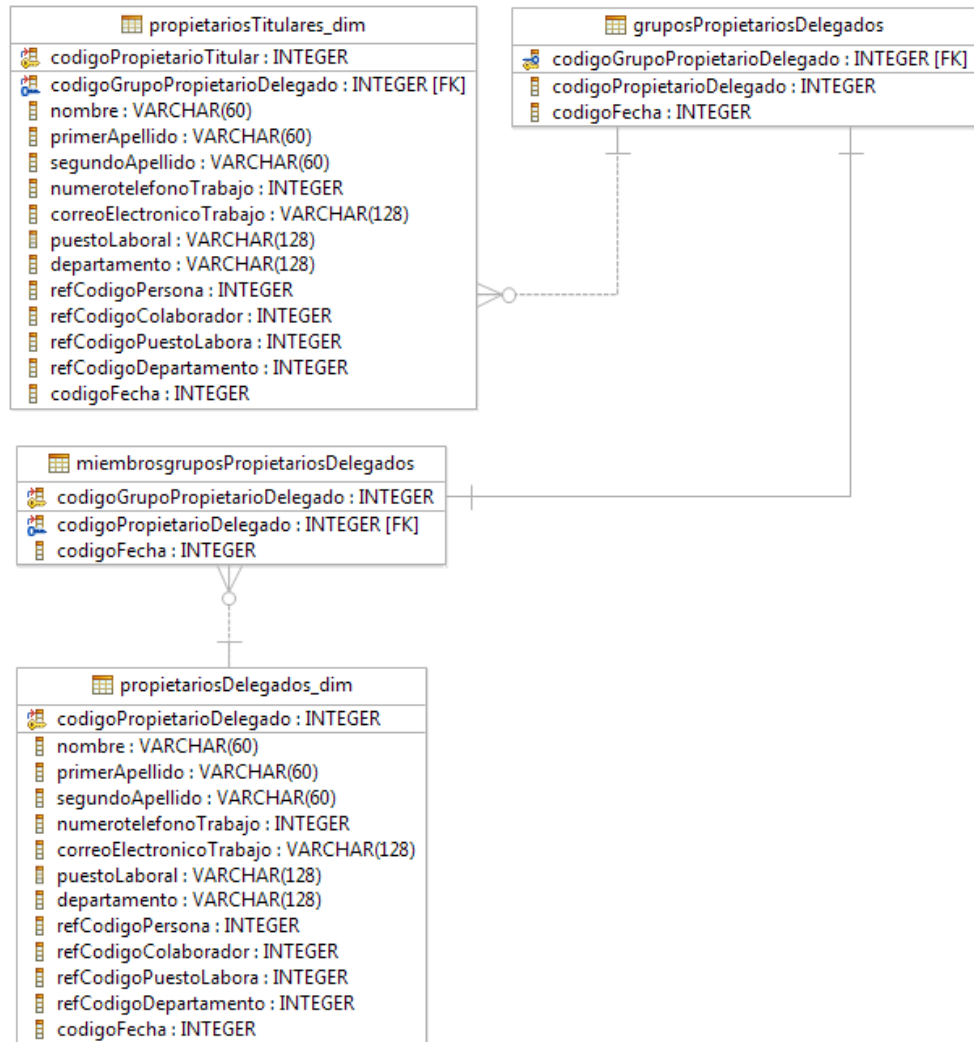


Figura 53. Modelo de datos físico para las dimensiones propietarios titulares y propietarios delegados.

4.5.4.2.1.7. Dimensiones esquemas, bases de datos, instancias de bases de datos y servidores

Las dimensiones esquemas, bases de datos, instancias de bases de datos y servidores, conforman un “copo de nieve”, el cual permite a los usuarios navegar, a través de estos, para localizar los objetos de bases de datos con los cuales se desea trabajar. En la figura 54 se muestra el modelo físico de datos.

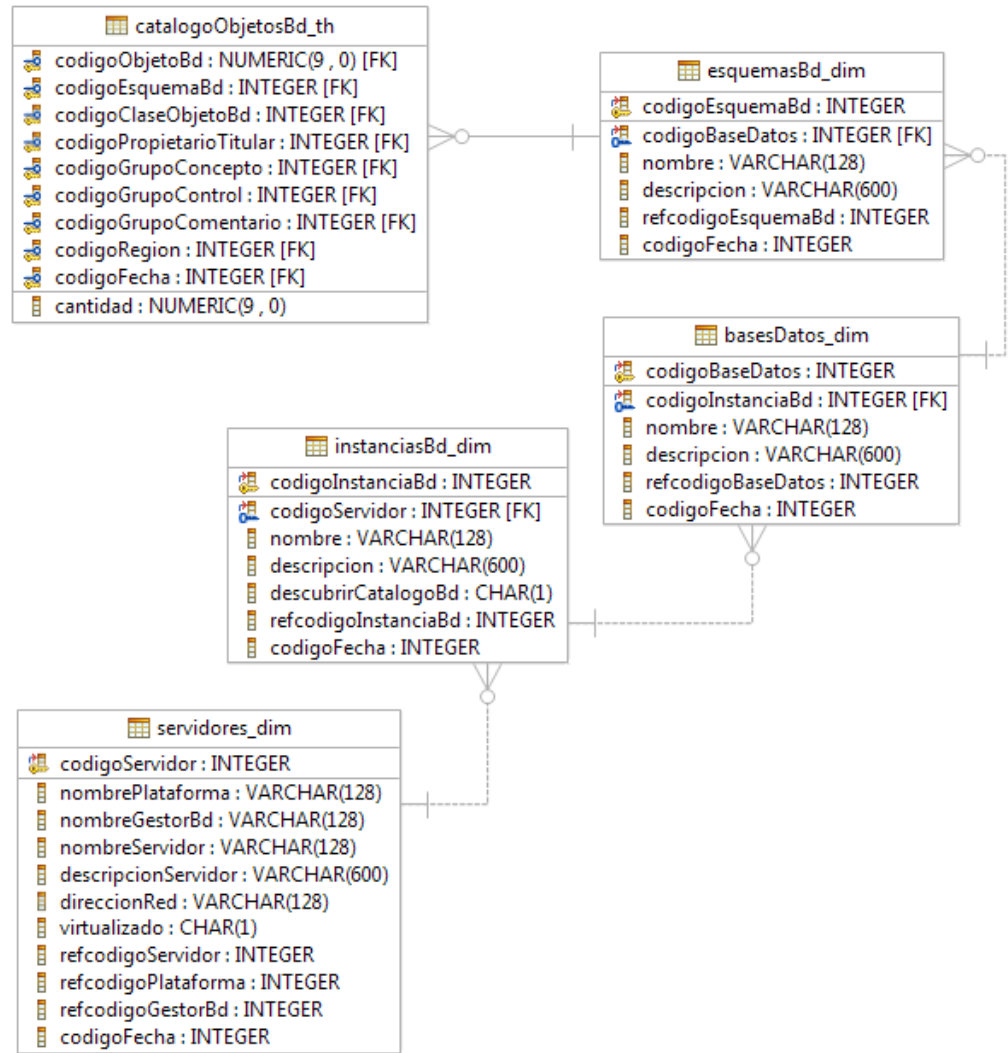


Figura 54. Modelo de datos físico para las dimensiones esquemas, bases de datos, instancias de bases de datos y servidores.

4.5.4.2.1.8. Dimensión objetos de base de datos

Esta dimensión permite consultar información específica sobre los objetos de base de datos. En la figura 55 se muestra el modelo físico de datos.

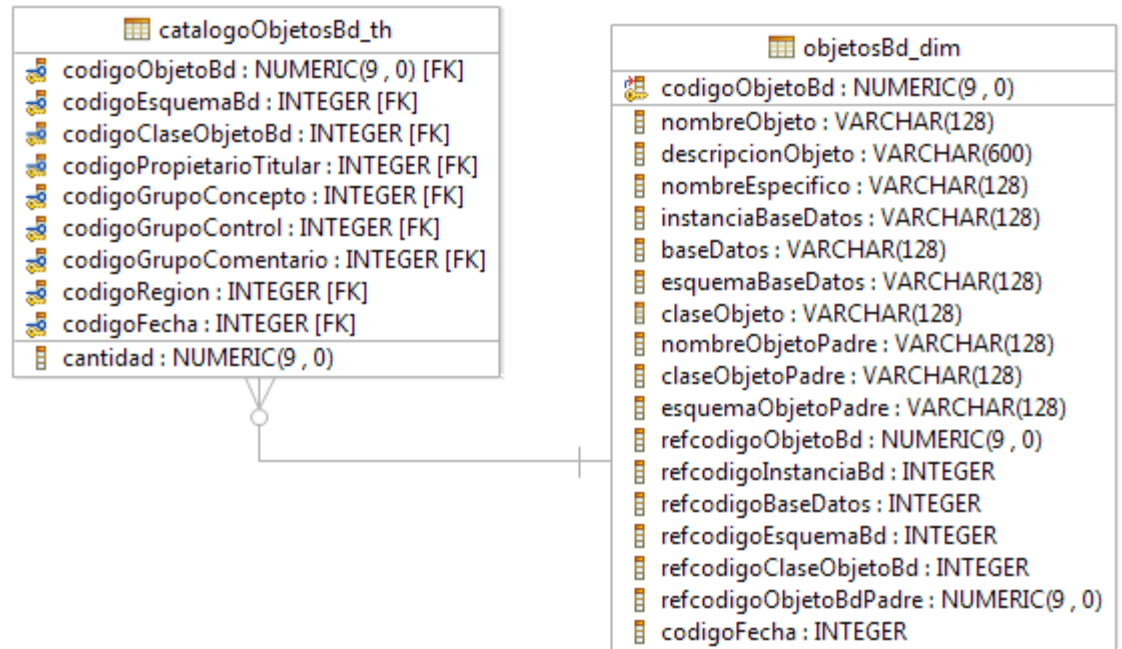


Figura 55. Modelo de datos físico para la dimensión conceptos.

4.5.4.2.1.9. Dimensión clases de objetos de bases de datos

Esta dimensión permite buscar los objetos de bases de datos por su clase de objeto de base de datos, tales como tablas y procedimientos. En la figura 56 se muestra el modelo físico de datos.

