

Universidad Central “Marta Abreu” de las Villas.

Facultad Matemática, Física y Computación

Licenciatura en Ciencia de la Computación



Trabajo de Diploma

**Título: DaGenSQL - una herramienta de
generación de datos para el desarrollo de
Sistemas de Información**

Autor: LAN VU THI HOANG

Tutor: M.Sc. Alain Pérez Alonso

Santa Clara, 2015

Dictamen



Hago constar que el presente trabajo fue realizado en la Universidad Central “Marta Abreu” de Las Villas como parte de la culminación de los estudios de la especialidad de Ciencia de la Computación, autorizando a que el mismo sea utilizado por la institución, para los fines que estime conveniente, tanto de forma parcial como total y que además no podrá ser presentado en eventos ni publicado sin la autorización de la Universidad.

Firma del autor

Los abajo firmantes, certificamos que el presente trabajo ha sido realizado según acuerdos de la dirección de nuestro centro y el mismo cumple con los requisitos que debe tener un trabajo de esta envergadura referido a la temática señalada.

Firma del tutor

**Firma del jefe
del Laboratorio**

Pensamiento

“La inteligencia da bondad, justicia y hermosura; como un ala, levanta el espíritu, como una corona, hace monarca al que la ostenta.”

José Martí

Dedicatoria

A mi padre: eres el sol que ilumina cada paso de mi vida

A mi madre: tu ternura me alivia de todas las preocupaciones y tristezas

A mis hermanos: son espejos que reflejan todo lo bueno del mundo para mí

A mi hermanita: te quiero más que a todos en este mundo

Agradecimientos

Quiero expresar mis más sinceros agradecimientos:

A toda mi familia por estar siempre al tanto de lo que acontece en mi vida.

Gracias a ella pude llegar hasta aquí.

*A mi tutor M.Sc. Alain Pérez Alonso por su disposición y paciencia
incomparable durante la realización de este trabajo.*

*A todos los demás profesores y amigos del laboratorio de Base de Datos por
siempre brindarme su atención y su cariño.*

*A todos los profesores de la Facultad de Matemática, Física y Computación por
todas sus enseñanzas; y por sus cariños y ayuda incondicional.*

*A mi amiga Mislady, por dedicarme su “especial” atención y su amor, por estar
siempre a mi lado, apoyarme y animarme en los momentos de tristezas.*

*A todos los amigos vietnamitas en Cuba, por siempre ayudarme y compartir
conmigo momentos inolvidables, tanto en las tristezas como en las alegrías.*

*A todos los amigos del aula, los demás amigos cubanos, y los compañeros de
otras nacionalidades por todos los momentos y recuerdos bonitos de la vida
estudiantil que hemos compartido.*

*Finalmente, a todos los que me han ayudado de una manera u otra durante la
realización de esta tesis.*

Agradecimientos

Muchas gracias a todos...

Lan Vu Thi Hoang

Resumen

Los Sistemas de Información (SI) en la actualidad se consideran indispensables debido a que mediante su implementación en una organización se obtienen ventajas significativas para el funcionamiento de la misma; como puede ser el ahorro de tiempo en la gestión de procesos. Existen muchas formas de implementar y comprobar un SI, una de ellas es mediante la ayuda de herramientas independientes que ayuden a generar grandes volúmenes de datos de manera sencilla.

En Cuba se utilizan herramientas propietarias de generación de datos con muchas prestaciones, pero no existe una herramienta propia. En el Laboratorio de Base de Datos del Centro de Estudios Informáticos (CEI) de la UCLV tampoco se ha trabajado en este tema; solo se han hecho proyectos para la generación automática de reglas de negocio en base de datos relacionales.

El propósito de este trabajo es desarrollar una aplicación de este tipo cuya interfaz permita crear reglas de generación de datos en diferentes gestores y se logre producir datos automáticamente. Este proceso ayuda a los desarrolladores a procesar y manejar una gran cantidad de información en el desarrollo de SI.

Abstract

The Information Systems (IS) are currently considered essential because through its implementation in an organization significant advantages are obtained for their operations; such as saving time on process management. There are many ways to implement and test an IS, one of them is with the help of independent tools to generate large volumes of data easily. Proprietary data generation tools with many interesting features are used in Cuba, but there is not an open source tool itself. The Database Laboratory of Center for Information Studies (CEI in Spanish) has also worked on this issue; but they have projects only for the automatic generation of business rules in relational databases.

The purpose of this work is to develop an application of this type whose interface allows to create data generation rules in different database management systems and be able to produce data automatically. This process helps developers to process and handle a lot of information in the IS development.

Tabla de contenidos

<i>Introducción</i>	<i>1</i>
Antecedentes	2
Planteamiento del problema	6
Objetivo General	6
Objetivos Específicos	6
Preguntas de la Investigación	7
Justificación del problema	7
Viabilidad de la investigación	7
Hipótesis general	8
Estructura del trabajo	8
<i>Capítulo 1. “Aspectos teóricos del proceso de la generación de datos”</i>	<i>10</i>
1.1 La generación de datos	10
1.1.1 Definición de la generación de datos.....	10
1.1.2 Generación de base de datos.....	10
1.1.3 Los pasos para generar datos	11
1.1.4 Generador de tipos datos o dominios	11
1.2 Gestores de base de datos	12
1.2.1 Concepto de Sistema de Gestión de Base de Datos	12
1.2.2 Las características principales de Gestores de Base de Datos.....	12
1.2.3 Ventajas y desventajas del uso Gestores de Base de Datos en proceso de generación de datos	13
1.2.4 Tipos de los Gestores de Base de Datos	14
1.3 Catálogo de un Sistema de Base de datos	17

1.3.1 ¿Qué es el catálogo del sistema?	17
1.3.2 Características del Catálogo de datos	17
1.3.3 Contenidos del Catálogo de datos	18
1.3.4 Actualización del Catálogo.....	19
1.3.5 El catálogo y el estándar ANSI/ISO	19
1.4 Metadatos para los Gestores de Base de datos	20
1.4.1 Definición de metadatos	20
1.4.2 Clasificación de metadatos	21
1.4.3 Tipos de metadatos	22
1.4.4 Ventajas y Desventajas de metadatos.....	22
1.5 El lenguaje estándar SQL.....	23
1.5.1 Lenguaje de definición de datos (DDL)	23
1.5.2 Lenguaje de manipulación de datos (DML)	24
1.6 Restricciones en SQL	25
1.6.1 Algunos tipos básicos de restricción	25
1.6.2 Restricción de la integridad	27
1.7 Algunas herramientas propietarias para la generación de datos	32
1.7.1 Databene Benerator.....	33
1.7.2 GenerateData	33
1.7.3 SQL Data Generator	34
1.7.4 Banner Datadect.....	35
1.7.5 Super Justo.....	35
1.8 Conexión a base de datos usando Java: JDBC.....	35
1.9 Extensión para otros orígenes mediante controladores.....	38
1.10 Conclusiones parciales del capítulo	38

Capítulo 2. “Análisis y diseño del sistema para la generación de datos”	40
2.1 Análisis y diseño de sistemas	40
2.2 Términos necesarios en el proceso de generar datos	40
2.3 Diagrama de actores y casos de uso	42
2.4 Descripción del caso de uso “Manipular las reglas para generar datos”	43
2.5 Descripción del caso de uso “Vaciar tabla”	46
2.6 Diagrama de actividad principal del sistema	47
2.7 Otros diagramas que componen el modelado del software	48
2.7.1 Diagrama de actividades para el caso de uso “Vaciar tabla”	48
2.7.2 Diagrama de actividades para el caso de uso “Establecer el orden de las reglas”	49
2.7.3 Diagrama de estados para manipular reglas	50
2.7.4 Diagrama de estados para ordenar reglas	51
2.7.5 Diagrama de despliegue	52
2.8 Diferentes formas a generar valores de un atributo	52
2.9 El paradigma Modelo – Vista – Controlador (MVC) del sistema	53
2.10 Extraer metadatos desde una base de datos	55
DataBaseMetaData	56
ResultSetMetaData	56
2.11 Conclusiones parciales del capítulo	56
Capítulo 3. “Características y uso de DaGenSQL”	58
3.1 El uso del sistema	58
3.2 Interfaz visual del sistema	59
3.2.1 Vista del logotipo del sistema.....	59
3.2.2 Crear la conexión con la base de datos.....	60

3.2.3 Opciones de la herramienta	61
3.2.4 Esquema relacional de la base de datos obtenido desde la herramienta.....	63
3.3 Crear reglas de generar datos de diferentes formas	64
3.3.1 Agregar regla para vaciar tablas	64
3.3.2 Crear regla de Inserción.....	66
3.3.3 Crear regla de Actualización	71
3.3.4 Crear regla de Eliminación	73
3.4 Modificar una regla.....	74
3.5 Conectar a otro origen de datos usando método Secuencia SQL	75
3.6 Resultados de experimentos con la herramienta	76
3.7 Conclusiones parciales del capítulo	79
<i>Conclusiones</i>	<i>80</i>
<i>Recomendaciones.....</i>	<i>81</i>
<i>Referencias Bibliográficas.....</i>	<i>82</i>

Lista de figuras

Figura 1. El concepto de Sistema de Información	2
Figura 2. Administrador de Drivers	36
Figura 3. Diagrama de Entidad Relación de una base de datos	41
Figura 4. Diagrama de Casos de uso.....	43
Figura 5. Diagrama de actividad principal.....	48
Figura 6. Diagrama de actividades para el caso de uso "Vaciar tabla"	49
Figura 7. Diagrama de actividades para el caso de uso "Establecer orden de las reglas" ...	50
Figura 8. Diagrama de estados para “Manipular reglas”	51
Figura 9. Diagrama de estados para “Ordenar reglas”	51
Figura 10. Diagrama de despliegue del sistema	52
Figura 11. Modelo Vista Controlador del sistema.....	54
Figura 12. El uso del sistema	58
Figura 13. Vista logotipo del sistema	59
Figura 14. Conectar al servidor.....	60
Figura 15. Crear Conexión.....	60
Figura 16. La pantalla principal de la herramienta “Generar datos SQL”	61
Figura 17. Esquema Relacional de la base de datos “Bodega”	64
Figura 18. Vaciar tablas	65
Figura 19. Ejecutar regla marcadas.....	65
Figura 20. Seleccionar la tabla y la operación de Inserción	66
Figura 21. Atributos y Métodos	67
Figura 22. Valores aleatorios	68
Figura 23. Lista de valores.....	68
Figura 24. Incremental	69
Figura 25. Secuencia SQL	69
Figura 26. Vista Previa	70
Figura 27. Opciones	70
Figura 28. Guardar resultados.....	71
Figura 29. Seleccionar la tabla y la operación de Actualización	72

Figura 30. Seleccionar campos a actualizar	72
Figura 31. Modificar datos.....	73
Figura 32. Crear regla de Eliminación	74
Figura 33. Inicio de la tabla “bodega” en PostgreSQL.....	75
Figura 34. Desmarcar “Usar Conexión actual”	76
Figura 35. Tomar datos de otro origen de datos	76
Figura 36. Experimento para 5000 tuplas en MySQL	77
Figura 37. Experimento con 300 operaciones	78
Figura 38. Experimento de insertar datos en BD y en formatos de texto	79

Lista de tablas

Tabla 1. Tabla de eventos para el caso “Manipular las reglas a generar datos”	46
Tabla 2. Tabla de eventos para el caso “Vaciar tabla”	47

Introducción

Hoy en día, con la necesidad de las empresas de desarrollar aplicaciones informáticas, los Sistemas de Información (SI) son utilizados ampliamente. En un SI interactúan varios elementos entre sí con el fin de apoyar las actividades de una empresa o negocio, pues se trata de reflejar parte del funcionamiento del mundo real con el objetivo de ayudar a las personas a resolver varios problemas.

A través del uso de SI, según (De Blanco, 1994), se logran importantes mejoras, pues se automatizan varios procesos operativos, se suministran una plataforma de información necesaria para la toma de decisiones, y lo más importante: su implantación logra ventajas competitivas. Así mismo aumenta la confiabilidad en la entrada, almacenamiento, procesamiento y salida de información.

Los sistemas de bases de datos son un tipo de “software” muy específico que nos ayuda a mantener información de un universo de discurso de forma computacional. Este universo es la fuente de información y para automatizar los procesos operativos de esa fuente es útil emplear los conocimientos de SI. Como herramienta computacional los SI son capaces de generar ventajas competitivas para la organización. Desde el punto de vista de la gestión empresarial, el conocimiento del entorno en un mundo cada vez más complejo y cambiante, origina una necesidad acuciante de información para la toma de decisiones, tanto para atacar nuevos mercados, como para proteger a la empresa de agentes externos que puedan vulnerar su estabilidad. (Cañavate, 2003)

En el laboratorio de Base de Datos del Centro de Estudios Informáticos (CEI) de la Universidad Central “Marta Abreu” de Las Villas se desarrolla un campo de investigación con respecto a la generación automática de datos en el desarrollo de SI. Para poder diseñar e implementar estas reglas en el desarrollo de SI se deben formalizar a través de una herramienta de trabajo efectiva y libre.

Por tanto, en el siguiente trabajo se presenta una herramienta que satisface la necesidad de manipular las reglas de generación de datos en diferentes gestores relacionales, la cual permitirá producir grandes volúmenes de datos con una significativa economía de tiempo.

Antecedentes

La información es considerada actualmente un recurso estratégico de gran importancia para las organizaciones. Contar con la información necesaria posibilita tomar las mejores decisiones, de ahí que el estudio de los sistemas de información constituya una prioridad empresarial.

En la visión de Lázaro Jesús un Sistema de Información (SI) se define como “*un conjunto de funciones interrelacionadas, hardware, software y de recurso humano que permite almacenar y procesar información*”. Los usuarios emplean esa información en dos actividades fundamentales: la toma de decisiones y el control. (Lázaro Jesús, 2012). La definición de SI se describe claramente en la **Figura 1**.



Figura 1. El concepto de Sistema de Información

Varios autores entre los que se encuentran (Stair and Reynolds, 2011), (Cañavate, 2003) y (Angie Mercedes, 2014) destacan el enfoque a procesos al definir un SI como un conjunto interrelacionado de elementos que recopilan, manipulan, almacenan y distribuyen datos e información con determinado mecanismo de retroalimentación, para dar apoyo al control y proceso de toma de decisiones en una organización. (Saroka Hernández, 2002) y (Cantalapiedra Llerena, 2006) centran la atención en que se trata de un proceso organizado en función del logro de determinados objetivos.

Aunque en principio, el SI puede existir sin el soporte de las tecnologías de la información, hoy en día, al hablar de un SI se hace alusión a un sistema basado en las ventajas que proveen las tecnologías de la información y esta será la acepción utilizada en el presente trabajo.

Desde el punto de vista funcional los elementos componentes del SI se pueden clasificar en subsistemas como son:

- **Subsistema físico o hardware:** Es el conjunto de dispositivos físicos interrelacionados sobre el cual el subsistema lógico realiza el tratamiento de los datos. Entre sus principales integrantes destacan los ordenadores y dispositivos de comunicación. Este subsistema realiza las funciones de entrada, salida, almacenamiento, procesamiento de datos y programas y comunicación entre dispositivos.
- **Subsistema lógico o software:** Conjunto de instrucciones, escritas en un lenguaje especial, y organizadas en programas que por una parte le dictan al subsistema físico que acciones realizar sobre los datos y por otra permiten la relación entre el usuario y la computadora.
- **Subsistema de datos:** Conjunto de datos a partir de los cuáles el sistema obtendrá información de salida útil tras un tratamiento adecuado. Los datos se encontrarán almacenados en archivos, ficheros y bases de datos.
- **Elementos humanos:** Usuarios del SI (internos o externos), personal técnico, personal de mantenimiento, analistas, administradores, entre otros.

- **Subsistema de procedimientos:** Formado por los criterios de calidad, normas, directrices, indicaciones, procedimientos, rutinas organizativas, entre otros.

En la vida real, todos los aspectos personales, naturales, culturales entre otros, necesitan ser organizados y generados para tener un buen formato. Los datos no escapan a esta filosofía y antes de publicarlos y servir a las demandas de las empresas es necesario generarlos y convertirlos lo cual es una tarea útil e importante.