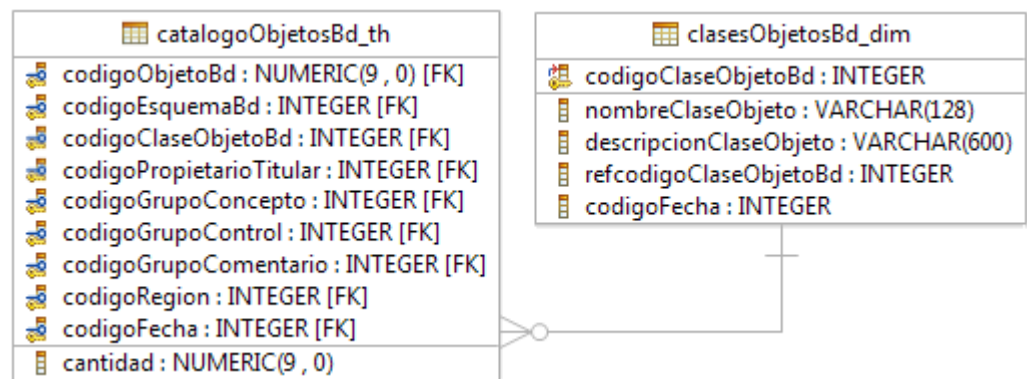


Figura 56. Modelo de datos físico para la dimensión clases de objetos de base de datos.

4.5.1. Justificación del diseño multidimensional

Se requiere de un modelo multidimensional por dos razones, primero porque este permite almacenar información de forma histórica, lo cual no es posible con el modelo relacional expuesto en almacén de datos operacional; la segunda es que facilita a los usuarios analizar los datos, permitiéndoles interactuar con estos en modo de “descubrimiento”.

4.5.2. Cubo multidimensional

Para el tratamiento del cubo multidimensional se genera un proyecto de *analysis services*, por medio de la herramienta de *Visual Studio*, en el cual se modela la estructura del cubo. Es importante recalcar que en el momento de compilación de este es cuando se realiza la carga de la información contenida en los orígenes de datos hacia el gestor de bases de datos multidimensional (*analysis services*). Para el caso en estudio es a partir del almacén de datos operacional. En la figura 57 se muestra la estructura del proyecto generado para la construcción del cubo, en la figura 58 se muestra cómo los usuarios pueden visualizar el cubo por medio la herramienta *Tableau*.

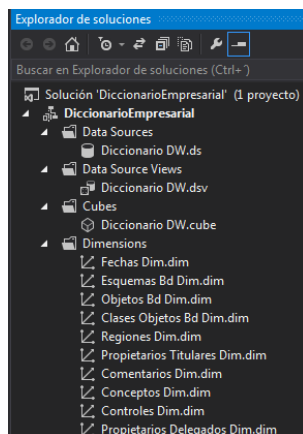


Figura 57. Modelo de cubo multidimensional en *Visual Studio*.

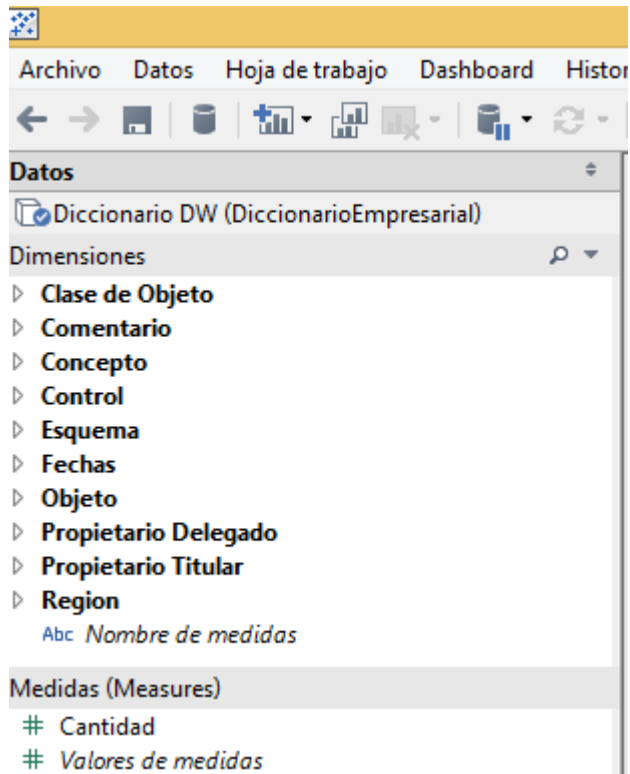


Figura 58. Visualización del cubo por medio de la herramienta *Tableau*.

4.5.3. Recomendación de la herramienta para la visualización de los datos

Como base para decidir la herramienta de visualización se utilizó el estudio expuesto por Gartner Inc. (Gartner Inc, 2015), en el cual analizan las principales herramientas analíticas existentes en el mercado que permiten la visualización de datos, siendo *Tableau* líder en el sector. Se recomienda su utilización, ya que proporciona los conectores de datos necesarios para comunicarse con diversos orígenes de datos, que permiten una conectividad a gestores de datos relacionales y multidimensionales. Para el caso en estudio, facilita la consulta de los datos expuestos en el modelo relacional del almacén de datos operacional y en el gestor multidimensional de *analysis services*. A su vez, permite satisfacer el requerimiento R-018, Solicitar autenticación para realizar consultas, ya que solicita que los usuarios se autentifiquen en el sistema para realizar las consultas. Una de sus características más destacadas es que es de fácil aprendizaje por parte de los

usuarios y provee interfaces para trabajar con diferentes tipos de dispositivos móviles que utilizan sistemas operativos Android e iOS. En la figura 59 se muestran algunas de las fuentes de datos a las cuales es capaz de conectar la herramienta de *Tableau*.

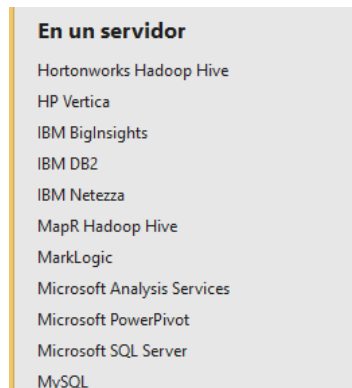


Figura 59. Fuentes de datos soportados por *Tableau*.

4.5.4. Modelamiento de pruebas

Las pruebas se diseñan para medir la funcionalidad del modelo con base en los requerimientos. Para tal finalidad se crea un plan de pruebas que asocia a cada uno de los requerimientos con su respectivo caso de pruebas, este se adjunta en el apéndice 5.

4.6. Construcción del prototipo

Con base en lo especificado en la etapa de diseño, se procede a construir el modelo en un gestor de base de datos *SQL Server*, el cual se utiliza para efectos de la experimentación y pruebas.

4.7. Evaluación del prototipo

Esta fase consta de dos etapas la ejecución de las pruebas y el análisis de resultados para evaluar al modelo propuesto.

4.7.1. Ejecución de pruebas

Se llevan a cabo las pruebas con los usuarios y se anotan en el documento plan de pruebas (adjunto en el apéndice 5), si el resultado de la prueba en específico cumple o no con el requerimiento solicitado.

4.7.2. Análisis de los resultados de las pruebas

Con base en los requerimientos se formuló el plan de pruebas y este demuestra la validez del modelo, el cual cumple en un 100% con la funcionalidad esperada, los resultados obtenidos en el documento de pruebas son tabulados como se muestra en la Tabla 44.

Tabla 44. Matriz de funcionalidad del modelo.

Funcionalidades a evaluar en el modelo					
ID	Funcionalidad	Valor %	Cantidad total de escenarios de pruebas	Cantidad de escenarios de prueba OK	Porcentaje de cumplimiento de funcionalidad
1	Permite identificar los objetos de diferentes bases de datos de manera integrada.	5,55	27	27	100
2	Permite identificar de forma única los conceptos del diccionario.	5,55	4	4	100
3	Permite agrupar conceptos del diccionario en jerarquías.	5,55	2	2	100
4	Permite asociar objetos con conceptos del diccionario.	5,55	1	1	100
5	Permite asociar objetos con	5,55	6	6	100

	propietarios.				
6	Permite asociar objetos con controles.	5,55	3	3	100
7	Permite asociar objetos observación de los propietarios.	5,55	1	1	100
8	Permite consultar información sobre los servidores remotos vinculados.	5,55	3	3	100
9	Permite registrar trazabilidad de los procesos del sistema.	5,55	1	1	100
10	Permite definir las bases de datos a las cuales realizar el proceso de extracción.	5,55	1	1	100
11	Permite definir los objetos participantes en el proceso de extracción.	5,55	1	1	100
12	Permite consultar objetos por conceptos definidos en el diccionario.	5,55	1	1	100
13	Permite consultar objetos por propietarios.	5,55	2	2	100
14	Permite consultar objetos por controles.	5,55	1	1	100
15	Permite consultar objetos por observaciones.	5,55	1	1	100
16	Permite escalabilidad en el versionamiento de gestores de bases de datos.	5,55	1	1	100
17	Permite escalabilidad funcional de forma modular.	5,55	4	4	100

18	Permite utilizar mecanismos de autenticación, para realizar las consultas.	5,55	1	1	100
----	--	------	---	---	-----

El valor porcentual de cada funcionalidad se calcula de la siguiente manera: 100 dividido entre la cantidad de funcionalidades por probar, lo que da como resultado $100/18 = 5,55\%$.

La fórmula para el cálculo del porcentaje de cumplimiento de funcionalidad es $(\text{cantidad de escenarios de prueba OK} * 100) / \text{cantidad de escenario de pruebas por requerimiento}$, posteriormente se realiza un promedio simple con la fórmula: $\text{suma de porcentajes de cumplimiento de funcionalidad} / (\text{cantidad de funcionalidades} * 100) * 100$. Para el caso en estudio el resultado obtenido es $(1800 / (18 * 100)) * 100 = 100\%$, por lo que se concluye que la solución cumple en un 100% con la funcionalidad esperada.