El desarrollo de los procesos de generación de datos han sido ampliamente empleados en el campo de las bases de datos. Se logran varios beneficios al utilizar las aplicaciones automáticas de generación de datos para obtener grandes volúmenes de datos, aumentar la velocidad trabajo y reducir el tiempo de trabajo. Alguno de estos beneficios están vinculados a:

1. **Información clasificada** para el desarrollo de SI como por ejemplo información: militar, de delitos, gobierno, personal, etc. La información clasificada es un tipo de información sensible que está restringida por las leyes para clases particulares de personas. Se requiere una habilitación formal de seguridad para manejar y acceder a documentos clasificados.
2. **Métodos educativos** en asignaturas de la disciplina Bases de Datos: que son los métodos para proporcionar, evaluar y publicar los datos y conocimientos.
3. **Verificación de Restricciones de integridad y recursos activos:** antes de utilizar y poblar los datos en la base de datos, los SI se enfocan en la protección y validación de su información. La generación de grandes volúmenes de datos permite validar las restricciones implementadas y comprobar el sistema antes de su implantación.
4. **Comprobación de las prestaciones del servidor:** La generación intensiva de eventos a la base de datos de la aplicación permite comprobar las prestaciones del servidor donde se implantará el sistema.
5. **Fases de pruebas de un Sistema de Información:**
El ciclo de vida de un sistema de información está conformado por siete fases, las cuales son: (Berzal, 2015)

- **Planificación:** En esta fase se determina:
 - ✓ El alcance del proyecto.
 - ✓ Limitaciones de factores como son los recursos, presupuesto y tiempo.
- **Análisis:** En esta fase se determina:
 - ✓ Los requisitos del proyecto.
 - ✓ La verificación de requisitos.
 - ✓ La validación por parte de los usuarios finales.
- **Diseño:** En esta fase se produce:
 - ✓ La transformación de requisitos en requisitos técnicos por parte de los especialistas en la información técnica.
 - ✓ Un diseño técnico preliminar para realizar un diseño técnico detallado.
 - ✓ La asignación de funciones para el desarrollo del sistema.
- **Implementación:** En esta fase se brinda:
 - ✓ Los requerimientos de hardware y software.
 - ✓ El ensamblaje del sistema de información.
- **Prueba:** En esta fase se verifica:
 - ✓ Que la evaluación cumpla con su objetivo.
 - ✓ La calidad del sistema por parte del analista.
- **Instalación y despliegue:** En esta fase se entrega el sistema a los usuarios finales y se detalla su manejo.
- **Uso y mantenimiento:** En esta fase se realiza:
 - ✓ El mantenimiento periódico garantizando al sistema de información.
 - ✓ La actualización del sistema de acuerdo a las necesidades.
 - ✓ Se perfecciona.

De hecho, han sido desarrollados varios software propietarios destinados enteramente a esta temática, algunos de los más significativos que pueden mencionarse son: *DTM Data Generator* (sqledit, 2015), *Databene Benerator* (Bergmann, 2015), *GenerateData* (generatedata, 2015) y *RedGate SQL Data Generator* (redgate, 2015).

Planteamiento del problema

En el desarrollo de sistemas de base de datos y en general de sistemas de información que utilicen gestores relacionales, es necesario probar los recursos y procesos de dicho sistema. Usualmente los datos necesarios para estas pruebas son creados “a mano”, con los consiguientes errores y problemas para los desarrolladores. Según la presente investigación, Cuba no posee una herramienta de este tipo y las disponibles son herramientas propietarias.

En el presente trabajo se desarrolla una herramienta capaz de generar automáticamente grandes volúmenes de datos, facilitando el desarrollo de sistemas de información y el trabajo a sus desarrolladores.

Por los motivos explicados anteriormente se proponen los siguientes objetivos:

Objetivo General

Desarrollar una herramienta para generar automáticamente datos en diferentes gestores de base de datos, a fin de que sean utilizados en el desarrollo de sistemas de información, usando los metadatos necesarios.

Objetivos Específicos

1. Extraer los metadatos de diferentes gestores de base de datos desde su catálogo, específicamente SQL Server, MySQL y PostgreSQL.
2. Establecer las diferentes formas para generar valores de un atributo.
3. Crear diferentes reglas para poblar las tablas del sistema y establecer un orden de las reglas creadas.
4. Diseñar e implementar una herramienta capaz de generar datos automáticamente de forma sencilla y amigable.

Preguntas de la Investigación

1. ¿Cómo generar correctamente los diferentes valores de los atributos, específicamente en la inserción y la actualización?
2. ¿Se generan sin límite las reglas de generación de datos en orden automático?
3. ¿Cómo lograr un mejor diseño de las funcionalidades de entrada de datos y poblar datos al sistema para la satisfacción del desarrollador?
4. ¿Cuáles son los aspectos principales que deben tenerse en cuenta para implementar la herramienta de generación de datos?

Justificación del problema

En el proceso de generación de datos es necesario extraer primeramente la información de una base de datos. Para tomar estos datos, existen dos formas: manual o automática. Las manuales son aquellas que se proporcionan en forma directa por el usuario, mientras que las automáticas son datos que provienen de otros sistemas o módulos.

Existen diversas herramientas para generar grandes volúmenes de datos por lo cual es importante realizar una investigación para conocer y tomar experiencia de cómo se realiza este trabajo en la actualidad.

Esta investigación puede ser utilizada en varios gestores de base de datos. Dado que mediante un catálogo de datos con sus restricciones se obtiene las reglas creadas en una herramienta y luego se generan los datos en forma automática y ordenada.

Viabilidad de la investigación

El estado actual de las características de los gestores de base de datos y del desarrollo de las herramientas para generar datos permite una amplia gama de ideas a exponer en este trabajo. Para el desarrollo de esta investigación se cuenta con la información necesaria de los datos y las sentencias del estándar SQL. Por otro lado, se cuenta con el apoyo del grupo de investigación de base de datos, con vasta experiencia en el área.

Hipótesis general

La utilización de las herramientas de generación de datos que se han desarrollado para almacenar grandes volúmenes de datos permite mejorar el desarrollo de SI ahorrando tiempo y recursos a sus desarrolladores. Se puede desarrollar una herramienta capaz de extraer datos obtenidos desde diferentes gestores de bases de datos relacionales, para generar los atributos en diferentes formas con cada gestor de base de datos correspondientes.

Estructura del trabajo

En el presente trabajo se estructura esencialmente en tres capítulos:

- **Capítulos 1:** se presenta una descripción fundamental sobre base de datos SQL y algunas características típicas de las herramientas propietarias de generación de datos que le permita al lector obtener un conocimiento general de este tema y una comprensión sobre los capítulos siguientes.
- **Capítulo 2:** se propone una presentación sobre la herramienta de generar datos a través del análisis y diseño del sistema.
- **Capítulo 3:** se dedica a describir el uso y la validación del sistema que permita apoyar a los lectores cómo utilizar la herramienta. También se presenta un experimento computacional para evaluar el proceso de generación de datos de esta herramienta.

Finalmente aparecen las conclusiones y referencias bibliográficas correspondientes.

CAPÍTULO 1:
“ASPECTOS TEÓRICOS DEL
PROCESO DE LA
GENERACIÓN DE DATOS”

Capítulo 1. “Aspectos teóricos del proceso de la generación de datos”

En este capítulo se presentan y describen los conceptos generales de generación de datos con las definiciones y clasificaciones necesarias de base de datos SQL; así como la información necesaria para la generación de datos en una base de datos. También se exponen algunas características de las herramientas propietarias sobre el enfoque de generación de datos que se han desarrollado para establecer una comparación entre ellas; a cual contribuirá a la base teórica de la investigación.

1.1 La generación de datos

1.1.1 Definición de la generación de datos

Existen diversos conceptos de la generación de datos, pero de la forma sencilla se puede definir como un proceso de producción de grandes volúmenes de datos.

La generación de datos se refiere a la teoría y los métodos usados por investigadores para crear datos de una fuente probada de datos en un estudio cualitativo. Las fuentes de datos incluyen a los participantes humanos, los documentos, las organizaciones, los eventos, etc. Los datos cualitativos son productos de las fuentes de datos y así también los incluyen cotizaciones, copias, observaciones y extractos de documentos como las imágenes y los artículos del periódico.

Para generar datos de una fuente probada de datos, los investigadores interactúan con la fuente de datos usando métodos cualitativos de investigación.

1.1.2 Generación de base de datos

Generalmente, los modelos de los datos se han desarrollado en tres generaciones. La primera generación de modelos de los datos se tiende a rechazar, sin embargo fue el origen de las

bases de datos. Hasta el momento, las bases de datos más comercialmente aceptadas han sido las bases de datos de segunda generación que utilizan el modelo relacional. Las bases de datos relacionales son definitivamente las que poseen la mayor parte del mercado. Sin embargo, el mundo de los lenguajes programación ha venido cambiando y se han presentado nuevas plataformas como lo son Java y Microsoft con .NET, entre otras. Las opciones del desarrollador son cada vez más amplias y para elegir el desarrollador cuenta con modelos orientado a objetos y modelos no orientados a objetos.

1.1.3 Los pasos para generar datos

En el proceso de generar datos desde una fuente de datos, los desarrolladores necesitan establecer los pasos siguientes:

- Crear un plan de generación de datos.
- Seleccionar las tablas que desea poblar con datos y especificar cuántos datos desea generar para cada tabla. En algunos casos sólo necesitará generar datos para algunas tablas.
- Seleccionar las columnas que desea poblar de datos y establecer la configuración de cada columna. Esta información es el núcleo del plan de generación de datos. Puede seleccionar generadores de datos y establecer las propiedades de cada columna para controlar de forma precisa los datos que se van a generar para rellenar cada columna.
- Ejecutar el plan de generación de datos para generar los datos.

1.1.4 Generador de tipos datos o dominios

Existen diversos generadores de datos para crear distintos tipos de datos o dominios. Por ejemplo, el generador de datos de tipo entero genera valores enteros aleatorios, el generador de cadenas genera cadenas aleatorias y el generador de expresiones regulares genera cadenas que coinciden con un modelo especificado.

Para extender las funciones de generación de datos, se pueden crear generadores de datos personalizados o extender los generadores de datos estándar. Si se tiene una regla de negocios que los generadores de datos estándar no pueden satisfacer, se puede crear un generador de

datos personalizado. Después, se puede utilizar el generador personalizado para llenar esa columna con datos.

1.2 Gestores de base de datos

1.2.1 Concepto de Sistema de Gestión de Base de Datos

Según (Adrian Martinez, 2015) un Gestor de Base de Datos o Sistema de Gestión de Base de Datos (SGBD) es un software que permite introducir, organizar y recuperar la información de las bases de datos; en definitiva, administrarlas. También en (Wikipedia, 2015b) se define SGBD permite el almacenamiento, modificación y extracción de la información en una base de datos, además de proporcionar herramientas para añadir, borrar, modificar y analizar los datos. Los usuarios pueden acceder a la información usando herramientas específicas de interrogación y de informes, o bien mediante aplicaciones al efecto.

Especialmente un SGBD permite controlar el acceso a los datos, asegurar su integridad, gestionar el acceso concurrente a ellos, recuperar los datos tras un fallo del sistema y hacer copias de seguridad. Las bases de datos y los sistemas para su gestión son esenciales para cualquier área de negocio y deben ser gestionados con esmero. (Wikipedia, 2015b)

1.2.2 Las características principales de Gestores de Base de Datos

Existen muchas características de Gestores de Base de Datos y cada autor tiene su forma para plantearlas. Pero generalmente se puede expresar algunas características principales como las siguientes (Adrian Martinez, 2015):

- **Independencia:** modificar el esquema (físico o lógico) de una base de datos sin tener que realizar cambios en las aplicaciones que se sirven de ella.
- **Redundancia mínima:** evitar la aparición de información repetida o redundante.
- **Consistencia:** todos los datos repetidos se actualizan de forma simultánea.
- **Seguridad:** garantizar que la información se encuentra asegurada frente a usuarios malintencionados.

- **Integridad:** garantizar la validez de los datos almacenados.
- **Recuperación:** copias de la información almacenada en la base de datos.
- **Control de la concurrencia:** controlar el acceso concurrente a la información, que podría derivar en inconsistencias.

1.2.3 Ventajas y desventajas del uso Gestores de Base de Datos en proceso de generación de datos

Cuando se trabaja con los Gestores de Base de Datos, se puede plantear algunas ventajas como:

- Mejoras en la integridad de datos.
- Mejoras en la seguridad.
- Mejoras en la accesibilidad a los datos.
- Mejoras en la productividad.
- Mejoras en el mantenimiento gracias a la independencia de datos.
- Aumento de la concurrencia.
- Mejoras en los servicios de copias de seguridad y de recuperación ante fallos.

Realmente existe también algunas desventajas cuando se utilizan los Gestores de Base de Datos, ellas son (Adrian Martinez, 2015):

- Típicamente, es necesario disponer de una o más personas que administren la base de datos, esto puede llegar a incrementar los costos de operación en una empresa.
- Si se tienen muy pocos datos que son usados por un único usuario por vez y no hay que realizar consultas complejas sobre los datos, entonces es posible que sea mejor usar una hoja de cálculo.
- Complejidad: el software muy complejo y las personas que vayan a usarlo deben tener conocimiento de las funcionalidades del mismo para poder aprovecharlo al máximo.

- **Tamaño:** la complejidad y la gran cantidad de funciones que tienen hacen que sea un software de gran tamaño, lo que requiere una gran cantidad de memoria para poderlo ejecutar.
- **Costo del hardware adicional:** los requisitos de hardware para ejecutar un SGBD por lo general son relativamente altos, por lo que estos equipos pueden llegar a costar gran cantidad de dinero.

1.2.4 Tipos de los Gestores de Base de Datos

Un SGBD es un sistema de software que permite la creación de bases de datos y es muy efectivo para el acceso concurrente de varios usuarios. Con el objetivo de garantizar la confidencialidad, la calidad, la seguridad y la integridad de los datos (Ecured.cu, 2015b), hay que conocer y seleccionar correctamente el tipo de SGBD.

Existen muchos tipos de Gestores de Bases de Datos, los típicos son:

- Gestores de datos libres
 - ✓ MySQL
 - ✓ Sydes (Linux)
 - ✓ SQLIFE
 - ✓ PostgreSQL
 - ✓ Filebird
 - ✓ ApacheDerby
- Gestores de datos propietarios
 - ✓ Microsoft Access
 - ✓ Microsoft SQL Server
 - ✓ Open Access
 - ✓ Oracle
 - ✓ Paradox
 - ✓ Adbantase

Los detalles de algunos son:

MySQL

MySQL es un SGBD relacional rápido, sólido y flexible. Es conveniente para la creación de bases de datos con acceso desde páginas web dinámicas y realizar múltiples y rápidas consultas.

Es un sistema cliente/servidor, por lo que permite trabajar como servidor multiusuario y de subprocesamiento múltiple, o sea, cada vez que se crea una conexión con el servidor, el programa servidor establece un proceso para manejar la solicitud del cliente, controlando así el acceso simultáneo de un gran número de usuarios a los datos y asegurando el acceso a usuarios autorizados solamente. Es uno de los Sistemas Gestores de Bases de Datos más utilizado en la actualidad. (Cobo, 2005)

PostgreSQL

Según (Lockhart, 2001), PostgreSQL es un Sistema Gestor de Bases de Datos Relacional Orientadas a Objetos, derivado de Postgres, desarrollado en la Universidad de California, en el Departamento de Ciencias de la Computación de Berkeley. Es un SGBD de código abierto, brinda un control de concurrencia multiversión que permite trabajar con grandes volúmenes de datos; soporta gran parte de la sintaxis SQL y cuenta con un extenso grupo de enlaces con lenguajes de programación.