5. Capítulo V: Comparación de herramientas

5.1. Comparación de herramientas

Para la comparativa de las herramientas que permiten la administración de metadatos se utilizó como base un estudio realizado por *Forrester Research* en donde Peyret y Michele (2014) exponen un análisis sobre las principales herramientas existentes en el mercado. Las herramientas seleccionadas son *Informatica Metadata Manager and Bussiness Glosary*, *IBM InfoSphere Information Server* y *Adaptive Metadata Manager*, adicionalmente se comparan con la solución propuesta en este trabajo, el resultado de la comparativa se muestra en la Tabla 45².

Tabla 45. Comparativa de herramientas para la administración de metadatos.

ID	Característica	Informática	Adaptative	IBM	Modelo
1	Vinculación de metadatos técnicos con definiciones de negocio	SÍ	SÍ	SÍ	SÍ
2	Capacidad de búsquedas avanzadas y filtros	SÍ	SÍ	SÍ	SÍ
3	Generación de atributos extendidos	SÍ	SÍ	SÍ	NO
4	Personalización de vistas (en interfaz gráfica)	SÍ	SÍ	SÍ	NO
5	Visualización de las estructuras de datos	SÍ	SÍ	SÍ	NO
6	Permite consultar información sobre los orígenes de datos	SÍ	SÍ	SÍ	SÍ
7	Linaje de datos				
8	Permite el uso del linaje de datos	SÍ	SÍ	SÍ	NO
9	Generador de diagramas gráficos	SÍ	SÍ	SÍ	NO

² Se utiliza como referencia lo expuesto por *Informatica Corporation* (Informatica Corporation, 2013), *Adaptive* (Adaptive, 2015) e *IBM* (IBM, 2011).

10	Análisis de impacto				
11	Catálogo de metadatos integrado	SÍ	SÍ	SÍ	SÍ
12	Conectores de metadatos	SÍ	SÍ	SÍ	SÍ
13	Consulta de dependencias sobre el entorno integrado	SÍ	SÍ	SÍ	SÍ
14	Herramientas de auditoría				
15	Rastreo de cambios	SÍ	SÍ	SÍ	SÍ
16	Herramientas de colaboración				
17	Sistema de mensajería integrado	SÍ	NO	NO	NO
18	Anotaciones en los objetos	SÍ	SÍ	SÍ	SÍ
19	Notificación de cambios	SÍ	SÍ	SÍ	NO
20	Permite adjuntar documentos a las definiciones	SÍ	NO	NO	NO
21	Herramientas de administración de cambios				
22	Sistema de aprobaciones de cambios	SÍ	SÍ	SÍ	SÍ
23	Administración de cambios basados en roles	SÍ	SÍ	SÍ	NO

Fuente: elaboración propia del investigador.

A continuación, se detallan cada una de las características analizadas.

La vinculación de metadatos técnicos con las definiciones de negocio se refiere a la asociación de un objeto de la base de datos con un término de negocio expuesto en un diccionario empresarial; por ejemplo, una tabla nombrada CUM001 se puede relacionar con el concepto de negocio Cliente del diccionario; por lo tanto, un usuario de negocio puede determinar que el objeto CUM001 tiene información de clientes.

La capacidad de filtros avanzados se refiere a la capacidad con la cual las herramientas permiten a los usuarios realizar consultas complejas.

La generación de atributos extendidos contempla si las herramientas permiten a los usuarios a la creación de características adicionales que les permita describir a los objetos de la base de datos.

La personalización de vistas se refiere a la capacidad que las herramientas brindan para personalizar los reportes o la interfaz gráfica de estas.

La visualización de las estructuras de datos se refiere a la funcionalidad de las herramientas que permiten a los usuarios visualizar los metadatos acerca de las estructuras de datos, tales como su nombre, tipo de datos, longitud y precisión, por medio de una interfaz gráfica.

La consulta sobre información de los orígenes de datos es la funcionalidad que permite a los usuarios consultar información sobre los orígenes de datos; por ejemplo, un usuario puede requerir el conocer en cuál servidor físico se almacena cierto tipo de información.

El linaje de datos es la característica que permite conocer los procesos por los cuales los datos fluyen; por ejemplo, un analista desde el punto de vista técnico, desea conocer cómo un cambio en una estructura de datos puede afectar a los procesos de negocio (por ejemplo, en un reporte), y desde el punto de vista de negocio, el usuario de negocio puede determinar de dónde se originan los datos para generar sus reportes.

La generación de diagramas gráficos es la capacidad que tienen las herramientas de generar un mapeo gráfico de los objetos de bases de datos y sus orígenes.

El catálogo de metadatos integrado es la capacidad que poseen las herramientas para mantener en un solo repositorio la información de múltiples orígenes de datos de manera integrada (metadatos).

Los conectores de metadatos son la característica que permiten a las herramientas conectarse con múltiples orígenes de datos.

La consulta sobre el entorno integrado se refiere a la capacidad que poseen las herramientas para su configuración interna; por ejemplo, si se desea conocer cuáles son todos los servidores vinculados con una herramienta.

El rastreo de cambios es la capacidad de dar trazabilidad a los cambios en el sistema; por ejemplo, si un usuario modifica un término de negocio y se desea conocer quién y cuándo realizó el cambio.

Sistemas de mensajería integrada es la capacidad que tiene la herramienta si desde ésta se pueden enviar mensajes para que sean revisados por otros usuarios.

Anotaciones en los objetos se refiere a la capacidad que tienen las herramientas de que los usuarios realicen comentarios sobre los objetos de las bases de datos.

Notificación de cambios se refiere a la capacidad de informar a los usuarios propietarios de los objetos sobre cambios para su aprobación.

Adjuntar documentos se refiere a la capacidad de agregar documentos electrónicos en las definiciones de términos.

Sistemas de aprobación de cambios se refiere a la capacidad de aprobación de cambios de manera mancomunada.

Administración de cambios basados en roles se refiere a que los usuarios solo puedan ver y modificar en las herramientas los datos con base en su rol (nivel de autoridad).

5.2. Ventajas y desventajas del desarrollo local de la solución de *software*

A continuación, se exponen algunas de las ventajas y desventajas del desarrollo del *software* a lo interno de la organización.

Ventajas:

- El código fuente es de la empresa, por lo que puede ser modificado el *software* de acuerdo con sus necesidades.
- El *software* se puede utilizar en una cantidad ilimitada de equipos sin necesidad de pagar licencias por cada uno de ellos.
- No se requiere del soporte exclusivo por parte del fabricante propietario del *software*.

Desventajas:

- Las herramientas existentes en la industria tienen más experiencia en el mercado, por lo que presentan menos errores, los cuales han sido corregidos durante la vida del producto, en comparación con la solución propuesta que es más reciente.
- Es necesario generar la documentación.
- No se cuenta con el respaldo y asesoría de un tercero.

6. Capítulo VI: Conclusiones, recomendaciones y trabajo futuro

6.1. Conclusiones

En el siguiente proyecto se plantearon los siguientes objetivos:

Objetivo general

Diseñar una solución que permita la gobernabilidad sobre los objetos de bases de datos, la cual facilite clasificar a estos con base en sensibilidad o riesgo, identificar sus propietarios, los controles de seguridad implementados y, a su vez, determinar en cuáles plataformas y gestores se encuentran.

Objetivos específicos

- Construir un modelo de datos que permita identificar de manera centralizada, los objetos de bases de datos existentes en toda la organización, para gobernarse de manera conjunta.
- Crear procesos que permitan integrar los objetos de bases de datos residentes en diferentes plataformas y gestores.
- Proponer una herramienta que permita visualizar información sobre los objetos de bases de datos, residentes en diferentes plataformas y gestores, de forma integrada, la cual implemente algún mecanismo de autenticación.

El primer objetivo específico “construir un modelo de datos que permita identificar de manera centralizada los objetos de bases de datos existentes en toda la organización”, se logró por medio del modelado de datos, el cual sirvió de base para la posterior construcción del prototipo.

El segundo objetivo específico “crear procesos que permitan integrar los objetos de bases de datos residentes en diferentes plataformas y gestores”, se logró definiendo los procesos para la integración de los datos (extracción transformación y carga).

El tercer objetivo específico “proponer una herramienta que permita visualizar información sobre los objetos de bases de datos, residentes en diferentes plataformas y gestores, de forma integrada, la cual implemente algún mecanismo de autenticación”, se logró sugiriendo la herramienta tecnológica para la visualización de los datos y la definición de la arquitectura empleada en la solución para lograr este objetivo.

En cuanto al objetivo general, se logró al utilizar la metodología *rapid warehousing*, la cual facilitó la elaboración del diseño completo de la solución.

Se aporta diseño, de libre uso para la industria, que permite administrar los objetos de bases de datos, residentes en plataformas y gestores heterogéneos, de forma integrada, que permite relacionar metadatos técnicos con metadatos de negocio, a través de los conceptos definidos en el diccionario de datos empresarial y, adicionalmente, permite asociar los objetos de bases de datos con propietarios, controles de seguridad y comentarios.

Es posible lograr una gobernabilidad sobre los objetos de bases de datos a través de los conceptos definidos en el diccionario de datos empresarial.

La solución se puede utilizar como parte de un programa de gobernabilidad de datos, ya que este permite administrar el inventario de algunos activos de información, como son los objetos de bases de datos, servidores, plataformas y aplicaciones.

El diseño permite asignar responsables sobre los objetos de bases de datos, los cuales tendrán que custodiar los activos de información asignados bajo su cargo; en caso de que se presente un error, existe un responsable, al cual la organización puede solicitar retroalimentación.

La propuesta permite identificar, de manera oportuna, los administradores de servidores y bases de datos existentes en una organización. Esto es especialmente útil cuando se requieren realizar cambios en los computadores de producción, en el cual se necesita identificar a los involucrados.

Se facilita el trabajo colaborativo entre los propietarios y los administradores de la información, ya provee un mecanismo, para el control de cambios, sobre los objetos, en el cual un propietario solicita el cambio por medio de los comentarios que realiza sobre los objetos, y un aprobador lo analiza para responder si aplica o no.

El diseño es escalable, ya que permite agregar nuevas plataformas y gestores de bases de datos al sistema con el menor impacto posible en el cambio de los procesos, ya que el modelo de datos permite la reutilización de estos.

Se concluye con base en las pruebas realizadas, que se construye una solución que cumple con la funcionalidad solicitada por el cliente.