Posee características significativas de los motores de datos, entre las que se pueden incluir las subconsultas, los valores por defecto, las restricciones a valores en los campos (constraints) y los disparadores (triggers). Ofrece funcionalidades en línea con el estándar SQL92, incluyendo claves primarias, identificadores entrecomillados, conversión de tipos y entrada de enteros binarios y hexadecimales. (Lockhart, 2001)

Microsoft Access

Microsoft Access es un Sistema de Gestión de Bases de Datos Relacionales para los sistemas operativos Microsoft Windows, desarrollado por Microsoft y orientado a ser usado en un entorno personal o en pequeñas organizaciones. Es un componente de la suite ofimática

Microsoft Office. Permite crear ficheros de bases de datos relacionales que pueden ser fácilmente gestionadas por una interfaz gráfica simple. Además, estas bases de datos pueden ser consultadas por otros programas. Este programa permite manipular los datos en forma de tablas (formadas por filas y columnas), crear relaciones entre tablas, consultas, formularios para introducir datos e informes para presentar la información. (Ecured.cu, 2015b)

Microsoft SQL Server

SQL Server es un Sistema Gestor de Base de Datos relacional producido por Microsoft. Es un sistema cliente/servidor que funciona como una extensión natural del sistema operativo Windows. Entre otras características proporciona integridad de datos, optimización de consultas, control de concurrencia, backup y recuperación. (Petkovic, 2005)

Es relativamente fácil de administrar a través de la utilización de un entorno gráfico para casi todas las tareas de sistema y administración de bases de datos. Utiliza servicios del sistema operativo Windows para ofrecer nuevas capacidades o ampliar la base de datos, tales como enviar y recibir mensajes y gestionar la seguridad de la conexión. Es fácil de usar y proporciona funciones de almacenamiento de datos que sólo estaban disponibles en Oracle y otros Sistemas Gestores de Bases de Datos más caros. (Petkovic, 2005)

Oracle

Oracle es un Sistema de Gestión de Base de Datos objeto-relacional, desarrollado por Oracle Corporation. Se considera a Oracle como uno de los Sistemas de Bases de Datos más completos, destacando las siguientes características: soporte de transacciones, estabilidad, escalabilidad y soporte multiplataforma.

Su dominio en el mercado de servidores empresariales ha sido casi total hasta hace poco. Recientemente, sufre la competencia del Microsoft SQL Server de Microsoft y de la oferta de otro Sistema de Gestión de Base de datos Relacionales (RDBMS) con licencia libre como PostgreSQL, MySQL o Firebird. Las últimas versiones de Oracle han sido certificadas para poder trabajar bajo GNU/Linux.

1.3 Catálogo de un Sistema de Base de datos

Un SGBD lleva gran cantidad de información con el fin de efectuar sus funciones de gestión de datos. En (cursos.itam.mx, 2015) se refiere: en una base de datos relacional, esta información está almacenada típicamente en el catálogo de sistema. La información del catálogo del sistema describe las tablas, las vistas, las columnas, los usuarios, los índices, los privilegios y otras características estructurales de la base de datos.

1.3.1 ¿Qué es el catálogo del sistema?

El catálogo de sistema es una colección de tablas especiales en una base de datos que son propiedad, están creadas y son mantenidas por el propio de SGBD. Estas tablas del sistema contienen datos que describen la estructura de la base de datos. Las tablas del catálogo de sistema son automáticamente creadas en la base de datos. Almacenando la información estructural en tablas del sistema, el SGBD puede utilizar sus propios métodos y lógica de acceso para recuperar rápida y eficientemente la información que necesita en la realización de esas tareas. (cursos.itam.mx, 2015)

El acceso de los usuarios al catálogo de sistema es de solo lectura. El SGBD impide a los usuarios actualizar o modificar directamente las tablas del sistema, ya que tales modificaciones destruirán la integridad a la base de datos. En su lugar, el propio SGBD tiene a su cargo insertar, eliminar y actualizar filas de las tablas del sistema cuando modifica la estructura de una base de datos. Las sentencias en Lenguaje de Definición de Datos (DDL), tales como: CREATE, ALTER, DROP, GRANT y REVOKE producen cambios en las tablas del sistema como consecuencia de sus acciones. (cursos.itam.mx, 2015)

1.3.2 Características del Catálogo de datos

El catálogo del SGBD tiene las características siguientes (Date, 2001):

- El catálogo está formado por tablas.
- Son tablas dinámicas que el SGBD mantiene.

- Se las puede relacionar (JOIN).
- Toda tabla tiene dueño: El dueño del Catálogo es System.
- System otorga GRANT SELECT TO PUBLIC.
- El SGBD lo usa para interpretar un comando y chequear autorizaciones.
- El optimizador usa información sobre los índices para elegir una estrategia de acceso posible.
- El subsistema de autorización utiliza información del catálogo referente a los usuarios y privilegios para conceder o rechazar solicitudes específicas de usuarios.
- Los catálogos no son idénticos, difieren en detalles.

1.3.3 Contenidos del Catálogo de datos

Cada tabla del catálogo de sistema contiene información referente a una sola clase de elemento estructural de la base de datos. Aunque los detalles varían, casi todos los productos SQL comerciales incluyen tablas del sistema que describen cada una de estas cinco entidades (cursos.itam.mx, 2015):

- **Tablas.** El catálogo describe cada tabla de la base de datos, identificando su nombre, su propietario, el número de columnas que contiene, su tamaño, etc.
- **Columnas.** El catálogo describe cada columna de la base de datos, proporcionando el nombre de la columna, la tabla a la que pertenece, su tipo de datos, su tamaño, si están permitidos los NULL.
- **Usuarios.** El catálogo describe a cada usuario autorizado de la base de datos, incluyendo el nombre, una forma cifrada de la contraseña del usuario y otros datos.
- **Vistas.** El catálogo describe cada vista definida en la base de datos, incluyendo su nombre, el nombre de su propietario, la consulta que define la vista, etc.

- **Privilegios.** El catálogo describe cada grupo de privilegios concedidos en la base de datos, incluyendo los nombres del donante y el donatario, los privilegios concedidos, el objetivo sobre el cual se han concedido los privilegios, etc.

1.3.4 Actualización del Catálogo

Las consultas al catálogo pueden realizarse con las mismas sentencias que se consultan cualquier base de datos o tabla. Sin embargo las actualizaciones al catálogo (INSERT, DELETE, UPDATE) no son posibles pues sería peligroso, por lo tanto, el sistema los rechaza (Bernardo, 2015). Se puede describir un ejemplo si se permitiera actualizaciones al catálogo: téngase la tabla *Estudiante* en el catálogo (la tabla contiene solo una fila por cada columna). Si se permitiera borrar una línea de la tabla *Estudiante* se estaría eliminando una columna del catálogo y para el sistema esa columna no existiría más. De manera que cualquier intento por recuperar datos de esa columna fracasaría.

Por tanto, es imposible actualizar el catálogo de datos directamente con sentencias INSERT, DELETE y UPDATE. Sin embargo existe una forma de actualizarlo indirectamente mediante sentencias CREATE TABLE, DROP TABLE y ALTER TABLE, ya que (Date, 2001):

- CREATE TABLE es el análogo al INSERT
- DROP TABLE es el análogo al DELETE
- ALTER TABLE es el análogo al UPDATE

1.3.5 El catálogo y el estándar ANSI/ISO

El estándar SQL ANSI/ISO actual no especifica la estructura y los contenidos del catálogo de sistema. De hecho, el estándar no requiere la existencia de un catálogo de sistema en absoluto. Sin embargo, todos los productos SGBD basados en SQL más importantes proporcionan un catálogo de sistema de una forma u otra. La estructura del catálogo y las

tablas que contiene varían considerablemente de un producto SGBD a otro. (cursos.itam.mx, 2015)

Debido a la creciente importancia de las herramientas de base de datos de propósito general que deben acceder al catálogo de sistema, el estándar SQL2 propuesto incluye una especificación de un catálogo de sistema mínimo. La especificación está basada generalmente en las tablas de sistema DB2, pero elimina los contenidos específicos de DB2.

1.4 Metadatos para los Gestores de Base de datos

1.4.1 Definición de metadatos

El término “metadatos” no tiene una definición única (Wikipedia, 2015a). También existen muchas descripciones sobre metadatos. Dentro de ellas, la más común y sencilla es la que conceptualiza los metadatos como datos que describen otros datos.

En general, un grupo de metadatos se refiere a un grupo de datos llamado recurso. El concepto de metadatos es análogo al uso de índices para localizar objetos en vez de datos. Por ejemplo, en una biblioteca se usan fichas que especifican autores, títulos, casas editoriales y lugares para buscar libros. Así, los metadatos ayudan a ubicar datos. (Wikipedia, 2015a)

La mayoría de las veces no es posible diferenciar entre datos y metadatos. Por ejemplo, un poema es un grupo de datos, pero también puede ser un grupo de metadatos si está adjuntado a una canción que lo usa como texto. Muchas veces, los datos son tanto "datos" como "metadatos". Por ejemplo, el título de un texto es parte del texto como a la vez es un dato referente al texto (dato como metadato). (Wikipedia, 2015a)

Según (Lapiente, 2012) un metadato no es más que un dato estructurado sobre la información, o sea, información sobre información, o de forma más simple, datos sobre datos. Los metadatos en el contexto de la Web que son datos que se pueden guardar, intercambiar y procesar por medio del ordenador y que están estructurados de tal forma que

permiten ayudar a la identificación, descripción, clasificación y localización del contenido de un documento o recurso *web* y que, por tanto, también sirven para su recuperación.

Sin embargo, si se acota la definición de metadatos dándole un sentido más estricto, los metadatos sólo serían posibles en un contexto digital y en red ya que sólo dentro de este contexto se pueden utilizar los metadatos con la función que les caracteriza, que es la de la localización, identificación y descripción de recursos, legibles e interpretables por máquina. La definición que dan Berners-Lee y el World Wide Web Consortium (W3C) en Metadata Architecture es esclarecedora para entender el concepto de metadatos en este estricto sentido: "*Los metadatos son información inteligible para el ordenador sobre recursos Web u otras cosas*". (Lapuente, 2012)

1.4.2 Clasificación de metadatos

Según (Wikipedia, 2015a), los metadatos se clasifican usando tres criterios:

- **Contenido:** Subdividir metadatos por su contenido es lo más común. Se puede separar los metadatos que describen el recurso mismo de los que describen el contenido del recurso.
- **Variabilidad.** Según la variabilidad se puede distinguir metadatos mutables e inmutables. Los inmutables no cambian, no importa qué parte del recurso se vea, por ejemplo el nombre de un fichero. Los mutables difieren de parte a parte, por ejemplo el contenido de un vídeo.
- **Función.** Los datos pueden ser parte de una de las tres capas de funciones: subsimbólicos, simbólicos y lógicos. Los datos subsimbólicos no contienen información sobre su significado. Los simbólicos describen datos subsimbólicos, es decir añaden sentido. Los datos lógicos describen cómo los datos simbólicos pueden ser usados para deducir conclusiones lógicas, es decir añaden comprensión.

1.4.3 Tipos de metadatos

Los tipos de metadatos por parte de diferentes autores son varios. Generalmente se puede establecer como las siguientes clasificaciones (blog.classora, 2015):

- **Metadatos descriptivos:** sirven para la descripción e identificación de los recursos de información. Permiten la búsqueda y recuperación de los datos, así como distinguir recursos y entender sus contenidos. Se realizan mediante estándares como Dublin Core, estándares MARC, micro datos de HTML... etc.
- **Metadatos estructurales:** son los que más influyen en la recuperación de la información, facilitando la navegación y presentación de los recursos. Ofrecen información sobre su estructura interna y permiten establecer relaciones entre ellos, de manera que pueden incluso unir los archivos de imagen y textos que están relacionados. Los estándares más difundidos para en este sentido son SGML, RDF y EAD (Encoded Archival Description).
- **Metadatos administrativos:** incluyen datos para la gestión de derechos, firma digital, auditorías de calidad, control de acceso mediante contraseñas, etc. También se podrían enmarcar en esta categoría los metadatos sobre preservación a largo y corto plazo. Ejemplo, imágenes DICOM que incluyen metadatos como: tipo y modelo de escáner utilizado, resolución, paciente, limitaciones de reproducción, etc.

1.4.4 Ventajas y Desventajas de metadatos

Las grandes ventajas del uso de metadatos radican en que se usa el mismo contenido del documento como un recurso de datos y que los metadatos valen también para recursos que no tienen únicamente la morfología de texto, sino para cualquier tipo de morfologías tales como vídeo, audio o imágenes.

Para trabajar con un recurso de datos, hay que tener en cuenta que también existen algunas desventajas de metadatos que son las siguientes categorías (Wikipedia, 2015a):

- Los metadatos son costosos y necesitan demasiado tiempo. Las empresas no van a producir metadatos porque no hay demanda y los usuarios privados no van a invertir tanto tiempo.
- Los metadatos son demasiado complicados.
- Los metadatos dependen del punto de vista y del contexto. No hay dos personas que añadan los mismos metadatos. Además, los mismos datos pueden ser interpretados de manera totalmente diferente, dependiendo del contexto.
- Los metadatos son ilimitados.
- Los metadatos son superfluos porque ya hay buscadores potentes para textos.

1.5 El lenguaje estándar SQL

SQL es un lenguaje de base de datos normalizado, utilizado por los diferentes motores de bases de datos para realizar determinadas operaciones sobre los datos o sobre la estructura de los mismos (personal.lobocom, 2015). También se puede definir SQL como un lenguaje estándar de definición y manipulación (y consulta) de bases de datos relacionales que incluye:

- Características del Álgebra Relacional.
- Características del Cálculo Relacional de Tuplas.

Existen dos tipos de comandos SQL:

- Lenguaje de definición de datos (DDL)
- Lenguaje de manipulación de datos (DML)

Se plantea las características con los operadores de cada tipo como siguientes:

1.5.1 Lenguaje de definición de datos (DDL)

En (Wikipedia, 2015c), DDL es el que permite crear y definir nuevas bases de datos, campos e índices. Este lenguaje también se hace la modificación de la estructura de los objetos de la

base de datos. Incluye órdenes para modificar, borrar o definir las tablas en las que se almacenan los datos de la base de datos.

Existen cuatro operaciones básicas:

- **CREATE** (crear): Utilizado para crear nuevas tablas, campos e índices.
- **ALTER** (modificar): Utilizado para modificar la estructura de una tabla u objeto. Se pueden agregar/quitar campos a una tabla, modificar el tipo de un campo, agregar/quitar índices a una tabla, modificar un trigger, etc.
- **DROP** (eliminar): permite eliminar un objeto de la base de datos. Puede ser una tabla, vista, índice, trigger, función, procedimiento o cualquier objeto que el motor de la base de datos soporte. Si se quiere borrar todo el contenido de la tabla, es mucho más rápido aunque la tabla es muy grande.
- **TRUNCATE** (borrar tabla): permite truncar todo el contenido de una tabla, sólo sirve cuando se quiere eliminar absolutamente todos los registros, ya que no se permite la cláusula WHERE. El comando TRUNCATE borra la tabla y la vuelve a crear y no ejecuta ninguna transacción.

1.5.2 Lenguaje de manipulación de datos (DML)

Es el lenguaje que permite generar consultas para ordenar, filtrar y extraer datos de la base de datos. El lenguaje de manipulación de datos más popular hoy día es SQL, usado para recuperar y manipular datos en una base de datos relacional. (Wikipedia, 2015c)

Existen cuatro operaciones básicas se este lenguaje (personal.lobocom, 2015):

- **SELECT**: Utilizado para consultar registros de la base de datos que satisfagan un criterio determinado.
- **INSERT**: Utilizado para cargar lotes de datos en la base de datos en una única operación.
- **UPDATE**: Utilizado para modificar los valores de los campos y registros especificados.
- **DELETE**: Utilizado para eliminar registros de una tabla de una base de datos.

Con cualquier de los dos tipos de comandos (DDL o DML), para definir los datos que desea seleccionar o manipular se utilizan las clausuras como son (personal.lobocom, 2015):

- **FROM:** Utilizado para especificar la tabla de la cual se van a seleccionar los registros.
- **WHERE:** Utilizada para especificar las condiciones que deben reunir los registros que se van a seleccionar.
- **GROUP BY:** Utilizado para separar los registros seleccionados en grupos específicos.
- **HAVING:** Utilizado para expresar la condición que debe satisfacer cada grupo.
- **ORDER BY:** Utilizado para ordenar los registros seleccionados de acuerdo con un orden específico.

De esos puntos se utilizan los operadores con las funciones correspondientes para controlar los datos necesarios.

1.6 Restricciones en SQL

Se utilizan restricciones para limitar los valores que puede recibir una columna de una tabla. Las restricciones definen reglas relativas a las columnas y constituyen el mecanismo estándar para exigir la integridad.

1.6.1 Algunos tipos básicos de restricción

En SQL existen las siguientes clases de restricciones:

- **NOT NULL:** especifica que una columna no acepta el valor NULL, es decir, que esa columna siempre tiene que tener algún valor, no puede estar vacía. NULL indica que el valor es desconocido. Un valor NULL no es lo mismo que un valor cero o vacío. No hay dos valores NULL que sean iguales. La comparación entre dos valores NULL, o entre un valor NULL y cualquier otro valor, tiene un resultado desconocido porque el valor de cada NULL es desconocido.

- **DEFAULT:** se utiliza para establecer un valor por defecto a una columna. Si no se especifica un valor al insertar una fila, entonces se pondrá el valor por defecto (DEFAULT) que tenga cada columna.
- **PRIMARY KEY:** identifica la columna o el conjunto de columnas cuyos valores identifican de forma exclusiva cada una de las filas de una tabla. Dos filas de la tabla no pueden tener el mismo valor ni valores NULL de clave principal. Todas las tablas tienen que tener una clave principal.
- **FOREIGN KEY:** identifica y exige las relaciones entre las tablas. Una clave externa de una tabla apunta a una clave candidata de otra tabla. No se puede insertar una fila que tenga un valor de clave externa, excepto NULL, si no hay una clave candidata con dicho valor.
- **CHECK:** exige la integridad del dominio mediante la limitación de los valores que se pueden asignar a una columna. Una restricción CHECK especifica una condición de búsqueda booleana (se evalúa como TRUE, FALSE o desconocido) que se aplica a todos los valores que se indican en la columna. Se rechazan todos los valores que se evalúan como FALSE. En una misma columna se pueden especificar varias restricciones CHECK.
- **UNIQUE:** garantiza que no se escriben valores duplicados en columnas específicas que no forman parte de una clave principal. También exige la unicidad; se usa si desea exigir la unicidad de una columna o una combinación de columnas que no forman la clave principal. Las restricciones UNIQUE permiten valores NULL. Sin embargo, de la misma forma que cualquier valor incluido en una restricción UNIQUE, solo se admite un valor NULL por columna.

Restricciones de tabla y de columna

Realmente en SQL existen restricciones de columna y restricciones de tabla. La restricción de columna se especifica como parte de una definición de columna y sólo se aplica a esa columna. Las restricciones de tabla se declaran de forma independiente de la definición de las columnas y se pueden aplicar a varias columnas de la tabla. Las restricciones de tabla se

deben utilizar cuando una restricción tiene que incluir varias columnas. (technet.microsoft, 2015)

1.6.2 Restricción de la integridad

1.6.2.1 Definición

En una base de datos existen diversos tipos de restricciones, dentro de ellos existen las restricciones de la integridad. La integridad de datos se define y se entiende de varias formas y depende cada autor y desarrolladores.

La integridad de datos puede referirse a (Leandro, 2015):

- ***Integridad de datos en general:*** hace referencia a que todas las características de los datos (reglas, definiciones, fechas, etc) deben ser correctos para que los datos estén completos.
- ***Integridad de datos en bases de datos:*** Integridad de datos se refiere al estado de corrección y completitud de los datos ingresados en una base de datos.

Según (Ecured.cu, 2015a) la integridad en una base de datos es la corrección y exactitud de la información. Además de conservar la seguridad en un sistema de bases de datos que permite el acceso a múltiples usuarios en tiempos paralelos.

Otro criterio, en la tesis de (Alain, 2014) se definió:

El término integridad es entendido de diversas maneras en la literatura técnica. Para Date (2008; 2009), por ejemplo, la integridad es un estado de la base de datos en el cual no se viola ninguna de las restricciones de integridad definidas, concepto que en (Olivé, 2007) se conoce bajo el término de consistencia de la base de datos. Para Olivé la integridad es un concepto aún más amplio, relacionado con la validez y la completitud. Se podrá afirmar entonces que los datos tienen integridad si todos los hechos contenidos en el sistema de información son válidos y están presentes aquellos considerados como relevantes. Nótese que un sistema podrá estar en un estado consistente y no poseer el grado máximo de

integridad si, por ejemplo, ciertas restricciones de integridad que debieran estar no se han incorporado en el sistema. En cambio, Date (2008; 2009) habla en términos de corrección, lo cual sería un reflejo verdadero de una base de datos del estado de cosas del mundo real, y para tener un “reflejo verdadero” es necesario que los datos registrados sean válidos y sean conocidos (lo verdadero no es desconocido, puesto que se conoce que es verdadero).

En consecuencia, las restricciones de integridad son condiciones que, bajo ciertos mecanismo creados para ello, si son satisfechas entonces se podrá tener cierta confianza sobre la integridad del sistema de información (Olivé, 2007). Date (2008; 2009) coincide también en gran medida cuando expresa que una restricción de integridad es “una expresión booleana con un nombre, que requiere ser satisfecha en todo momento”. Por lo cual: “Una restricción de integridad es una expresión basada en una condición booleana ϕ que requiere ser satisfecha en todo momento, en caso de lo cual se podrá tener cierta confianza de la integridad de un sistema de información.”

1.6.2.2 Condición de la integridad

Para obtener integridad en los datos deben cumplirse dos condiciones (Ecured.cu, 2015a):

- Las restricciones de integridad de usuario: son condiciones específicas de una base de datos concreta; son las que se deben cumplir en una base de datos articular con unos usuarios concretos, pero que no son necesariamente relevantes en otra base de datos.
- Las reglas de integridad de modelo: son condiciones propias de un modelo de datos, y se deben cumplir en toda base de datos que siga dicho modelo.

Los SGBD deben proporcionar la forma de definir las restricciones de integridad de usuario de una base de datos y una vez definida, debe velar por su cumplimiento. Las reglas de integridad del modelo, en cambio, no se deben definir para cada base de datos en concreto, porque se consideran preestablecidas para todas las bases de datos de un modelo. Un SGBD de un modelo determinado debe velar por el cumplimiento de las reglas de integridad preestablecidas por su modelo.

1.6.2.3 Reglas de la integridad

Existen cuatro reglas de la integridad de base de datos. Ellas son las siguientes (Ecured.cu, 2015a):

- **Regla de integridad de unicidad de la clave primaria:** está relacionada con la definición de clave primaria que establece que toda clave primaria que se elija para una relación no debe tener valores repetidos.
- **Regla de integridad de entidad de la clave primaria:** dispone que los atributos de la clave primaria de una relación no pueden tener valores nulos. Esta regla es necesaria para que los valores de las claves primarias puedan identificar las tuplas individuales de las relaciones. Si las claves primarias tuviesen valores nulos, es posible que algunas tuplas no se pudieran distinguir. Un SGBD relacional tendrá que garantizar el cumplimiento de esta regla de integridad en todas las inserciones y en todas las modificaciones que afecten a atributos que pertenecen a la clave primaria de la relación.
- **Regla de integridad referencial:** está relacionada con el concepto de clave foránea, lo que determina que todos los valores que toma una clave foránea deben ser valores nulos o valores que existen en la clave primaria que referencia. La necesidad de esta regla es debido a que las claves foráneas tienen por objetivo establecer una conexión con la clave primaria que referencian.
- **Regla de integridad de dominio:** está relacionada con la noción de dominio. Esta regla establece dos condiciones.
 - ✓ La primera condición consiste en que un valor no nulo de un atributo A_i debe pertenecer al dominio del atributo A_i ; es decir, debe pertenecer a dominio (A_i). Esta condición implica que todos los valores no nulos que contiene la base de datos para un determinado atributo deben ser del dominio declarado para dicho atributo.
 - ✓ La segunda condición sirve para establecer que los operadores que pueden aplicarse sobre los valores dependen de los dominios de estos valores; es

decir, un operador determinado sólo se puede aplicar sobre valores que tengan dominios que le sean adecuados.

1.6.2.4 Clasificaciones de restricciones de integridad

En (Alain, 2014) se plantea una clasificación adecuada de las restricciones de integridad, será posible descartar todas aquellas que no serán consideradas por causa del alcance de esta investigación, o porque ya son asumidas por el modelo ER.

Tanto si son reconocidas o no, las restricciones de integridad se pueden clasificar según:

- Origen: la fuente que dio origen a la formulación de la restricción.
- Ámbito: los hechos involucrados por la restricción.
- Violación: causa de la violación de la restricción.

Según el origen

Según el origen de la restricción, es posible encontrar las siguientes clasificaciones:

- Analíticas: surge por la definición o el significado de los hechos involucrados. Por ejemplo: la edad de una persona no puede ser negativa.
- Deónicas: surge de la imposición de una condición por parte de un agente autorizado. Por ejemplo: una persona no puede abrir dos o más cuentas bancarias en el mismo día.
- Empíricas: surge de una condición que contiene empíricamente el dominio sin que nadie la establezca. Por ejemplo: la edad de una persona no puede ser mayor que 200 años.

Según el alcance

Usualmente, las restricciones involucran un conjunto limitado de datos y eventos que permite clasificarlas según el alcance en cinco grupos:

- Estáticas: involucran los datos de uno y cada uno de los estados del sistema.

- De transición: involucran los datos de dos o más estados del sistema. Generalmente, restringen si la transición de un estado, pero por extensión el término se usa para aquellas que deben ser satisfechas en algunos estados.
- De evento: involucran un solo evento.
- De eventos históricos: involucran dos o más eventos que ocurren al mismo tiempo, o en momentos diferentes.
- Globales: involucran uno o más estado y uno o más eventos. Como subtipo, se distinguen las precondiciones, las cuales involucran un solo evento y el estado del sistema cuando ocurre el evento.

Según causa de violación

En cuanto a la causa de violación, se conocen dos tipos, las de violación por evento y las de violación por tiempo. Puede que parezca evidente que la causa de violación es la ocurrencia de un evento, pero una restricción también podría ser evaluada luego de que transcurrió un lapso definido de tiempo sin la ocurrencia de un evento.

En la presente investigación solo se tratarán aquellas restricciones de integridad no reconocidas por el modelo ER, estáticas y de violación por causa de eventos. El origen, en cambio, no es relevante para lograr una identificación acertada de las formas posibles de expresión de las restricciones, dado que solo se refiere al proceso mediante el cual el modelador las formuló.

1.6.2.5 Las restricciones de integridad en el análisis con el Modelo Entidad-Relación (MER)

Para (Alain, 2014), existen muchas restricciones que por su complejidad o naturaleza no son recogidas por el modelo, lo cual es visto por (Date, 2001) como una debilidad. En (Moriarty, 1997 ver Date p. 448) se afirma que “las reglas declarativas son demasiado complejas para ser captadas como parte del modelo del negocio y deben ser definidas por separado por parte del analista o desarrollador”. Sin embargo, algunas de ellas no son tan complejas. Un ejemplo podría ser una restricción de una entidad sobre valores posibles de atributos, como sucede si

se declara para cierto esquema que “el salario de una enfermera tiene que ser mayor que \$500.00”, donde el salario es un atributo del tipo de entidad enfermera.

En (Badia, 2004) se describen algunos tipos de restricciones que no son reconocidos por el modelo. Lo interesante se deriva del hecho de que no todas las restricciones se representan satisfactoriamente, dado que algunas tienen una naturaleza distinta a aquellas que sí han sido representadas. La carencia del modelo ER de proveer un mecanismo satisfactorio para la representación de las restricciones, es compensada con la escritura de las reglas en la etapa de especificación de requisitos del sistema. Esta información se pierde con frecuencia dado que la ambigüedad y la vaguedad dificultan su correcta interpretación, además de no existir métodos establecidos para incorporar tal información en el diseño.

1.7 Algunas herramientas propietarias para la generación de datos

Generalmente las herramientas de generación de datos poseen las siguientes características (iteramos, 2015):

- Flexibilidad para generar los datos de una tabla.
- Capacidad para generar pequeñas y grandes conjuntos de datos (> 1 millón de filas o más).
- Generar secuencia de comandos SQL (sentencias INSERT, UPDATE, DELETE) o en un formato de archivo adecuado para la importación masiva (que generalmente es más rápida).
- Una interfaz de línea de comandos para una fácil de secuencias de comandos.
- Extensible, de código abierto, escrito en un lenguaje dinámico.

Con las características previamente expuestas se pueden enumerar algunas herramientas para generar datos como:

1.7.1 Databene Benerator

Databene Benerator es un generador de datos de prueba que se acerca a las necesidades de los desarrolladores y que tiene características principales como:

- Puede generar los datos de una tabla existente en la definición.
- Puede generar más grande conjunto de datos (sin límite de tamaño).
- XML scripts, compatible con DbUnit, se admite la entrada de varios (CSV, Archivos Planos, DBUnit) y el formato de salida (CSV, Archivos Planos, DBUnit, XML, Excel, secuencias de comandos).
- Puede ser utilizado en la línea de comandos o a través de un plugin de maven.
- Es de código abierto y personalizable.
- Open source (GPL) Java code, acceso a muchas bases de datos directamente a través de JDBC.
- Uso mediante línea de comandos.

El benerator realmente reduce la cantidad de tiempo para crear un esquema útil de datos de una Schema de la empresa de mediana clasificada según el tamaño aplicación o XML de semanas del 2-4 para los días del 1-2!

Soporta (pero no está restringido para): El oracle, DB, IBM DB2, lenguaje normalizado de consulta de Mississippi Server, MySQL, PostgreSQL, HSQL, Derby, Firebird, XML Files, XMLSchema, CSV Files, Limas Planas, Excel Sheets.

1.7.2 GenerateData

Es una herramienta libre (expedida bajo el GNU GPL V3 Licence) y de código abierto escrito en JavaScript, PHP y MySQL que permite generar rápidamente grandes volúmenes de datos personalizados en una variedad de formatos para su uso en pruebas de software, rellenar Bases de Datos, etc.

GenerateData es un software de fuerza industrial, capaz de generar archivos de datos con millones de registros y tiene opción para generar datos en XML, Excel, HTML, CSV o SQL. GenerateData requiere:

- Hasta 512Mb Ram.
- El espacio libre del disco de alrededor 3 veces el tamaño del archivo más grande generado.
- Un sistema operativo compatible. Ventanas XP y más alto debería estar bien.
- Un JRE compatible: una Java Runtime Environment (JRE) en versión 1.6 o más alto.

1.7.3 SQL Data Generator

Es una herramienta rápida y simple para datos de prueba. Los generadores para valores de atributos pueden ser especificados a la medida e incluir generadores mediante código en Python.

Esta herramienta no es libre pero es muy interesante. Permite establecer dependencias entre las columnas y genera datos reales de los objetos de negocio, como números de teléfono, direcciones, nombres, etc.

SQL Data Generator puede poblar tablas vacías con datos de prueba que lucen auténticas. Elegir entre los datos basados en los datos existentes en otras bases de datos, criterios predefinidos, o expresiones regulares. Una API está disponible para los criterios de generación de complejos y altamente refinados de datos.

Esta herramienta tiene las características siguientes (Red Gate, 2015):

- Crea datos realistas basados en los metadatos de las columnas de la tabla.
- Generación de datos sumamente rápida.
- Sobre 60 generadores incorporados con opciones de configuración.
- Los generadores creados por el usuario dan completo control sobre los datos.
- La generación aleatoria de datos permite generar la misma recopilación de datos todo el tiempo.
- El soporte de dependencias entre tablas.
- El soporte de línea de comando para la generación automatizada de datos.
- Importa datos de fuentes existentes de datos.

- Opcionalmente desactiva triggers y restricciones para evitar interferir con lógica de la base de datos.
- Interfaz de usuario fácil para usar lo que facilita poblar una base de datos inmediatamente.

1.7.4 Banner Datadect

Es una herramienta de generación de datos que requiere Windows. No posee soporte para MySQL. Tiene interfaz gráfica de usuario pero no soporta línea de comandos o secuencias de comandos. Esta herramienta no es gratis.

1.7.5 Super Justo

Es una herramienta muy simple de usar que está diseñada específicamente para MySQL. Super Justo permite generar datos aleatorios para insertar en las tablas de la base de datos. Es personalizable, permitiendo el uso de los paquetes, o cualquier conjunto de datos de su elección.