Con base en la comparación de herramientas existentes en la industria, se concluye que la solución propuesta no abarca todas las funcionalidades que estas ofrecen; sin embargo, contiene la funcionalidad suficiente para cumplir el objetivo para la cual fue construida.

Con los conocimientos adquiridos en la maestría de base de datos de Cenfotec, es posible la construcción de soluciones empresariales innovadoras y útiles para la industria.

La experiencia vivida durante la elaboración proyecto es muy enriquecedora, ya que se trabaja con profesores, compañeros de clase y profesionales de áreas afines, de los cuales quedan grandes aprendizajes y amistades. Se aplica una gran cantidad de conocimiento adquirido durante la carrera de forma práctica. Al igual que aquellos que brindaron su apoyo, el investigador espera apoyar a otros, para algún día ser digno de ser llamado maestro.

6.2. Recomendaciones

Se recomienda lo siguiente:

- Contar con el apoyo de la alta gerencia para implementar la solución propuesta, ya que se requiere involucrar la cultura organizacional para dar importancia a la clasificación de los objetos. Sin este factor, la implementación no será exitosa.

- Realizar capacitaciones sobre el uso de la solución, esta puede ser presencial o haciendo uso de tecnologías digitales como *wikimedia*, en la cual se pueden publicar, tutoriales y videos explicativos, entre otros.
- Se recomienda la elaboración de lineamientos, en los cuales se definan los roles y responsabilidades que tienen los propietarios y los administradores de información (los cuales pueden ser los aprobadores de cambios sobre los objetos de bases de datos).
- El modelo de datos relacional, expuesto en el almacén de datos operacional, puede ser construido en otro gestor de bases de datos, distinto a *SQL Server*; sin embargo, deben tomar en consideración que los conectores de datos no siempre son gratuitos y se puede incurrir en costos adicionales, como es el caso de *DB2*, en el cual es necesario adquirir componentes de *InfoSphere Federation Server*.

6.3. Trabajo futuro

Como trabajo futuro que queda por realizar está:

- Implementar el modelo propuesto en ambientes de producción.
- Evaluar el rendimiento de la solución en sistemas de producción con cargas reales de trabajo, en el cual los recursos del computador se encuentren utilizados por múltiples procesos.
- En la solución propuesta se crearon procedimientos almacenados por cada uno de los motores de bases de datos en estudio (*DB2*, *Oracle* y *SQL Server*); sin embargo, se pueden crear más procedimientos o funciones que permitan extender la funcionalidad que se proporciona en lo descrito en este documento.
- Se puede agregar el soporte a otras bases de datos como *Postgresql* y *Mysql* para mencionar algunas.
- La solución no tiene todas las características que ofrecen herramientas similares en el mercado, por lo que puede incorporarse en futuras investigaciones.

Referencias

- Abraham S., Henry F. y S. Sudarshan. (2002). *Fundamentos de bases de datos*. (Cuarta Edición ed.). McGraw Hill.
- Adamson, C. (2010). *The complete reference Star Schema*.
- Adaptative. (2015). *How is Metadata Manager different?* Retrieved 01 01 2015, from <http://www.adaptive.com/products/metadata-manager/how-is-metadata-manager-different/>
- Amazon. (2015). *Samsung 850 EVO 1TB SSD*. Retrieved 01 01 2015, from <http://www.amazon.com/Samsung-2-5-Inch-Internal-MZ-75E1T0B-AM/dp/B00OBRFFAS>
- ANSI/ISO/IEC. (1999). *Database Lenguaje SQL - Part 2 (SQL/Foundation)*.
- BAC Credomatic . (2014). Historia del Grupo BAC | Credomatic. Retrieved 08 24 2014, from <https://www.bac.net/regional/esp/banco/acerca.html>
- BAC Credomatic. (2013). *Visión, Misión y Valores*. Retrieved 08 24 2014, from <https://www.bac.net/regional/esp/banco/empresa.html>
- BAC Credomatic. (2013b). *Estados financieros 2013*. Retrieved 11 01 2014, from <https://www.bac.net/bacsanjose/esp/banco/nuestraemp/nemestadosfinancieros2.html>
- Barrios, L. E. (2009). *Sistemas de bases de datos federadas*. Retrieved 11 25 2014, from <http://eolo.cps.unizar.es/docencia/MasterUPV/Articulos/Sistemas%20de%20Bases%20de%20Datos%20Federadas.pdf>
- Biolchini J., Gomes Mian P., Cruz Natal, A. y Horta Travassos G. (2005). *Systematic Review in Software Engineering*. Retrieved 10 01 2014, from <http://www.cin.ufpe.br/~in1037/leitura/systematicReviewSE-COPPE.pdf>

- Dario, B. R. (2009). *Data Warehousing: Investigación y Sistematización de Conceptos*. Retrieved 01 01 2015, from <http://www.dataprix.com/datawarehouse-manager>
- Data Governance Institute. (2015). *Definitions of data governance*. Retrieved 02 01 2015, from http://www.datagovernance.com/adg_data_governance_definition/
- Engelhart M. D., Hill W. H., Furst E. J., Krathwohl D. R. y Bloom B. S. (1956). *Taxonomy of educational objectives: The classification of educational goals*.
- Espinoza, R. (2010). *Kimball vs Inmon. Ampliación de conceptos del Modelado Dimensional*. Retrieved 12 01 2014, from <https://churriwifi.wordpress.com/2010/04/19/15-2-ampliacion-conceptos-del-modelado-dimensional/>
- Gartner Inc. (2014). *Magic Quadrant for Data Integration Tools*. Retrieved 09 10 2014, from <http://www.gartner.com/technology/reprints.do?id=1-1YAXV15&ct=140728&st=sb>
- Gartner Inc. (2015). *Magic Quadrant for Business Intelligence and Analytics Platforms*. Retrieved 02 24 2015, from <http://www.gartner.com/technology/reprints.do?id=1-2ACLP1P&ct=150220&st=sb>
- González Alvarado C. (2009). *Sistemas de bases de datos*. Editorial Tecnológica de Costa Rica.
- Gruber, T. (1993). *A translating approach to portable ontology specifications*. Retrieved 08 02 2014, from <http://www.dbis.informatik.hu-berlin.de/dbisold/lehre/WS0203/SemWeb/lit/KSL-92-17.pdf>
- Hurtado de Barrera, J. (2005). *Como formular objetivos de investigación: Un acercamiento desde la investigación Holística*. Fundación Sypal.
- IBM. (2001). *Consulta de SQL versión 7*. Retrieved 01 01 2015, from <ftp://ftp.software.ibm.com/ps/products/db2/info/vr7/pdf/letter/nlv/db2s0z71.pdf>

- IBM. (2002). *DB2 Universal Data Base - Consulta de SQL Volumen 1*. Retrieved 11 01 2014, from <ftp://public.dhe.ibm.com/ps/products/db2/info/vr8/pdf/letter/nlv/db2s1z80.pdf>
- IBM. (2011). *Metadata Managment with IBM InfoSphere Information Server*. Retrieved 02 01 2015, from <http://www.redbooks.ibm.com/redbooks/pdfs/sg247939.pdf>
- Informatica Corporation. (2013). *Metadata Management for Holistic Data Governance Using Informatica Metadata Manager & Business Glossary to Govern Your Data and Deliver Better Business Decisions*. Retrieved 01 01 2013, from https://www.informatica.com/content/dam/informatica-com/global/amer/us/collateral/white-paper/metadata-management-data-governance_white-paper_2163.pdf
- Inmon, W. (1992). *Building the Data Warehouse*. John Wiley & Sons Inc.
- Inmon, W. H. (2005). *Building the Data WareHouse*. Wiley Publishing, Inc.
- Intricity. (2014). *What is Metadata Management?* Retrieved 11 10 2014, from <https://www.youtube.com/watch?v=Ioxr2mcOLjE>
- IT Governace Institute. (2007). *Cobit 4.1*.
- ITIL V3. (2014). *RACI - Gestión de servicios de IT*. Retrieved 11 10 2014, from http://itilv3.osiatis.es/disenio_servicios_TI/modelo_RACI.php
- Jacobs, J. (2003). *The Rap on RUP: An Introduction to the Rational Unified Process*. Retrieved 11 10 2014, from <http://www.jeffreyjacobs.com/JJ%20Presents/RaponRupSPINforWeb.pdf>
- Kimball, R. (1998). *Is Data Staging Relational?* Retrieved 01 01 2015, from <http://www.kimballgroup.com/1998/04/is-data-staging-relational/>
- Kriegel, A. (2011). *Discovering SQL a Hands-on Guide for Beginers*. Wiley Publishing Inc.

- MDM Institute. (2014). *What is Data Governance?* Retrieved from <http://0046c64.netsolhost.com/whatIsDataGovernance.html>
- Microsoft. (2014). *Catalog Views (Transact-SQL)*. Retrieved 11 01 2014, from <http://msdn.microsoft.com/en-us/library/ms174365.aspx>
- Ministerio de Trabajo y Seguridad Social de Costa Rica. (2014). *Fijación de salarios mínimos para el sector privado que regirán a partir del primero de enero del 2015 - Decreto Ejecutivo 38728*. Retrieved 01 01 2015, from <https://docs.google.com/viewer?a=v&pid=sites&srcid=bXRzcy5nby5jcnxtdHNzfGd4OjUwYzZlZDU3MmVkZmJlYjk>
- National Information Standards Organization. (2015). *Understanding Metadata*. Retrieved 10 01 2014, from <http://www.niso.org/publications/press/UnderstandingMetadata.pdf>
- Newegg. (2015). *SAMSUNG 850 EVO SSD*. Retrieved 01 01 2015, from http://www.newegg.com/Product/Product.aspx?Item=9SIA29P2TN6084&cm_re=1tb_ssd_-_20-147-374_-_Product
- OGC. (2009). *ITIL V3*.
- Oracle. (2002). *Database Architecture: Federated vs. Clustered*. Retrieved 25 11 2014, from <http://www.oracle.com/technetwork/database/windows/clustercomp-134873.pdf>
- Oracle. (2007). *A Data Mart Concepts*. Retrieved 12 01 2014, from http://docs.oracle.com/html/E10312_01/dm_concepts.htm#CIHFHBJD
- Oracle. (2014). *Database Concept: The Data Dictionary*. Retrieved 11 01 2014, from Oracle Help Center: http://docs.oracle.com/cd/B28359_01/server.111/b28318/datadict.htm#CNCPT002

- Oracle. (2014b). *TimesTen in-Memory Database - System Tables and View Reference*. Retrieved 11 01 2014, from http://docs.oracle.com/cd/E21901_01/timesten.1122/e21644.pdf
- PCI Security Standards Council. (2013). *PCI - Normas de seguridad de datos*. Retrieved 01 01 2015, from <https://es.pcisecuritystandards.org/minisite/en/>
- Peyret H. y Michele, G. (2014). *The Forrester Wave: Data Governance Tools, Q2 2014*. Forrester Research. Retrieved from http://www.sas.com/content/dam/SAS/en_us/doc/analystreport/forrester-wave-data-governance-107200.pdf
- Saltor, F. (2014). *Sistemas de bases de datos federadas*. Retrieved 11 25 2014, from http://www.tamps.cinvestav.mx/~vjsosa/clases/tsbd/BD_Federadas.pdf
- Sánchez Zambrano D. y Tarapuez Roa J. (2010). *Evaluación de los costos de implementación del software en una organización*. Retrieved 06 01 2015, from <http://fce.unal.edu.co/wiki/images/7/7e/costosdeimplementaciondelsoftwarelibre.pdf>
- SAS Institute. (2002). *SAS Rapid Warehousing Methodology 4.1*.
- U.S Securities and Exchange Commissions. (2002). *Ley Sarbanes-Oxley*. Retrieved 01 01 2015, from <https://www.sec.gov/about/laws/soa2002.pdf>

Apéndices

Apéndice 1 - Especificación de requerimientos del sistema	197
Apéndice 2. Diagrama de datos a nivel conceptual del diseño relacional.....	210
Apéndice 3. Diagrama de datos a nivel físico del diseño relacional.	210
Apéndice 4. Consultas distribuidas para DB2, <i>Oracle</i> y <i>SQL Server</i>	211
Apéndice 5. Plan de pruebas.	211
Apéndice 6. Diagrama de datos a nivel conceptual del diseño multidimensional.	229
Apéndice 7. Diagrama de datos a nivel físico del diseño multidimensional.....	229
Apéndice 8. Código fuente.....	230
Apéndice 9. <i>Scripts</i> de pruebas <i>SQL</i>	230

Apéndice 1 - Especificación de requerimientos del sistema

Especificación de Requerimientos del Sistema (Versión 1)

Número de proyecto	CENFO-001				
Líder de proyecto	Marco Hernández Vásquez				
Fecha de documento	01/10/2014				
Histórico de revisiones					
Fecha	Forma	Fondo	Revisado por	Rol	Comentario
01/10/2014	x	x	Marco Hernández Vásquez	Arquitecto de datos.	N/A

1. Objetivos

Identificador	O-001
Fecha	01/10/2014
Solicitante	Marco Hernández Vásquez
Descripción	Permitir la gobernabilidad de estructuras de objetos de base de datos contenidas en distintos gestores de base de datos y en plataformas heterogéneas.
Comentarios	N/A

2. Lista de requerimientos funcionales

Identificador: R-001	Identificar los objetos de diferentes bases de datos de manera integrada
Fecha de creación	01/10/2014
Versión	1
Fecha de modificación	01/10/2014
Solicitante	Marco Hernández Vásquez
Objetivos asociados	O-001
Descripción	<p>El sistema debe permitir identificar, de forma explícita y directa a cada uno de los objetos de base de datos, en múltiples gestores y múltiples plataformas.</p> <p>Los tipos de objetos de sistema que deben ser integrados son los siguientes:</p> <ul style="list-style-type: none"> • Catálogo • Tablas • Columnas • Vistas • Índices • Secuencias • Disparadores • Funciones • Procedimientos almacenados
Importancia	Alta
Comentarios	N/A

Identificador: R-002	Conceptos del diccionario
Fecha de creación	01/10/2014
Versión	1
Fecha de modificación	01/10/2014
Solicitante	Marco Hernández Vásquez
Objetivos asociados	O-001
Descripción	<p>El sistema debe permitir una serie de conceptos, los cuales determinarán el contexto de cada objeto de base de datos.</p> <p>Cada objeto deberá estar asignado a un determinado concepto, de tal manera que se pueda acceder de forma sencilla y global, todos los objetos pertenecientes a un concepto determinado, inclusive sin importar su gestor o plataforma.</p> <p>Un ejemplo de concepto es la clasificación de objetos, de tal manera que se puedan identificar los objetos “sensibles” de la compañía.</p>
Importancia	Alta
Comentarios	N/A

Identificador: R-003	Agrupar conceptos del diccionario en jerarquías
Fecha de creación	01/10/2014
Versión	1
Fecha de modificación	01/10/2014
Solicitante	Marco Hernández Vásquez

Objetivos asociados	O-001
Descripción	<p>El sistema debe permitir la agrupación de conceptos por medio de jerarquías.</p> <p>Por ejemplo:</p> <pre> graph TD D[Diccionario] --- S[Seguridad] D --- A[Aplicaciones] D --- Des[Descripción] S --- C[Cifrado] S --- Sen[Sensible] S --- Con[Confidencial] A --- P[Prestamos] A --- R[Riesgo] A --- AN[Aplicación N..] Des --- DE1[Descripción extendida 1] Des --- DEN[Descripción extendida N..] </pre>
Importancia	Alta
Comentarios	N/A

Identificador: R-004	Asociar objetos con conceptos del diccionario de datos
Fecha de creación	01/10/2014
Versión	1
Fecha de modificación	01/10/2014

Solicitante	Marco Hernández Vásquez
Objetivos asociados	O-001
Descripción	El sistema debe permitir la asignación de uno o varios conceptos a un mismo objeto, sin importar el gestor de base de datos al que pertenece, ni la plataforma sobre la cual reside.
Importancia	Alta
Comentarios	N/A

Identificador: R-005	Asociar los objetos con propietarios
Fecha de creación	01/10/2014
Versión	1
Fecha de modificación	01/10/2014
Solicitante	Marco Hernández Vásquez
Objetivos asociados	O-001
Descripción	<p>El sistema debe permitir la asociación de objetos de base de datos (en distintos gestores y distintas plataformas) con propietarios.</p> <p>Se debe contar con un repositorio de propietarios, los cuales son colaboradores de la empresa. Además, cada propietario puede tener uno o más delegados, los cuales ejercerán tareas administrativas sobre los objetos pertenecientes a su propietario.</p> <p>Las reglas a seguir son:</p> <ul style="list-style-type: none"> • Todo objeto de base de datos posee un propietario. • Un objeto de base de datos debe estar asociado únicamente a un propietario titular. • Un mismo propietario puede tener bajo su responsabilidad

	<p>más de un objeto.</p> <ul style="list-style-type: none"> • Un propietario puede tener uno o más delegados, quienes administrarán los objetos bajo su responsabilidad.
Importancia	Alta
Comentarios	N/A

Identificador: R-006	Asociar los objetos con controles
Fecha de creación	01/10/2014
Versión	1
Fecha de modificación	01/10/2014
Solicitante	Marco Hernández Vásquez
Objetivos asociados	O-001
Descripción	<p>El sistema debe permitir la asociación de controles de seguridad a objetos de base de datos con una determinada clasificación.</p> <p>Se debe diseñar de tal manera que sea posible identificar de forma explícita todos los controles de un objeto, pero además, todos los objetos sobre los cuales se está aplicando un determinado control.</p>
Importancia	Alta
Comentarios	N/A

Identificador: R-007	Asociar los objetos con observaciones de sus propietarios.
Fecha de creación	01/10/2014
Versión	1
Fecha de modificación	01/10/2014

Solicitante	Marco Hernández Vásquez
Objetivos asociados	O-001
Descripción	<p>El sistema debe permitir la inclusión de observaciones que los propietarios de cada uno de los objetos de su responsabilidad puedan agregar.</p> <p>Por ejemplo, un usuario puede comentar que un campo es sensible, para que posteriormente sea revisado por personal de seguridad y lo tomen en consideración.</p>
Importancia	Alta
Comentarios	N/A

Identificador: R-008	Consultar información sobre los servidores remotos vinculados
Fecha de creación	01/10/2014
Versión	1
Fecha de modificación	01/10/2014
Solicitante	Marco Hernández Vásquez
Objetivos asociados	O-001
Descripción	El sistema debe permitir consultar sobre datos particulares de cada gestor de base de datos, de cada plataforma y sus posibles combinaciones.
Importancia	Alta
Comentarios	N/A

Identificador: R-09	Registrar trazabilidad de los procesos del sistema
---------------------	--

Fecha de creación	01/10/2014
Versión	1
Fecha de modificación	01/10/2014
Solicitante	Marco Hernández Vásquez
Objetivos asociados	O-001
Descripción	El sistema debe registrar en una bitácora el resultado de la ejecución de los procesos que utilice, debe incluir el estado de la ejecución del proceso y mensajes de error con la finalidad de dar trazabilidad.
Importancia	Alta
Comentarios	N/A

Identificador: R-010	Definir las bases de datos a las cuales realizar la extracción de datos
Fecha de creación	01/10/2014
Versión	1
Fecha de modificación	01/10/2014
Solicitante	Marco Hernández Vásquez
Objetivos asociados	O-001
Descripción	El sistema debe permitir enlistar todos los posibles gestores de bases de datos y sus respectivas plataformas disponibles; además, permitir parametrizar la inclusión o exclusión de alguno de ellos dentro del sistema. Debe incluir la correspondiente versión del gestor.
Importancia	Alta
Comentarios	N/A

Identificador: R-011	Definir objetos participantes
Fecha de creación	01/10/2014
Versión	1
Fecha de modificación	01/10/2014
Solicitante	Marco Hernández Vásquez
Objetivos asociados	O-001
Descripción	El sistema debe permitir enlistar todos los objetos ubicados en un gestor de una plataforma específica, y permitir incluir o excluir a los objetos del aplicativo.
Importancia	Alta
Comentarios	N/A