1.8 Conexión a base de datos usando Java: JDBC

Cuando se plantea realizar un programa en Java que utiliza una base de datos, es necesario definir el acceso a esa base de datos. Aunque es sencillo, se requieren unos conocimientos intermedios. Desde ahí, el necesario conocimiento de la Conectividad de Java a Base de Datos (JDBC por sus siglas en inglés). JDBC permite conectarse a una base de datos, consultarla o actualizarla usando sentencias SQL.

Java permite conectarse a bases de datos usando SQL-92. La gran ventaja de esta utilidad es que una base de datos puede ser accedida desde el mismo programa Java (César Julio, 2015). La máquina virtual de Java es la encargada de realizar las conversiones de SQL a tipos Java de la forma más “razonable” posible.

Según (Lapuente, 2012), JDBC es creado y sostenido por la empresa Sun y es la API estándar de acceso a bases de datos con Java. Sun optó por crear una nueva API en lugar de utilizar ODBC (variante similar para windows), porque esta última presentaba algunos problemas desde ciertas aplicaciones Java. ODBC es una interfaz escrita en lenguaje C, que al no ser un

lenguaje portable, hacía que las aplicaciones Java también perdiesen la portabilidad. Además, ODBC tiene que instalarse manualmente en cada máquina, mientras que los controladores (*drivers*) JDBC que están escritos en Java son automáticamente instalables y portables. El nivel de abstracción al que trabaja JDBC es más alto que el de ODBC y, de esta forma, se pueden crear librerías de más alto nivel.

Para trabajar con JDBC es necesario tener controladores que permitan acceder a las distintas bases de datos. Sin embargo, ODBC sigue siendo hoy en día la API más popular para acceso a bases de datos, por lo que Sun se ha visto obligada a diseñar un puente que permite utilizar la API de JDBC en combinación con controladores ODBC. Esta idea se puede ver en la **Figura 2** (profesores.elo.utfsm.cl, 2015).

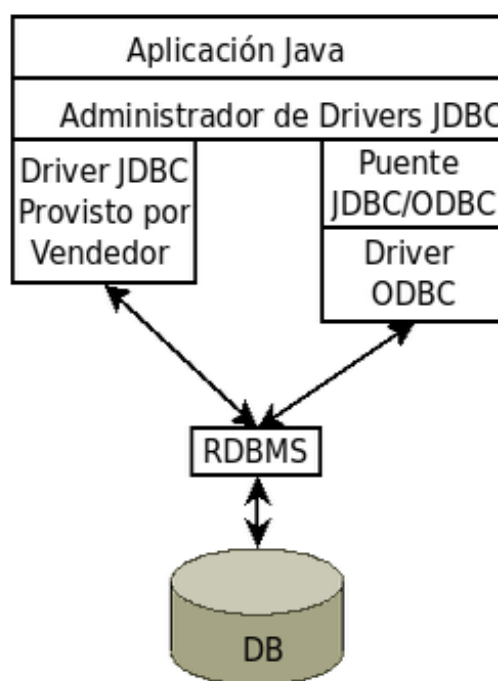


Figura 2. Administrador de Drivers

Algunas características de JDBC son las siguientes (Lapiente, 2012):

- Tecnología hecha por Sun Microsystems.
- Tiene la función de ser una capa intermedia para la aplicación con respecto a la base de datos.

- Como modelo cliente/servidor, JDBC se encuentra en el equipo cliente, conectándose directamente con la base de datos.
- Como modelo de tres capas, el JDBC se encuentra en una capa intermedia, donde todos los usuarios pasarán por él para poder acceder a la base de datos.
- Existen módulos JDBC que son propios de los fabricantes de DBMS, que son utilizados para el rápido acceso a la información de las bases de datos de los mismos.
- JDBC no se encuentra ligado a trabajar con alguna tecnología en específica, ya que se elaboró con la finalidad de ser portable.
- En aplicaciones Web, JDBC se encontrará laborando en conjunto con código HTML, mediante el mecanismo del Java script.

Capas JDBC

- Capa Aplicación (Application Layer): Conjunto de clases (API) usadas para hacer llamados a la base de datos a través de SQL, obtener y procesar los resultados.
- Capa Manejador (Driver Layer): Conjunto de clases (API) que se encargan de establecer, manejar y terminar la comunicación con un RDBMS (Implementación específica de un driver).

Conexión a la BD desde JDBC

Para representar una sesión con la conexión a la base de datos, se crea como los pasos siguientes indican:

- ***Connection conn=DriverManager.getConnection (URL);***
- ***Connection conn=DriverManager.getConnection (URL, user, password);***

El driver manager intentará encontrar el driver cargado que pueda usar el protocolo especificado en la URL. Mediante la instancia de ***Connection(conexión)*** obtenida a través de ***getConnection*** se pueden ejecutar expresiones SQL. (César Julio, 2015)

1.9 Extensión para otros orígenes mediante controladores

Todos los programas Java que se conectan a una base de datos vía JDBC, utilizan un driver o controlador de bases de datos. Es el intermediario entre la capa de negocio y la capa de base de datos y así mismo funge como el “traductor” de las sentencias Java a sentencias SQL propias del manejador de bases de datos.

Para que la aplicación Java pueda realizar conexiones con la base de datos será necesario utilizar el driver (controlador) JDBC adecuado para el sistema con el que se trabaje. Antes de realizar la conexión con la base de datos es necesario registrar el driver correspondiente al SGBD utilizado, que en cualquier caso debe implementar la interfaz *java.sql.Driver*. Esta operación se puede realizar de dos formas diferentes:

- Utilizando la variable del sistema “*sql.driver*”, variable que mantiene todas las clases de los drivers separados por comas.
- Usando el método *Class.forName (driver)*. Este suele ser el más habitual.

En esta herramienta se utiliza la segunda forma para registrar drivers. Los métodos correspondientes a cada tipo de gestor de bases de datos son los siguientes:

- *Class.forName("com.microsoft.sqlserver.jdbc.SQLServerDriver");*
- *Class.forName("com.mysql.jdbc.Driver");*
- *Class.forName("org.postgresql.Driver");*

1.10 Conclusiones parciales del capítulo

En este capítulo se ha brindado un cuadro teórico sobre el enfoque de generar datos, haciendo énfasis en los elementos principales de una base de datos lo cual tiene gran importancia en el proceso de generación de datos. Se mostraron diferentes herramientas para generar datos y se expuso los aspectos necesarios para la conexión de una herramienta a una base de datos.

CAPÍTULO 2:
“ANÁLISIS Y DISEÑO DEL
SISTEMA PARA LA
GENERACIÓN DE DATOS”

Capítulo 2. “Análisis y diseño del sistema para la generación de datos”

En el presente capítulo se abordan los aspectos que se consideran necesarios para una estrategia de generación automática de reglas usando la herramienta de generación de datos, dividiendo las etapas del desarrollo del software: análisis y diseño. Se describen los elementos principales relacionados con el sistema y se realiza un análisis del entorno orientado a objetos con UML (Lenguaje Unificado de Modelado) exponiendo los diferentes diagramas que permiten modelar, construir y documentar los elementos esenciales que forman el sistema. Finalmente se explica el diseño modelado del proceso de la generación de datos.

2.1 Análisis y diseño de sistemas

La creación o el establecimiento de un nuevo sistema de información puede ser una tarea compleja. Con este tipo de situaciones existe un proceso de análisis y diseño de sistemas que auxilia en la resolución de tales problemas. El análisis y diseño de sistemas es una guía útil que permite estructurar el proceso de desarrollo de proyectos de software. Este proceso permite aumentar la productividad personal y resuelve los problemas en un área de trabajo.

Según (Pressman et al., 2003) esta etapa es de vital importancia en el ciclo de vida del proyecto, pues este procedimiento permite reducir al mínimo el riesgo de fracaso de nuevos proyectos, ya que la instalación de un sistema sin la adecuada planificación puede conducir a grandes frustraciones y ocasionar que el mismo sea subutilizado, o peor aún, deje de ser utilizado al no cumplir con las expectativas que le dieron origen.

2.2 Términos necesarios en el proceso de generar datos

Los términos de este sistema y las reglas extraídas se muestran en el diagrama de Entidad Relación, donde cada término del sistema se presenta como una entidad del modelo, el cual

contiene siete entidades. Un buen diseño de la base de datos le imprime un carácter flexible a la información. En la **Figura 3** se muestra el diagrama Entidad Relación reflejando los términos conceptuales necesarios para el sistema diseñado.

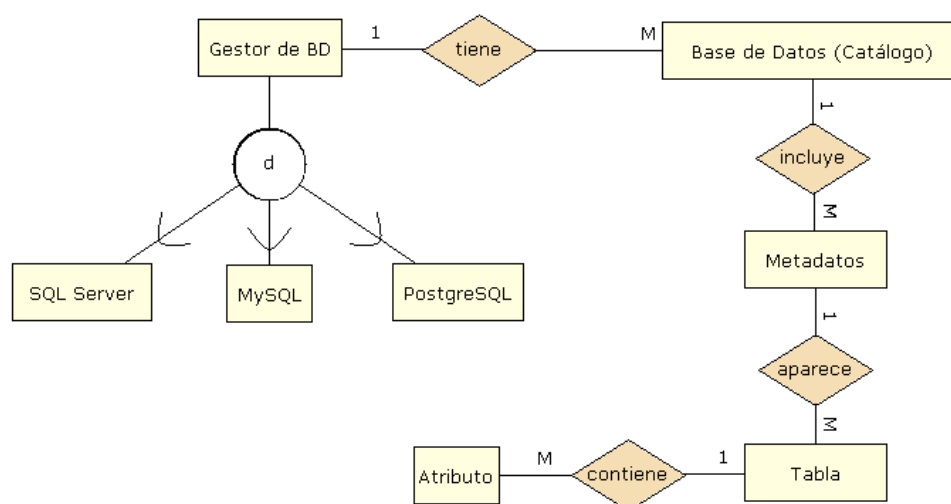


Figura 3. Diagrama de Entidad Relación de una base de datos

A partir de la descripción de la herramienta, se descubren los elementos de una base de datos que se necesitan en el proceso de generar datos. Los términos principales que se establecen son:

- **Gestor de Base de Datos:** en el presente proyecto, se trabajan con tres tipos Gestores de Base de Datos – SQL Server, MySQL y PostgreSQL. El usuario o el desarrollador trabaja con el tipo de gestor seleccionado en dependencia de su objetivo o el objetivo del cliente.
- **Base de datos (o Catálogo del sistema):** se refiere a una colección de tablas que contienen los datos describiendo la estructura de una base de datos. Para el proceso de generar datos, es necesario extraer y almacenar la información de las tablas del sistema que se encuentra en su catálogo.
- **Metadatos:** se refiere a un grupo de datos que describen otros datos. Con el acceso a los metadatos se puede conocer la estructura de la base de datos pero no su contenido. Mediante la herramienta de generación de datos, los desarrolladores pueden extraer

y procesar uno o varios metadatos, en dependencia de sus objetivos. En nuestro proyecto, se puede trabajar con varias bases de datos a la vez, es por lo cual no se detallan los metadatos de una base de datos.

- **Tabla:** se refiere a los metadatos de una tabla con sus columnas y propiedades. Las propiedades de cada tabla de la base de datos se describen en los atributos de cada tupla.
- **Atributo:** se refiere a los metadatos de un atributo de una tabla específica. Cada atributo de este tipo de entidad refleja una propiedad de los atributos de una tabla.

A partir de los términos anteriores se obtienen las reglas esenciales siguientes:

- Un tipo de Gestor de Base de Datos puede tener uno o muchos Catálogo del sistema.
- Una base de datos (Catálogo del sistema) incluye uno o muchos Metadatos.
- Los Metadatos aparecen en una o muchas tablas.
- Una Tabla contiene al menos uno o muchos atributos.

2.3 Diagrama de actores y casos de uso

A partir de la descripción del problema, se elabora el siguiente diagrama de actores y casos de uso del sistema, identificándose un actor que es un desarrollador y luego definiéndose sus casos de uso según los roles en que participe. Este diagrama de casos de uso se utiliza para describir el comportamiento del sistema al afrontar una tarea de negocio o un requisito de negocio. Desde allí se puede tener un entendimiento común entre el cliente/usuario y el equipo de desarrollo. Los casos de uso del actor de este sistema se identifican en la **Figura 4**.

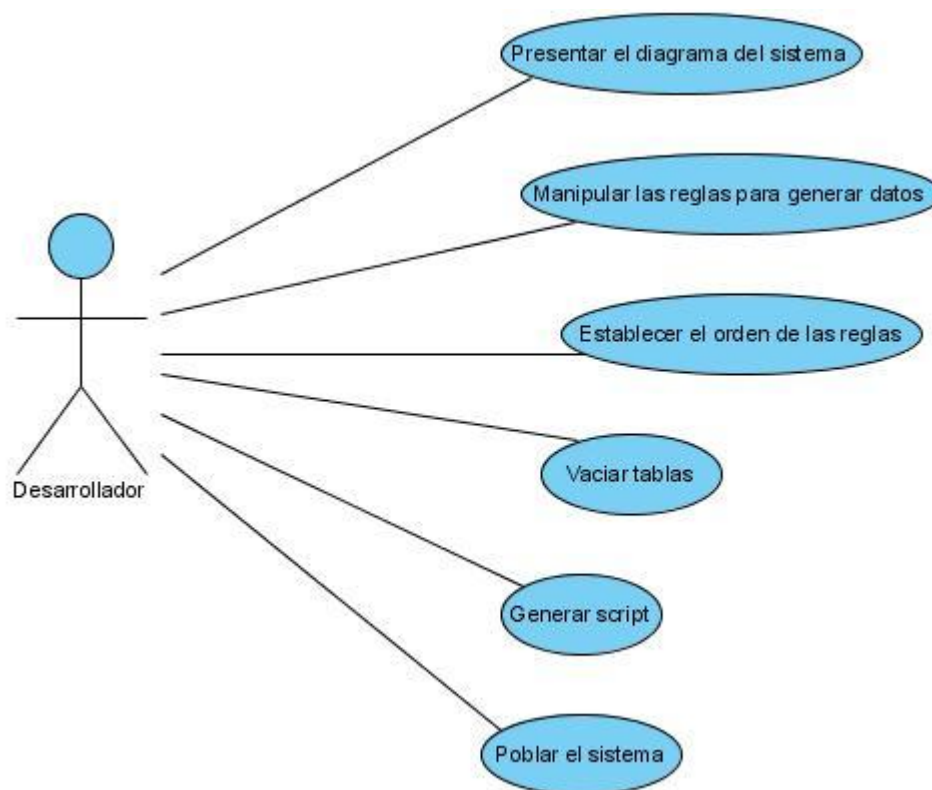


Figura 4. Diagrama de Casos de uso

2.4 Descripción del caso de uso “Manipular las reglas para generar datos”

Después de conectar con la base de datos exitosamente, el actor desarrollador puede crear una regla a generar datos: selecciona la opción “Crear regla” o se puede seleccionar dentro del menú “Reglas” la opción “Crear regla”, luego se muestra una ventana para realizar los pasos para crear una regla, luego se hace clic en el botón “Terminar” para guardar la regla.

Caso de Uso	Crear regla para generar datos
Actores	Desarrollador

Propósito	Crear regla en dependencia del objetivo del desarrollador (<i>Inserción, Actualización, Eliminación</i>) para generar script los datos en la base de datos.
Resumen	Con el objetivo de desarrolladores o de los clientes, se elaboran reglas para generar los datos de una fuente de datos. Existen tres tipos de reglas: <i>Inserción, Actualización y Eliminación</i> que ayudan a lograr los resultados deseados.

Curso normal de los eventos a crear reglas de generar datos

¿QUÉ HACE EL ACTOR?	¿QUÉ HACE EL SISTEMA?
Dar clic en el botón “Crear regla” o se puede seleccionar dentro del menú “Reglas” la opción “Crear regla”.	Se muestra una ventana con la lista de tablas que permite elegir una tabla con una operación <i>Inserción, Actualización o Eliminación</i> depende el objetivo del desarrollador.
Selecciona una tabla con una operación. Si se elige <i>Inserción</i> , se da clic en el botón “Siguiente”.	Se muestra una ventana con la lista de las columnas de la tabla seleccionada anteriormente con sus tipos de atributos, las restricciones de Null o Not Null y la llave primaria. Además se muestra una lista de métodos a utilizar.
Si se elige <i>Actualización</i> , se da clic en el botón “Siguiente”.	

<p>Si se elige <i>Eliminación</i> hay que introducir la cantidad de filas que se quieren eliminar. Luego se da clic en el botón “Terminar”.</p>	<p>Se muestra una ventana que permite elegir el campo (lista de columnas) para actualizar.</p> <p>Guardar la regla.</p>
<p><u>De caso <i>Inserción</i>:</u></p> <p>Selecciona un método correspondiente con el tipo de atributo de cada columna (<i>Valores aleatorios, Lista de valores, Incremental, Secuencia SQL</i>) e inserta los datos necesarios. Luego se da clic en el botón “Siguiente”.</p> <p><u>De caso <i>Actualización</i>:</u></p> <p>Elige el campo a actualizar. Luego se da clic en el botón “Aceptar”.</p> <p>Modifica los valores de los atributos deseados con los métodos correspondientes (<i>Valores aleatorios, Lista de valores, Incremental, Secuencia SQL</i>). Luego se da clic en el botón “Siguiente”.</p>	<p>Se muestra la vista previa para revisar si se introducen correctamente los datos o no.</p> <p>Se muestra una ventana que contiene la lista de columnas y de métodos a utilizar como es el caso de <i>Inserción</i>.</p> <p>Se muestra la vista previa para revisar si se actualiza correctamente los datos o no.</p>
<p>Dar clic en el botón “Siguiente”.</p>	<p>Se muestra una ventana de <i>Opciones</i> que permite introducir número de filas deseadas a generar en la tabla seleccionada.</p>

Introduce la cantidad de filas deseadas a generar. Dar clic en el botón “Siguiente”.	Se muestra una ventana que permite guardar el resultado en formatos.
Dar clic en el botón “Terminar”.	Guardar la regla.

Tabla 1. Tabla de eventos para el caso “Manipular las reglas a generar datos”

2.5 Descripción del caso de uso “Vaciar tabla”

Para agregar una regla para vaciar tabla: selecciona la opción “Agregar regla para vaciar tabla”, donde se muestra una ventana *Vaciar tabla* que contiene una lista de tablas de la base de datos, se selecciona una o muchas tablas que se quieran vaciar, luego se da clic en el botón “Aceptar” y se guarda la regla.

Caso de Uso	Agregar regla para vaciar tabla
Actores	Desarrollador
Propósito	Vaciar una o varias tablas seleccionadas
Resumen	Elaborar una regla que se selecciona una o varias tablas que quería vaciar el desarrollador, desde allí se pueden llenar otros datos a la base de datos.

Curso normal de los eventos a vaciar tabla

¿QUÉ HACE EL ACTOR?	¿QUÉ HACE EL SISTEMA?
Dar clic en el botón “Agregar regla a vaciar tabla”.	Se muestra la ventana que tiene una lista de las tablas de la base de datos.

Selecciona una o varias tablas en la lista de tablas. Si la selección está mal, da clic en el botón “Limpiar selección” y vuelva a escoger las tablas deseadas.	Actualiza las seleccionadas.
Dar clic en el botón “Aceptar”.	Guardar la regla.
Con la regla creada, se puede eliminarla. Da clic en el botón “Eliminar regla” o se puede seleccionar dentro del menú “Reglas” la opción “Eliminar regla”.	Se muestra una ventana a confirmar si está seguro a eliminar la regla o no. Si la selección es “Yes”, se elimina la regla.

Tabla 2. Tabla de eventos para el caso “Vaciar tabla”

2.6 Diagrama de actividad principal del sistema

En este diagrama de actividades se presentan los flujos del trabajo paso a paso del proceso y las interacciones de los componentes en el sistema. Se profundiza cada caso en los detalles internos que ayudan a entender el comportamiento de la ejecución del sistema. Las acciones de flujos del trabajo se identifican en la **Figura 5**.

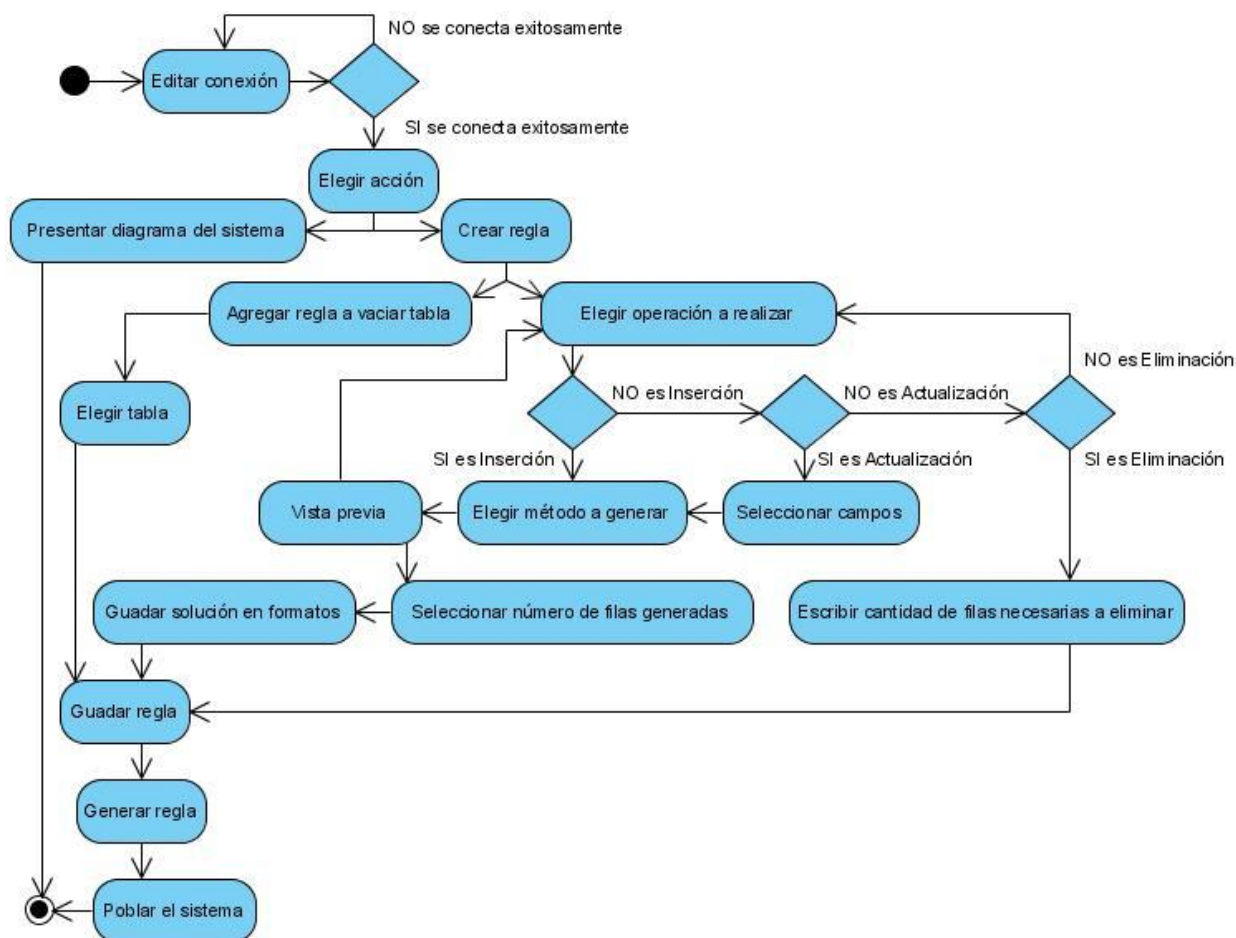


Figura 5. Diagrama de actividad principal

2.7 Otros diagramas que componen el modelado del software

Los demás diagramas que componen el modelado del software son los siguientes:

2.7.1 Diagrama de actividades para el caso de uso “Vaciar tabla”

Se presenta el flujo del proceso que cómo crear una regla para vaciar datos de una tabla seleccionada. Aunque se modela esta parte en el diagrama de actividades completo, aquí se muestra nuevamente a través de un flujo más detallado. El diagrama se identifica en la **Figura 6**.

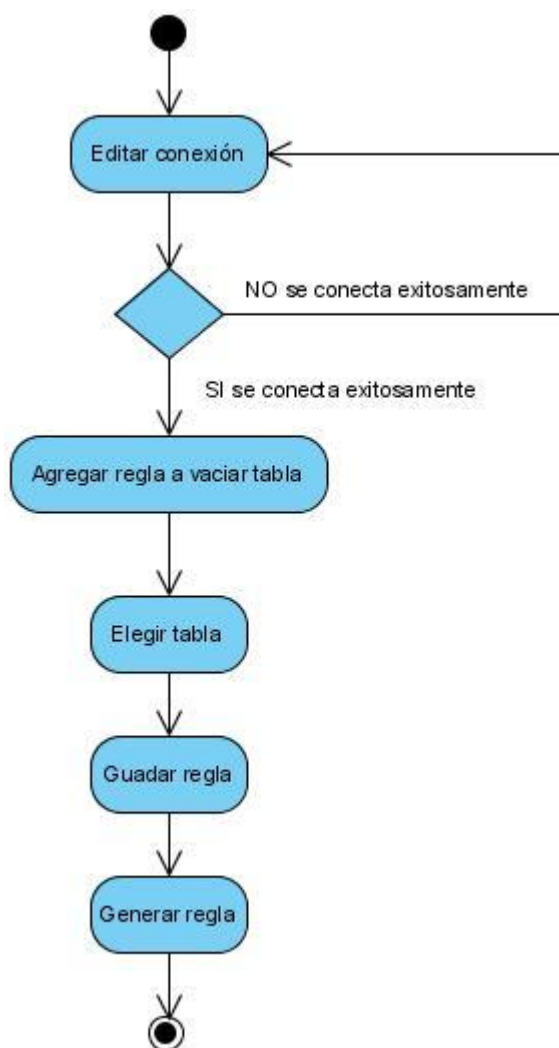


Figura 6. Diagrama de actividades para el caso de uso "Vaciar tabla"

2.7.2 Diagrama de actividades para el caso de uso “Establecer el orden de las reglas”

En este diagrama se muestra un flujo de proceso para ordenar la posición de las reglas creadas (**Figura 7**). Este proceso permite a los desarrolladores ordenar su lista de reglas en dependencia de sus objetivos y les ayuda a revisar y trabajar con rapidez y de modo efectivo. Una regla se puede subir o bajar para tener una lista organizada.

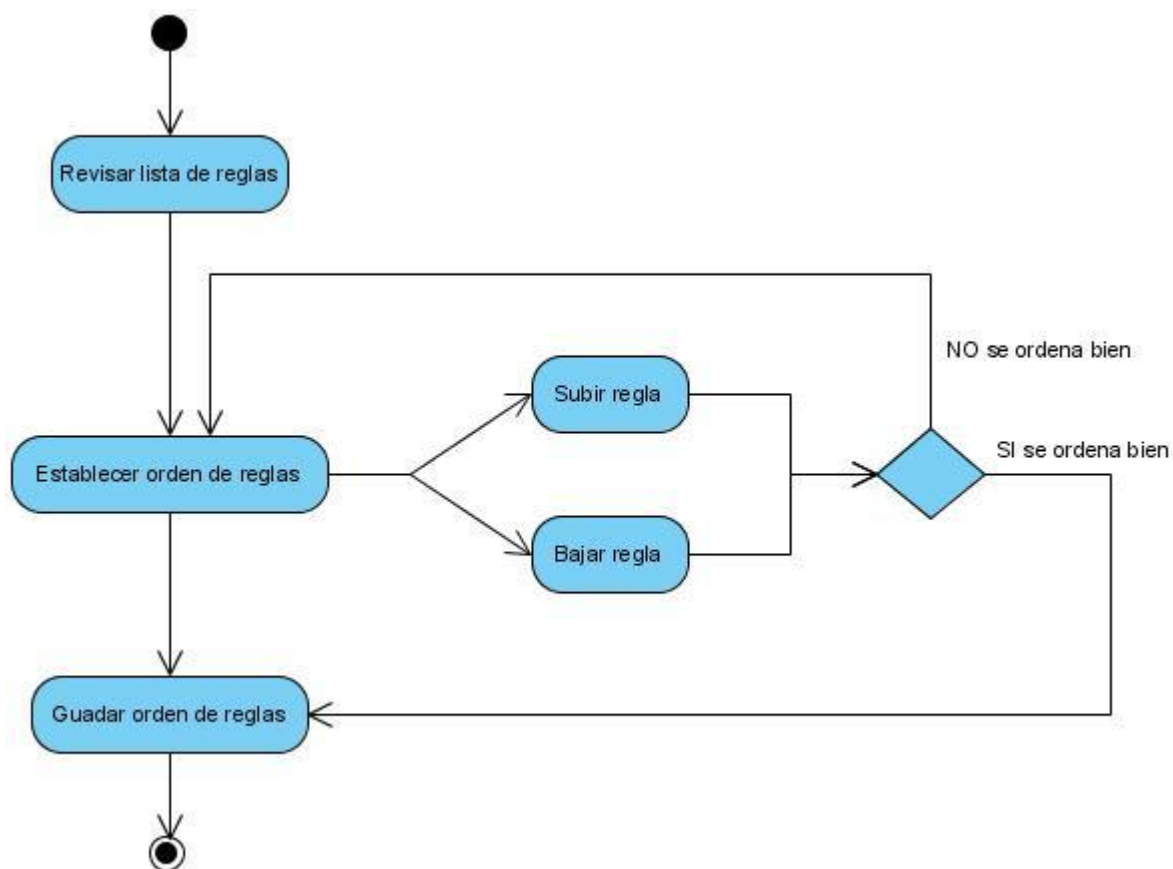


Figura 7. Diagrama de actividades para el caso de uso "Establecer orden de las reglas"

2.7.3 Diagrama de estados para manipular reglas

Este diagrama de estados describe la estructura de un sistema y proporciona al usuario una idea de cómo navegar en el sistema. Es un diagrama utilizado para identificar cada una de las rutas que puede tomar un flujo de información. Se permite visualizar de una forma secuencial la ejecución de cada uno de los procesos. En este diagrama se puede ver claramente que en la lista de reglas creadas, las reglas para vaciar tabla solo se pueden eliminar, mientras que las reglas para generar datos se pueden modificar y eliminar. Todos se describen en la **Figura 8**.

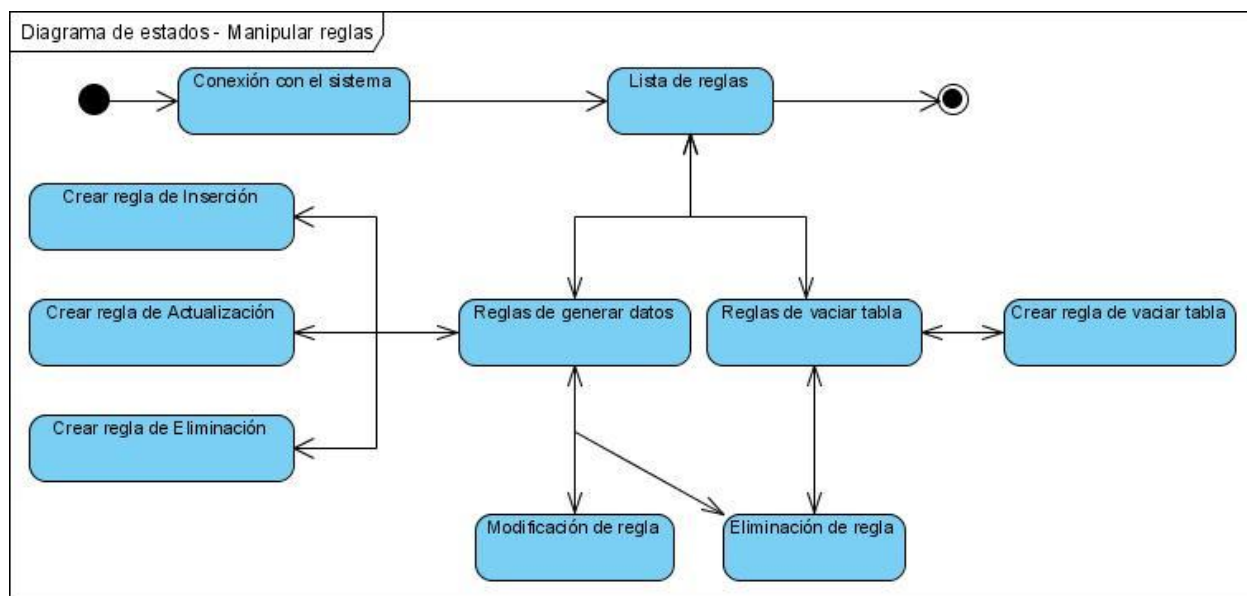


Figura 8. Diagrama de estados para “Manipular reglas”

2.7.4 Diagrama de estados para ordenar reglas

En este diagrama de estados se describe una forma para ayudar a los desarrolladores a establecer el orden de las reglas creadas. El sistema se caracteriza por un conjunto de estados válidos para cada caso a ordenar, los cuales se presentan en la **Figura 9**.