Identificador: R-012	Consultar objetos por conceptos del diccionario
Fecha de creación	01/10/2014
Versión	1
Fecha de modificación	01/10/2014
Solicitante	Marco Hernández Vásquez
Objetivos asociados	O-001
Descripción	El sistema debe permitir a los usuarios, consultar los objetos asociados a un concepto en particular, así como todos los conceptos asignados un objeto determinado.
Importancia	Alta
Comentarios	N/A

Identificador: R-013	Consultar objetos por propietarios
Fecha de creación	01/10/2014
Versión	1
Fecha de modificación	01/10/2014
Solicitante	Marco Hernández Vásquez
Objetivos asociados	O-001
Descripción	El sistema debe permitir a los usuarios, consultar el propietario y los delegados asignados a un objeto en particular, así como también todos los objetos asignados a un propietario en específico.
Importancia	Alta
Comentarios	N/A

Identificador: R-014	Consultar objetos por controles
Fecha de creación	01/10/2014
Versión	1
Fecha de modificación	01/10/2014
Solicitante	Marco Hernández Vásquez
Objetivos asociados	O-001
Descripción	El sistema debe permitir a los usuarios, consultar todos los controles asignados a un objeto en particular, así como también se podrá consultar todos los objetos a los cuales tienen asignado un control específico.
Importancia	Alta
Comentarios	N/A

Identificador: R-015	Consultar las observaciones de los objetos
Fecha de creación	01/10/2014
Versión	1
Fecha de modificación	01/10/2014
Solicitante	Marco Hernández Vásquez
Objetivos asociados	O-001
Descripción	El sistema debe permitir a los usuarios, consultar las observaciones realizadas a los objetos.
Importancia	Alta
Comentarios	N/A

Identificador: R-016	Escalabilidad en versionamiento de gestores
Fecha de creación	01/10/2014
Versión	1
Fecha de modificación	01/10/2014
Solicitante	Marco Hernández Vásquez
Objetivos asociados	O-001
Descripción	El sistema debe permitir una alta escalabilidad en lo que a versiones del gestor de base de datos se refiere. En otras palabras, que el cambio de versión del gestor de base de datos implique el mínimo esfuerzo y su impacto sea transparente al aplicativo.
Importancia	Alta

Comentarios	N/A
-------------	-----

Identificador: R-017	Escalabilidad
Fecha de creación	01/10/2014
Versión	1
Fecha de modificación	01/10/2014
Solicitante	Marco Hernández Vásquez
Objetivos asociados	O-001
Descripción	El sistema debe permitir una alta escalabilidad en su funcionalidad; por tanto, debe ser diseñado e implementado con las mejores prácticas y los patrones más adecuados, de tal manera que permita incrementar funcionalidad de una manera modularizable.
Importancia	Alta
Comentarios	N/A

Identificador: R-018	Solicitar autenticación para realizar consultas
Fecha de creación	01/10/2014
Versión	1
Fecha de modificación	01/10/2014
Solicitante	Marco Hernández Vásquez
Objetivos asociados	O-001
Descripción	El sistema debe solicitar al usuario que se autentique para poder realizar las consultas del sistema.

Importancia	Alta
Comentarios	N/A

3. Información adicional

¿Existe algo en el sistema que realice algo similar a lo solicitado en alguno de los requerimientos especificados?	No
--	----

4. Glosario de Términos

Término	Descripción
Metadatos	Se refiere específicamente a los objetos que se encuentran en los diccionarios de los sistemas de bases de datos.
Gestor	<i>Software</i> administrador de la base de datos. Ej. <i>SQL/Server</i> , <i>DB2</i> , etc.
Plataforma	Sistema Operativo sobre el cual trabaja el gestor. Ej. <i>Windows</i> , <i>AS/400</i> , etc.

5. Oportunidades de Mejora

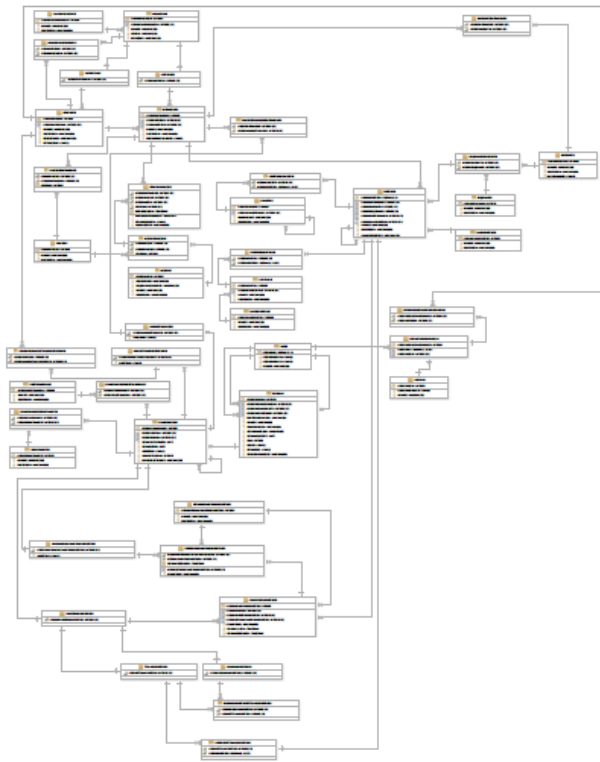
Secuencia	Fecha de ingreso	Prioridad	Persona que ingresa
Descripción:			
Responsable de atender	Fecha de atención	Observaciones del responsable	

Apéndice 2. Diagrama de datos a nivel conceptual del diseño relacional.



Este diseño se adjunta en formato digital.

Apéndice 3. Diagrama de datos a nivel físico del diseño relacional.



Este diseño se adjunta en formato digital.

Apéndice 4. Consultas distribuidas para DB2, Oracle y SQL Server.

Estas se adjuntan en formato digital.

Apéndice 5. Plan de pruebas.

Plan de pruebas

Nombre del Caso de Prueba	CP001 - Poblar catálogo integrado de objetos para DB2.			
Número y nombre del requerimiento	R-001 - Identificar los objetos de diferentes bases de datos de manera integrada.			
Objetivo	<i>Poblar el catálogo de objetos, utilizando el procedimiento almacenado usp_ejecutarProcesosPorInstancia</i>			
Escenario	Descripción	Resultado Esperado	Resultado	Prioridad
	1 El sistema debe consultar los objetos de tipo base de datos y almacenarlos en un catálogo de objetos.	Se registra el objeto en la tabla objetosBd.	OK	Alta
	2 El sistema debe consultar los objetos de tipo columna y almacenarlos en un catálogo de objetos.	Se registra el objeto en la tabla objetosBd.	OK	Alta
	3 El sistema debe consultar los objetos de tipo disparador y almacenarlos en un catálogo de objetos.	Se registra el objeto en la tabla objetosBd.	OK	Alta
	4 El sistema debe consultar los objetos de tipo función y almacenarlos en un	Se registra el objeto en la tabla objetosBd.	OK	Alta

	catálogo de objetos.			
5	El sistema debe consultar los objetos de tipo índice y almacenarlos en un catálogo de objetos.	Se registra el objeto en la tabla objetosBd.	OK	Alta
6	El sistema debe consultar los objetos de tipo procedimiento almacenado y almacenarlos en un catálogo de objetos.	Se registra el objeto en la tabla objetosBd.	OK	Alta
7	El sistema debe consultar los objetos de tipo secuencia y almacenarlos en un catálogo de objetos.	Se registra el objeto en la tabla objetosBd.	OK	Alta
8	El sistema debe consultar los objetos de tipo tabla y almacenarlos en un catálogo de objetos	Se registra el objeto en la tabla objetosBd.	OK	Alta
9	El sistema debe consultar los objetos de tipo vista y almacenarlos en un catálogo de objetos.	Se registra el objeto en la tabla objetosBd.	OK	Alta
Comentarios	N/A			

Nombre del Caso de Prueba	CP002 - Poblar catálogo integrado de objetos para <i>Oracle</i> .			
Número y nombre del requerimiento	R-001 - Identificar los objetos de diferentes bases de datos de manera integrada.			
Objetivo	<i>Poblar el catálogo de objetos, utilizando el procedimiento almacenado usp_ejecutarProcesosPorInstancia</i>			
Escenario	Descripción	Resultado Esperado	Resultado	Prioridad
1	El sistema debe consultar los objetos de tipo base de datos y	Se registra el objeto en la tabla	OK	Alta

	almacenarlos en un catálogo de objetos.	objetosBd.		
2	El sistema debe consultar los objetos de tipo columna y almacenarlos en un catálogo de objetos.	Se registra el objeto en la tabla objetosBd.	OK	Alta
3	El sistema debe consultar los objetos de tipo disparador y almacenarlos en un catálogo de objetos.	Se registra el objeto en la tabla objetosBd.	OK	Alta
4	El sistema debe consultar los objetos de tipo función y almacenarlos en un catálogo de objetos.	Se registra el objeto en la tabla objetosBd.	OK	Alta
5	El sistema debe consultar los objetos de tipo índice y almacenarlos en un catálogo de objetos.	Se registra el objeto en la tabla objetosBd.	OK	Alta
6	El sistema debe consultar los objetos de tipo procedimiento almacenado y almacenarlos en un catálogo de objetos.	Se registra el objeto en la tabla objetosBd.	OK	Alta
7	El sistema debe consultar los objetos de tipo secuencia y almacenarlos en un catálogo de objetos.	Se registra el objeto en la tabla objetosBd.	OK	Alta
8	El sistema debe consultar los objetos de tipo tabla y almacenarlos en un catálogo de objetos.	Se registra el objeto en la tabla objetosBd.	OK	Alta
9	El sistema debe consultar los objetos de tipo vista y almacenarlos en un catálogo de objetos.	Se registra el objeto en la tabla objetosBd.	OK	Alta
Comentarios	N/A			