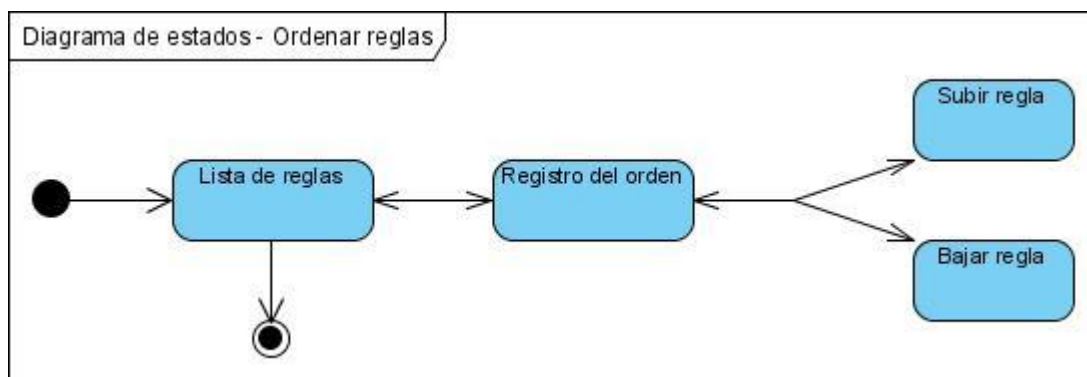


Figura 9. Diagrama de estados para “Ordenar reglas”

2.7.5 Diagrama de despliegue

Los diagramas de despliegue o de distribución muestra la disposición física de los distintos nodos que componen un sistema y el reparto de los componentes sobre los mismos. Los nodos se utilizan para modelar la topología del hardware sobre el que se ejecuta el sistema (Rodríguez, 2005)

Este diagrama muestra la configuración de nodos de procesamiento en tiempo de ejecución y las instancias de componentes y objetos que se encuentran dentro de esos nodos. La **Figura 10** muestra el diagrama de despliegue del sistema:

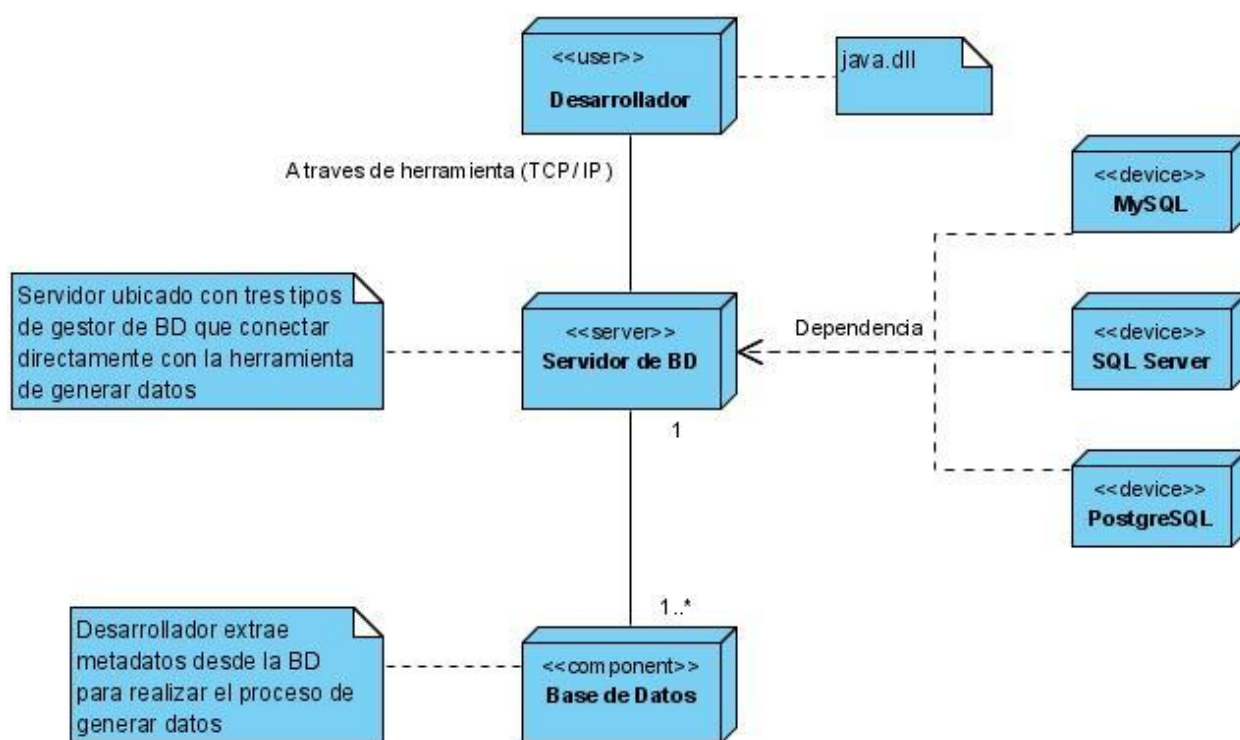


Figura 10. Diagrama de despliegue del sistema

2.8 Diferentes formas a generar valores de un atributo

Un atributo es una especificación que define una propiedad de un objeto o un elemento. Para mayor claridad, los atributos deben ser considerados más correctamente como metadatos. En

el presente trabajo, se utilizan cuatro métodos para generar valores de un atributo en la base de datos. Se describen las características de los métodos de la siguiente manera:

1. **Valores aleatorios:** este método permite a los usuarios seleccionar un intervalo de valores, en dependencia del tipo del atributo que estén trabajando. El objetivo es que se genere el atributo con diferentes valores en ese intervalo.
2. **Lista de valores:** con este método los usuarios tienen que agregar los valores al atributo, en dependencia de su tipo. Si se inserta mal un valor, se puede eliminar. Si se tiene una lista de valores, se puede limpiar y se empieza a agregar un nuevo valor con sus objetivos.
3. **Incremental:** permite generar valores enumerados con un orden. Se empieza con un valor inicial y el incremento es la distancia entre los valores consecutivos de un atributo. El sistema lo genera correctamente con la distancia del incremento.
4. **Secuencia SQL:** este método permite utilizar sentencias en SQL para obtener los valores de un atributo. En este método, los usuarios tienen dos opciones para realizar su trabajo: usan la conexión actual o pueden conectarse con otro servidor de la base de datos. Si quieren establecer una conexión con otro servidor, solo se selecciona la opción “Usar la conexión actual” y el sistema muestra una ventana que permite conectar con otro servidor.

Las diferentes formas para generar valores de un atributo que se describieron anteriormente son los métodos más útiles en la generación de datos.

2.9 El paradigma Modelo – Vista – Controlador (MVC) del sistema

Modelo Vista Controlador (MVC) es un estilo de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos. El modelo es el Sistema de Gestión de Base de Datos y la Lógica de negocio, y el controlador es el responsable de recibir los eventos de entrada desde la vista.

El modelo MVC de este sistema separa totalmente las tres capas donde el acceso de datos es fundamental para el desarrollo de arquitecturas consistentes, reutilizables y más fácil de mantener. Se pueden agregar nuevas vistas, nuevas formas de recolectar las órdenes de usuario, modificar los datos de este sistema sin necesidad de provocar que todo el sistema se paralice. Al final resulta un modelo bien sencillo a comprender y ahorra tiempo para el desarrollo de proyectos posteriores.

A partir de la funcionalidad de la herramienta con el objetivo de los usuarios, se elabora el funcionamiento de MVC como se muestra en la **Figura 11**.

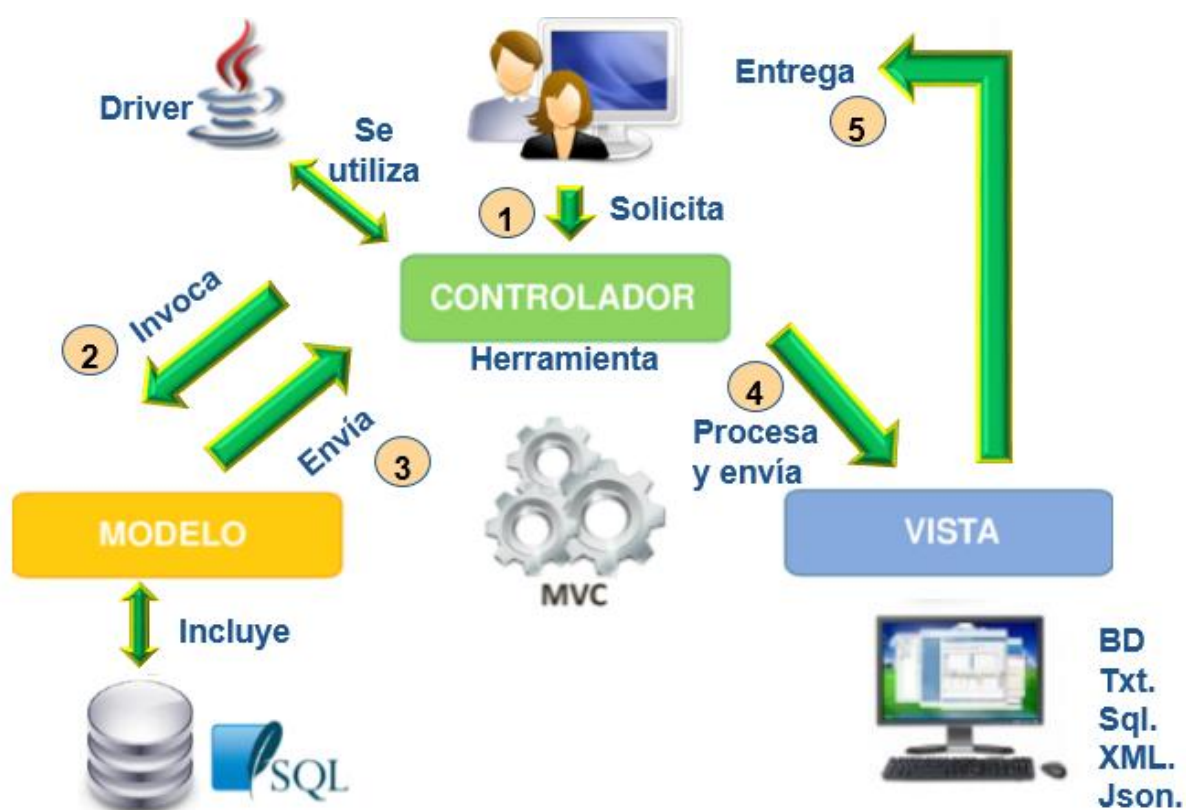


Figura 11. Modelo Vista Controlador del sistema

En esta Figura se pueden identificar claramente tres capas con cinco pasos del procedimiento MVC del sistema. Se describen de la siguiente manera:

1. Los usuarios solicitan al Controlador que se encuentra en la herramienta de generación de datos. Esta herramienta utiliza los drivers en Java.

2. Después de solicitar la conexión al Controlador, los usuarios invocan al Modelo que incluye las bases de datos SQL.
3. Se extraen los metadatos del Gestor de Base de Datos (que puede ser de varios tipos) y se las envía al Controlador.
4. Con la herramienta, los datos se procesan y cuando termine el proceso se envían a la Vista. Desde la Vista, los usuarios y los clientes pueden revisar los datos en la fuente de datos, también en diferentes formas de texto como: Txt, SQL, XML y Json.
5. Los usuarios obtienen los resultados necesarios cuando revisan la Vista.

2.10 Extraer metadatos desde una base de datos

Cuando se desarrolla una aplicación con JDBC, una de las tareas que se programan frecuentemente es pasar un objeto de tipo *ResultSet* a una lista de objetos. Para acceder a los metadatos de un objeto *ResultSet* se utiliza el método *getMetadata ()*.

Para obtener las tablas que tiene la base de datos, se realizan los pasos siguientes:

1. Elige el gestor que se quiere para trabajar.
2. Se conecta a la base de datos.
3. Se utiliza el método *getCatalog ()* para obtener el catálogo de la base de datos.
4. Se obtiene el esquema de la base de datos.
5. Se obtienen las tablas del catálogo del sistema.

A veces se necesitan hacer aplicaciones en Java que trabajen con una base de datos desconocida, de la que no se conoce qué tablas, ni qué columnas contiene. En *java*, las clases *DataBaseMetaData* y *ResultSetMetaData* permiten, respectivamente, analizar la estructura de una base de datos (qué tablas tiene, que columnas cada tabla, de qué tipos, etc.) o de un *ResultSet* de una consulta, para averiguar cuántas columnas tiene dicho *ResultSet*, de qué columnas de base de datos proceden, de qué tipo son, etc.

DataBaseMetaData

Para ver qué catálogos, esquemas, tablas, tipos de tablas, columnas de las tablas y demás de una base de datos, se tiene la clase *DataBaseMetaData*. Una vez establecida la conexión, se puede obtener la instancia correspondiente de dicha clase con el método *getMetaData ()* de la clase *connection*.

```
DatabaseMetaData metaData = connection.getMetaData();
```

Con este método *getMetadata ()* se obtienen los metadatos que se necesitan para generar datos.

ResultSetMetaData

El objeto *ResultSetMetaData* es utilizado para obtener información de los datos y propiedades de las columnas contenidas como resultado en un objeto *ResultSet*. En pocas palabras contiene los metadatos (los datos de los datos) como pueden ser: número de columnas, nombres de columnas, tipos de datos de las columnas, etc.

A partir del *ResultSet*, se puede obtener el *ResultSetMetaData*, al que se interroga sobre lo deseado.

```
ResultSetMetaData metaData = rset.getMetaData();
```

2.11 Conclusiones parciales del capítulo

En este capítulo se realizó un estudio sobre el diseño lógico del sistema para comprender cómo funciona la herramienta. Se abordaron los elementos básicos con los cuatro métodos necesarios que ayudan a crear una regla de generación de datos correspondiente a las operaciones *Inserción* y *Actualización* para una base de datos. También se describió brevemente desde una óptica de análisis y diseño cómo se registran los drivers en Java y cómo se extraen los metadatos desde su catálogo para utilizar en el proceso de creación de las reglas necesarias.

CAPÍTULO 3:
“CARACTERÍSTICAS Y USO
DE DAGENSQL”

Capítulo 3. “Características y uso de DaGenSQL”

En este capítulo se describe una presentación al usuario de las funcionalidades del sistema a trabajar en la herramienta de generación de datos. A partir de los casos de uso diseñados, se expone la forma de trabajar y las posibles acciones que se pueden realizar en la herramienta. Se efectúa una validación detallada de las opciones y modo de uso en cada caso para establecer formas de generar datos, ayudando al usuario a trabajar con la herramienta de modo efectivo y rápido.

3.1 El uso del sistema

En el desarrollo del Sistema de Información, se plantea esta herramienta para generar automáticamente grandes volúmenes de datos en diferentes gestores de bases de datos. Desde la base de datos, se extraen los metadatos necesarios, y cuando termina el proceso de generar datos, se puebla la base de datos del sistema. Los usuarios obtienen los datos desde el sistema publicado. El uso de esta herramienta se describe en la **Figura 12**.



Figura 12. El uso del sistema

3.2 Interfaz visual del sistema

3.2.1 Vista del logotipo del sistema

La implementación de la interfaz visual responde al diseño realizado en correspondencia con las necesidades de los usuarios, esta se aprecia en la **Figura 13**.