Nombre del Caso de Prueba	CP003 - Poblar catálogo integrado de objetos para <i>SQL Server</i> .			
Número y nombre del requerimiento	R-001 - Identificar los objetos de diferentes bases de datos de manera integrada.			
Objetivo	<i>Poblar el catálogo de objetos, utilizando el procedimiento almacenado usp_ejecutarProcesosPorInstancia</i>			
Escenario	Descripción	Resultado Esperado	Resultado	Prioridad
	1 El sistema debe consultar los objetos de tipo base de datos y almacenarlos en un catálogo de objetos.	Se registra el objeto en la tabla objetosBd.	OK	Alta
	2 El sistema debe consultar los objetos de tipo columna y almacenarlos en un catálogo de objetos.	Se registra el objeto en la tabla objetosBd.	OK	Alta
	3 El sistema debe consultar los objetos de tipo disparador y almacenarlos en un catálogo de objetos.	Se registra el objeto en la tabla objetosBd.	OK	Alta
	4 El sistema debe consultar los objetos de tipo función y almacenarlos en un catálogo de objetos.	Se registra el objeto en la tabla objetosBd.	OK	Alta
	5 El sistema debe consultar los objetos de tipo índice y almacenarlos en un catálogo de objetos.	Se registra el objeto en la tabla objetosBd.	OK	Alta
	6 El sistema debe consultar los objetos de tipo procedimiento almacenado y almacenarlos en un catálogo de objetos.	Se registra el objeto en la tabla objetosBd.	OK	Alta
	7 El sistema debe consultar los objetos de tipo secuencia y almacenarlos en	Se registra el objeto en la tabla	OK	Alta

	un catálogo de objetos.	objetosBd.		
	8 El sistema debe consultar los objetos de tipo tabla y almacenarlos en un catálogo de objetos.	Se registra el objeto en la tabla objetosBd.	OK	Alta
	9 El sistema debe consultar los objetos de tipo vista y almacenarlos en un catálogo de objetos.	Se registra el objeto en la tabla objetosBd.	OK	Alta
Comentarios	N/A			

Nombre del Caso de Prueba	CP004 - Crear conceptos en diccionario.			
Número y nombre del requerimiento	R-002 - Identificar de forma única los conceptos del diccionario.			
Objetivo	<i>Crear conceptos en diccionario</i>			
Escenario	Descripción	Resultado Esperado	Resultado	Prioridad
	<p>1 El actor ingresa una palabra clave y su descripción en el diccionario, utilizando el siguiente instrucción:</p> <pre> SET IDENTITY_INSERT diccionario.conceptos ON; INSERT INTO diccionario.conceptos (codigoConcepto, codigoConceptoPadre, palabraClave, descripcion) VALUES (1, null, 'Palabra01', 'Descripción01'); SET IDENTITY_INSERT </pre>	El sistema registra la palabra clave y su descripción en la tabla conceptos, adicionalmente genera un identificador único numérico en la tabla conceptos.	OK	Alta

	<code>diccionario.conceptos OFF;</code>			
2	<p>El actor ingresa palabras clave duplicadas, utilizando la siguiente instrucción:</p> <pre>SET IDENTITY_INSERT diccionario.conceptos ON; INSERT INTO diccionario.conceptos (codigoConcepto, codigoConceptoPadre, palabraClave, descripcion) VALUES (1, null, 'Palabra01', 'Descripción01'); SET IDENTITY_INSERT diccionario.conceptos OFF;</pre>	<p>El sistema no debe permitir la inclusión de palabras claves duplicadas en la tabla conceptos.</p>	OK	Alta
3	<p>El actor modifica la palabra clave, utilizando el siguiente comando:</p> <pre>update diccionario.conceptos set palabraClave = 'Palabra02' where palabraClave = 'Palabra01';</pre>	<p>El sistema registra los cambios sin afectar el identificador único asignado a la palabra clave en la tabla conceptos.</p>	OK	Alta
4	<p>El actor modifica la descripción, utilizando el siguiente comando:</p> <pre>update diccionario.conceptos set descripcion = 'Descripción02' where descripcion = 'Descripción01';</pre>	<p>El sistema registra los cambios sin afectar el identificador único asignado a la palabra clave en la tabla concepto.</p>	OK	Alta

Comentarios	N/A
--------------------	-----

Nombre del Caso de Prueba	CP005 - Agrupar conceptos del diccionario en jerarquías.
Número y nombre del requerimiento	R-003 - Agrupar conceptos del diccionario en jerarquías.
Objetivo	<i>Agrupar conceptos de forma jerárquica</i>

Escenario	Descripción	Resultado Esperado	Resultado	Prioridad																														
	<p>1 El actor ingresa los conceptos de forma que se genera la siguiente jerarquía, en el orden que se muestra en la tabla del resultado esperado.</p> <pre> graph TD Diccionario --- Seguridad Diccionario --- Aplicaciones Seguridad --- Cifrado Seguridad --- Sensible Seguridad --- Confidencial Aplicaciones --- Préstamos Aplicaciones --- Riesgo Aplicaciones --- Seguros </pre> <p>Utiliza la siguiente sentencia:</p> <pre> delete from diccionario.conceptos; SET IDENTITY_INSERT diccionario.conceptos ON; </pre>	<p>El sistema debe almacenar los datos de manera:</p> <table border="1"> <thead> <tr> <th>IDCONCEPTO</th> <th>IDCONCEPTOPADRE</th> <th>PALABRACLAVE</th> </tr> </thead> <tbody> <tr><td>1</td><td>NULL</td><td>Diccionario</td></tr> <tr><td>2</td><td>1</td><td>Aplicaciones</td></tr> <tr><td>3</td><td>2</td><td>Préstamo</td></tr> <tr><td>4</td><td>2</td><td>Riesgo</td></tr> <tr><td>5</td><td>2</td><td>Seguros</td></tr> <tr><td>6</td><td>1</td><td>Seguridad</td></tr> <tr><td>7</td><td>6</td><td>Cifrado</td></tr> <tr><td>8</td><td>6</td><td>Sensible</td></tr> <tr><td>9</td><td>6</td><td>Confidencial</td></tr> </tbody> </table>	IDCONCEPTO	IDCONCEPTOPADRE	PALABRACLAVE	1	NULL	Diccionario	2	1	Aplicaciones	3	2	Préstamo	4	2	Riesgo	5	2	Seguros	6	1	Seguridad	7	6	Cifrado	8	6	Sensible	9	6	Confidencial	OK	Alta
IDCONCEPTO	IDCONCEPTOPADRE	PALABRACLAVE																																
1	NULL	Diccionario																																
2	1	Aplicaciones																																
3	2	Préstamo																																
4	2	Riesgo																																
5	2	Seguros																																
6	1	Seguridad																																
7	6	Cifrado																																
8	6	Sensible																																
9	6	Confidencial																																

```
INSERT INTO
diccionario.conceptos
(codigoConcepto,
codigoConceptoPadre,
palabraClave, descripcion)
VALUES (1, null, 'Diccionario',
'Descripción01');
INSERT INTO
diccionario.conceptos
(codigoConcepto,
codigoConceptoPadre,
palabraClave, descripcion)
VALUES (2, 1, 'Aplicaciones',
'Descripción01');
INSERT INTO
diccionario.conceptos
(codigoConcepto,
codigoConceptoPadre,
palabraClave, descripcion)
VALUES (3, 2, 'Préstamo',
'Descripción01');
INSERT INTO
diccionario.conceptos
(codigoConcepto,
codigoConceptoPadre,
palabraClave, descripcion)
VALUES (4, 2, 'Riesgo',
'Descripción01');
INSERT INTO
diccionario.conceptos
(codigoConcepto,
codigoConceptoPadre,
palabraClave, descripcion)
VALUES (5, 2, 'Seguros',
'Descripción01');
INSERT INTO
diccionario.conceptos
```

```

(codigoConcepto,
codigoConceptoPadre,
palabraClave, descripcion)
VALUES (6, 1, 'Seguridad',
'Descripción01');
INSERT INTO
diccionario.conceptos
(codigoConcepto,
codigoConceptoPadre,
palabraClave, descripcion)
VALUES (7, 6, 'Cifrado',
'Descripción01');
INSERT INTO
diccionario.conceptos
(codigoConcepto,
codigoConceptoPadre,
palabraClave, descripcion)
VALUES (8, 6, 'Sensible',
'Descripción01');
INSERT INTO
diccionario.conceptos
(codigoConcepto,
codigoConceptoPadre,
palabraClave, descripcion)
VALUES (9, 6, 'Confidencial',
'Descripción01');
SET IDENTITY_INSERT
diccionario.conceptos OFF;

```

2 El usuario reorganiza la jerarquía de la siguiente manera:

El sistema debe almacenar los datos de manera:

Alta

IDCONCEPTO	IDCONCEPTOPADRE	PALABRACLAVE
1	NULL	Diccionario
2	1	Aplicaciones
3	2	Préstamo
4	2	Riesgo
5	2	Seguros
6	2	Seguridad
7	6	Cifrado
8	6	Sensible
9	6	Confidencial

	<pre> graph TD D[Diccionario] --- A[Aplicaciones] A --- P[Préstamos] A --- R[Riesgo] A --- S[Seguros] A --- Se[Seguridad] Se --- C[Cifrado] Se --- Sns[Sensible] Se --- Con[Confidencial] </pre> <p>Utiliza la siguiente sentencia:</p> <pre> update diccionario.conceptos set codigoConceptoPadre = '2' where codigoConcepto = 6; </pre>		
Comentarios	N/A		

Nombre del Caso de Prueba	CP006 - Crear asociaciones de los objetos con conceptos.			
Número y nombre del requerimiento	R-004 -Asociar objetos con conceptos del diccionario de datos.			
Objetivo	<i>Asociar objetos de base de datos con conceptos</i>			
Escenario	Descripción	Resultado Esperado	Resultado	Prioridad
	1 El actor ejecuta el <i>script</i> de prueba SC0029, para asociar y consultar la relación entre un objeto y un concepto.	Se registra en la tabla objetosBdConceptos, la relación entre el concepto y objeto de base de datos.	OK	Alta
Comentarios	N/A			

Nombre del Caso de Prueba	CP007 - Crear asociaciones de los objetos con propietarios.			
----------------------------------	---	--	--	--

Número y nombre del requerimiento	R-005 - Asociar los objetos con propietarios.			
Objetivo	<i>Asociar objetos de base de datos con propietarios</i>			
Escenario	Descripción	Resultado Esperado	Resultado	Prioridad
	1 El actor ejecuta el <i>script</i> de prueba SC0014, para registrar un propietario titular	El sistema debe registrar al propietario titular.	OK	Alta
	2 El actor ejecuta el <i>script</i> de prueba SC0015, para registrar un propietario titular.	El sistema debe registrar al propietario delegado.	OK	Alta
	3 El actor ejecuta el <i>script</i> de prueba SC0016, registrar delegados asignados por los titulares.	El sistema debe registrar la relación.	OK	Alta
	4 El actor ejecuta el <i>script</i> de prueba SC0017, para asignar objetos a un propietario titular.	El sistema debe registrar la relación.	OK	Alta
	5 El actor ejecuta el <i>script</i> de prueba SC0018, para asignar objetos a un propietario delegado, que no le corresponde al titular que le delegó.	El sistema no permite la asignación.	OK	Alta
	6 El actor ejecuta el <i>script</i> de prueba SC0019, para asignar objetos a un propietario delegado.	El sistema debe registrar la relación.	OK	Alta
Comentarios	N/A			

Nombre del Caso de Prueba	CP008 - Crear asociaciones de los objetos con controles.
----------------------------------	--

Número y nombre del requerimiento	R-006 - Asociar los objetos con controles.			
Objetivo	<i>Asociar objetos de base de datos con controles de seguridad</i>			
Escenario	Descripción	Resultado Esperado	Resultado	Prioridad
	1 El actor ejecuta el <i>script</i> de prueba SC0024, para registrar clases de controles de seguridad.	El sistema debe registrar la clase de control.	OK	Alta
	2 El actor ejecuta el <i>script</i> de prueba SC0025, para registrar controles de seguridad.	El sistema debe registrar el control.	OK	Alta
	3 El actor ejecuta el <i>script</i> de prueba SC0025, para registrar la relación entre controles de seguridad y los objetos de base de datos.	El sistema debe registrar la relación.	OK	Alta
Comentarios	N/A			

Nombre del Caso de Prueba	CP009 - Crear asociaciones de los objetos con observaciones de sus propietarios.			
Número y nombre del requerimiento	R-007 - Asociar los objetos con observaciones de sus propietarios.			
Objetivo	<i>Crear asociaciones de los objetos de base de datos con observaciones de los propietarios.</i>			
Escenario	Descripción	Resultado Esperado	Resultado	Prioridad
	1 El actor ejecuta el <i>script</i> de prueba SC0022, para registrar la relación entre comentarios y los objetos de base de datos.	El sistema debe registrar la relación.	OK	Alta

Comentarios	N/A
--------------------	-----

Nombre del Caso de Prueba	CP010 - Consultar información sobre los servidores vinculados.			
Número y nombre del requerimiento	R-008 - Consultar información sobre los servidores remotos vinculados.			
Objetivo	<i>Consultar información sobre los servidores vinculados.</i>			
Escenario	Descripción	Resultado Esperado	Resultado	Prioridad
	1 El actor ejecuta el <i>script</i> de prueba SC0035, para consultar los administradores por servidor.	El sistema realiza la consulta.	OK	Alta
	2 El actor ejecuta el <i>script</i> de prueba SC0036, para consultar los administradores de bases de datos por instancia.	El sistema realiza la consulta.	OK	Alta
	3 El actor ejecuta el <i>script</i> de prueba SC0005, para consultar información detallada sobre el servidor.	El sistema realiza la consulta.	OK	Alta
Comentarios	N/A			

Nombre del Caso de Prueba	CP011 - Consultar trazabilidad a los procesos del sistema del diccionario.			
Número y nombre del requerimiento	R-009 - Registrar trazabilidad de los procesos del sistema.			
Objetivo	<i>Consultar trazabilidad a los procesos del sistema</i>			
Escenario	Descripción	Resultado Esperado	Resultado	Prioridad

	1	El actor ejecuta el <i>script</i> de prueba SC0037, para consultar el resultado de los procesos ejecutados en el sistema, en la bitácora, para dar trazabilidad a estos.	El sistema debe poblar la tabla bitacoraProcesos	OK	Alta
Comentarios	N/A				

Nombre del Caso de Prueba	CP012 - Filtrar las bases de datos a las cuales realizar la extracción de datos.				
Número y nombre del requerimiento	R-010 - Definir las bases de datos a las cuales realizar la extracción de datos.				
Objetivo	<i>Filtrar las bases de datos a las cuales realizar la extracción de datos</i>				
Escenario	Descripción	Resultado Esperado	Resultado	Prioridad	
	1	El actor ejecuta el <i>script</i> de prueba SC0008, para consultar información detallada sobre la base de datos, el campo incluirAnálisis es el utilizado para definir cuál base de datos debe ser incluida en el análisis.	El sistema realiza la consulta.	OK	Alta
Comentarios	N/A				

Nombre del Caso de Prueba	CP013 - Definir los objetos participantes en el proceso de extracción.				
Número y nombre del requerimiento	R-011 - Definir objetos participantes.				
Objetivo	<i>Consultar los objetos de bases de datos pertenecientes a una base de datos.</i>				

Escenario	Descripción	Resultado Esperado	Resultado	Prioridad
	1 El actor ejecuta el <i>script</i> de prueba SC0038, para listar los objetos a integrar.	El sistema realiza la consulta.	OK	Alta
Comentarios	N/A			

Nombre del Caso de Prueba	CP014 - Consultar objetos por conceptos definidos en el diccionario.			
Número y nombre del requerimiento	R-012 - Consultar objetos por conceptos del diccionario			
Objetivo	<i>Asociar objetos de base de datos con conceptos definidos en el diccionario de datos empresarial.</i>			
Escenario	Descripción	Resultado Esperado	Resultado	Prioridad
	1 El actor ejecuta el <i>script</i> de prueba SC0039, para listar los objetos relacionados con un concepto.	El sistema realiza la consulta.	OK	Alta
Comentarios	N/A			

Nombre del Caso de Prueba	CP015 - Consultar objetos por propietarios			
Número y nombre del requerimiento	R-013 - Consultar objetos por propietarios			
Objetivo	<i>Consultar los objetos de base de datos por propietario</i>			
Escenario	Descripción	Resultado Esperado	Resultado	Prioridad
	1 El actor ejecuta el <i>script</i> de prueba SC0017, para listar los objetos que	El sistema realiza la consulta.	OK	Alta

	pertenecen a un propietario titular.			
	El actor ejecuta el <i>script</i> de prueba SC0019, para listar los objetos que pertenecen a un propietario delegado.	El sistema realiza la consulta.	OK	Alta
Comentarios	N/A			

Nombre del Caso de Prueba	CP016 - Consultar objetos por observaciones.			
Número y nombre del requerimiento	R-015 - Consultar las observaciones de los objetos.			
Objetivo	<i>Consultar las observaciones realizadas a los objetos de base de datos.</i>			
Escenario	Descripción	Resultado Esperado	Resultado	Prioridad
	1 El actor ejecuta el <i>script</i> de prueba SC0022, para listar las observaciones de un objeto.	El sistema realiza la consulta.	OK	Alta
Comentarios	N/A			

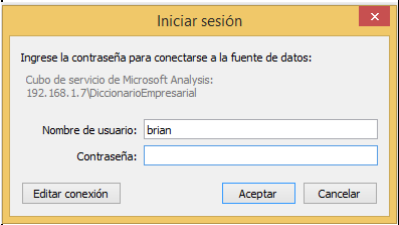
Nombre del Caso de Prueba	CP017 - Consultar objetos por controles.			
Número y nombre del requerimiento	R-014 - Consultar objetos por controles.			
Objetivo	<i>Consultar los controles relacionados a los objetos de base de datos</i>			
Escenario	Descripción	Resultado Esperado	Resultado	Prioridad
	1 El actor ejecuta el <i>script</i> de prueba SC0026, para listar los objetos que	El sistema realiza la consulta.	OK	Alta

	pertenecen a un control.			
Comentarios	N/A			

Nombre del Caso de Prueba	CP018 - Escalabilidad en el versionamiento de gestores de bases de datos.			
Número y nombre del requerimiento	R-016 - Escalabilidad en versionamiento de gestores.			
Objetivo	<i>Registrar bases de datos en diferente versión y tipo</i>			
Escenario	Descripción	Resultado Esperado	Resultado	Prioridad
	1 El actor ejecuta el <i>script</i> de prueba SC0002, para registrar múltiples gestores de bases de datos en múltiples versiones.	El sistema permite registrar múltiples gestores de bases de datos en múltiples versiones.	OK	Alta
Comentarios	N/A			

Nombre del Caso de Prueba	CP019 - Registrar funcional de forma modular.			
Número y nombre del requerimiento	R-017 - Escalabilidad.			
Objetivo	<i>Registrar funcionalidad de forma modular.</i>			
Escenario	Descripción	Resultado Esperado	Resultado	Prioridad
	1 El actor ejecuta el <i>script</i> de prueba SC0030 para registrar acciones.	El sistema permite registrar acciones.	OK	Alta
	2 El actor ejecuta el <i>script</i> de prueba SC0031 para registrar procesos.	El sistema permite registrar procesos.	OK	Alta

	3	El actor ejecuta el <i>script</i> de prueba SC0032, para registrar acciones por proceso.	El sistema permite registrar acciones por proceso.	OK	Alta
	4	El actor ejecuta el <i>script</i> de prueba SC0033, para registrar procesos por instancia.	El sistema permite registrar procesos por instancia.	OK	Alta
Comentarios	N/A				

Nombre del Caso de Prueba	CP020 - Solicitar autenticación para realizar las consultas.				
Número y nombre del requerimiento	R-018 - Solicitar autenticación para realizar consultas.				
Objetivo	<i>Solicitar autenticación para realizar consultas.</i>				
Escenario	Descripción	Resultado Esperado	Resultado	Prioridad	
	1	El actor ingresa a <i>Tableau</i> . Se muestra la siguiente pantalla:	El sistema solicita que se autentique por medio de su usuario y contraseña. Se muestra la siguiente pantalla: 	OK	Alta
Comentarios	N/A				

Apéndice 8. Código fuente.

Este se adjunta en formato digital.

Apéndice 9. *Scripts* de pruebas *SQL*.

Este se adjunta en formato digital.