Figura 13. Vista logotipo del sistema

Se puede observar que esta interfaz es muy sencilla, la cual se describe a continuación:


- ✓ La herramienta es capaz de generar datos desde diferentes gestores de base de datos entonces se le denominó con el nombre **DaGenSQL**.
- ✓ **La barra de Menú:** muestra los menús que permiten abrir la pantalla para trabajar con el sistema, también pueden revisar las informaciones necesarias sobre la herramienta.
- ✓ **La imagen de la herramienta:** se muestra el logotipo con el nombre de tres tipos de gestores necesarios a trabajar en este sistema.

3.2.2 Crear la conexión con la base de datos

Desde la pantalla de logotipo, se selecciona dentro del menú “Abrir” la opción “Abrir pantalla principal” para mostrar la pantalla de esta herramienta que permite conectar al servidor como la **Figura 14** indica.



Figura 14. Conectar al servidor

Dar clic en el botón  “Conectar al servidor” o se puede seleccionar dentro del menú “Archivo” la opción “Conectar” para establecer la conexión con la base de datos desde una fuente de datos probada. Se muestra la ventana de *Crear Conexión* en la **Figura 15**.

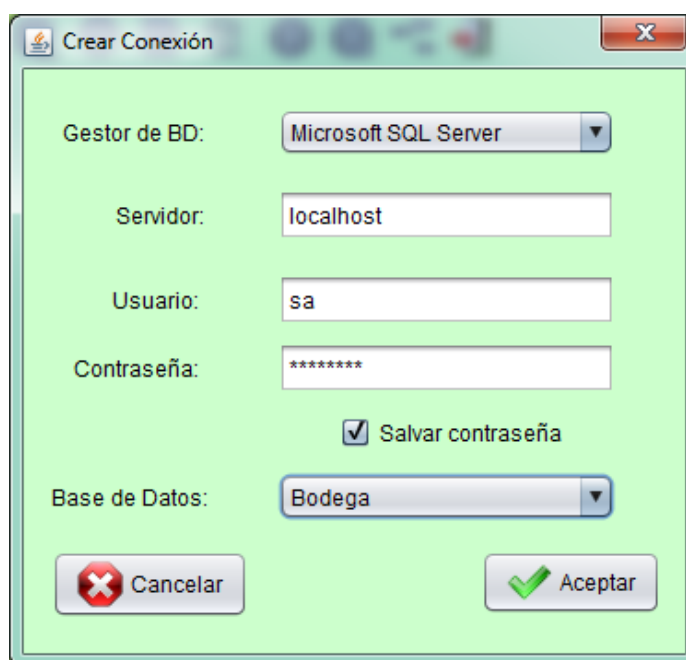


Figura 15. Crear Conexión

En esta ventana, se puede seleccionar uno dentro de tres tipos de gestores de base de datos: *Microsoft SQL Server*, *MySQL* o *PostgreSQL* depende del usuario. En la zona del *Servidor* se introducen datos como “localhost” o “127.0.0.1” o la dirección IP de la máquina en la que se está conectando. Luego se introducen datos de *Usuario* y *Contraseña* que son los mismos datos usados en el gestor a conectar.

Cuando se llena toda la información necesaria, aparecen los nombres de las bases de datos desde la fuente de datos. Se elige una base de datos y se da clic en el botón “Aceptar” para comenzar la conexión.

3.2.3 Opciones de la herramienta

En cuanto se conecta bien con la base de datos, se muestra la ventana principal de esta herramienta como indica **Figura 16**.

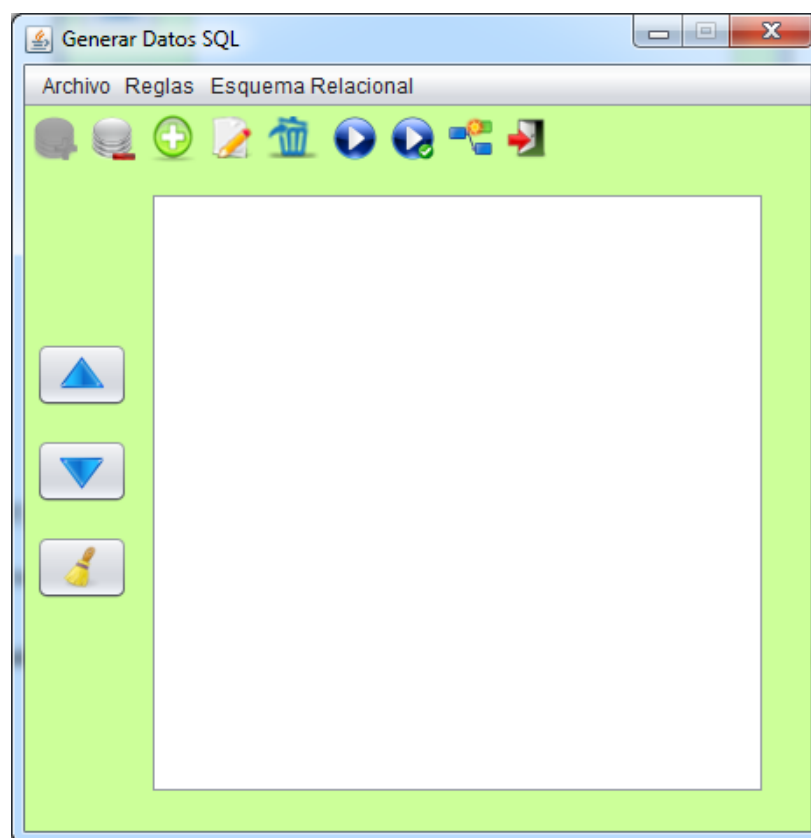


Figura 16. La pantalla principal de la herramienta “Generar datos SQL”

Se tuvo en cuenta que la pantalla principal de esta herramienta contiene:

- **Menú principal:** muestra los diferentes menús a los que tiene acceso el usuario.
- **Submenú asociado a menú principal:** cuando una de las opciones del menú principal es seleccionada se muestran las sub-opciones asociadas a este.
- **La barra rápida de opciones:** le ayuda al usuario a trabajar directo y más rápido con los botones de opciones sin buscarlos en menú principal.
- Por la izquierda contiene los botones de establecer el orden de las reglas y para agregar reglas de vaciar tablas. El centro de la pantalla principal es la zona blanca para guardar las reglas creadas.

Las opciones rápidas de esta herramienta se describen como siguiente:



Conectar al servidor: permite realizar la conexión con la base de datos.



Desconectar del servidor: permite cerrar la conexión actual.



Añadir regla: permite crear regla de diferentes formas como *Inserción*, *Actualización* o *Eliminación*.



Modificar regla: permite editar una regla necesaria. Se tiene en cuenta que en esta herramienta se modifica las reglas de generar datos (*Inserción*, *Actualización* y *Eliminación*), no las reglas de vaciar tablas.



Eliminar regla: permite eliminar una regla.



Ejecutar reglas: permite ejecutar una o varias reglas (marcadas o no).



Ejecutar reglas chequeadas: permite ejecutar una o varias reglas pero tiene que ser marcada.



Generar Esquema Relacional: muestra el esquema de la base de datos.



Salir: permite cerrar la ventana principal a la ventana de logotipo del sistema.



Subir regla: permite subir una regla seleccionada.




Bajar regla: permite bajar una regla seleccionada.



Agregar regla para vaciar tablas: permite crear regla para vaciar tablas.

3.2.4 Esquema relacional de la base de datos obtenido desde la herramienta

Desde la pantalla principal de la herramienta, se da clic en el botón  “Generar Esquema Relacional” para ver el esquema de la base de datos conectada. Desde este esquema aparecen las tablas con: el nombre de la tabla, la llave Primary Key, las llaves Foráneas, el nombre de cada columna con su tipo de atributo.

Es muy interesante que se puede mover el esquema completo y también mover cada tabla con las interrelaciones entre las tablas. A continuación se muestra un ejemplo de la base de datos “*Bodega*” conectada de gestor *Microsoft SQL Server* por esta herramienta, se describe en la **Figura 17**.

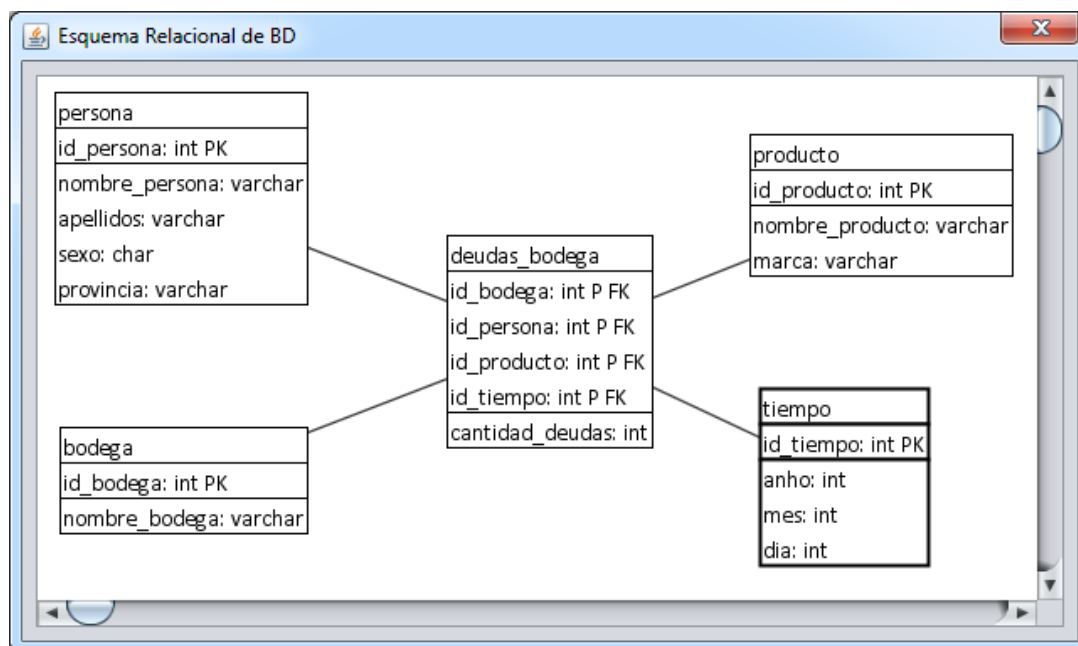



Figura 17. Esquema Relacional de la base de datos “Bodega”

3.3 Crear reglas de generar datos de diferentes formas

La herramienta puede trabajar con tres tipos de gestores: *Microsoft SQL Server*, *MySQL*, *PostgreSQL*. En este caso de prueba, se va a trabajar con la tabla “*persona*” de la base de datos “*Bodega*” que se conecta desde gestor *Microsoft SQL Server*.

3.3.1 Agregar regla para vaciar tablas

Cuando el usuario quiere crear la regla para vaciar tablas, se da clic en el botón  “Agregar regla para vaciar tablas”, se muestra la ventana como la **Figura 18** indica.

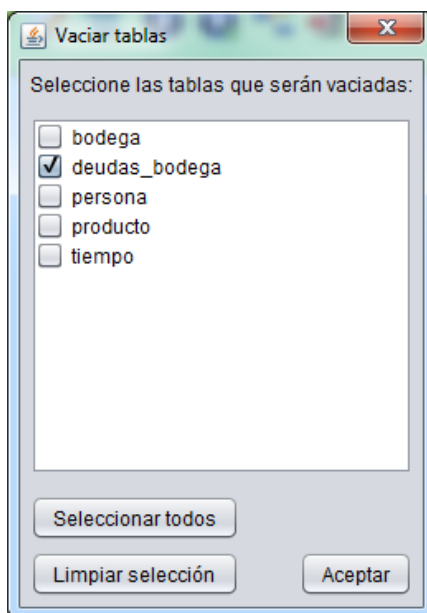





Figura 18. Vaciar tablas

De modo que aparece la lista de tablas que contiene la base de datos conectada y al mismo tiempo permite seleccionar las tablas que serán vaciadas. Se puede marcar una o varias tablas para vaciar. “Seleccionar todos” permite marcar todas las tablas. “Limpiar selección” permite eliminar la tabla marcada. Con el botón “Aceptar” se obtiene una regla creada para vaciar tabla.

La regla para vaciar tabla se guarda en la zona principal de la pantalla. Si no se marca, se le ejecuta con el botón  “Ejecutar reglas”. En caso la regla marcada como la **Figura 19** indica, se le ejecuta bien cuando da clic en uno de dos tipos de botones  “Ejecutar reglas” o bien cuando se da clic en  “Ejecutar reglas chequeadas”.

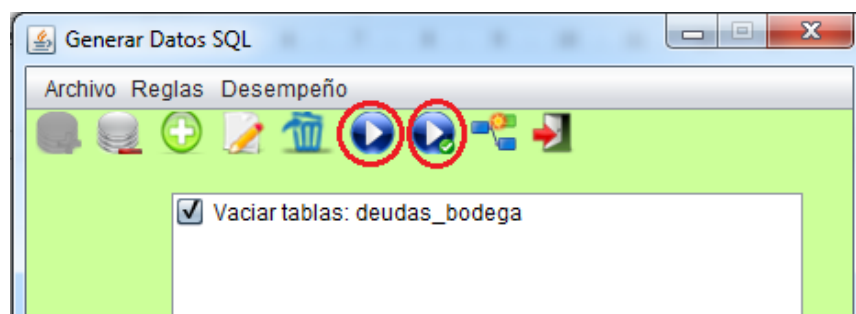



Figura 19. Ejecutar regla marcadas

Cuando la regla se ejecuta correctamente, se muestra una ventana de mensaje de ejecución exitosa.

3.3.2 Crear regla de Inserción

Desde la herramienta se puede insertar grandes volúmenes de datos a la base de datos conectada; método más efectivo y rápido que la inserción “a mano”. Para realizar este objetivo, se presiona el botón  “Añadir regla” o también se puede seleccionar dentro del menú “Reglas” la opción “Añadir regla”. Inmediatamente se muestra una ventana con la ficha *Tablas* como la primera opción dentro de un asistente para crear una regla que contiene además las fichas: *Atributos*, *Vista Previa*, *Opciones* y *Guardar Resultados*. Esta opción inicial permite seleccionar la tabla deseada para insertar los datos. Este paso se muestra en la **Figura 20**.

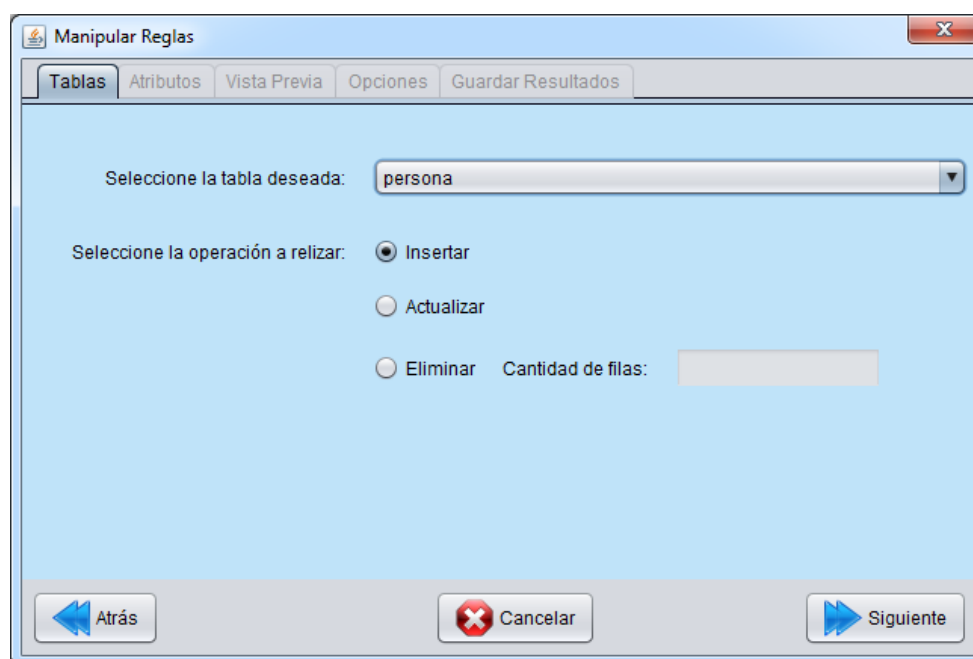


Figura 20. Seleccionar la tabla y la operación de Inserción

La opción *Insertar* se selecciona por defecto. Se elige una tabla deseada para insertar datos. Después de seleccionar la opción “Siguiente” se muestra una ficha *Atributos* como la **Figura 21** indica.

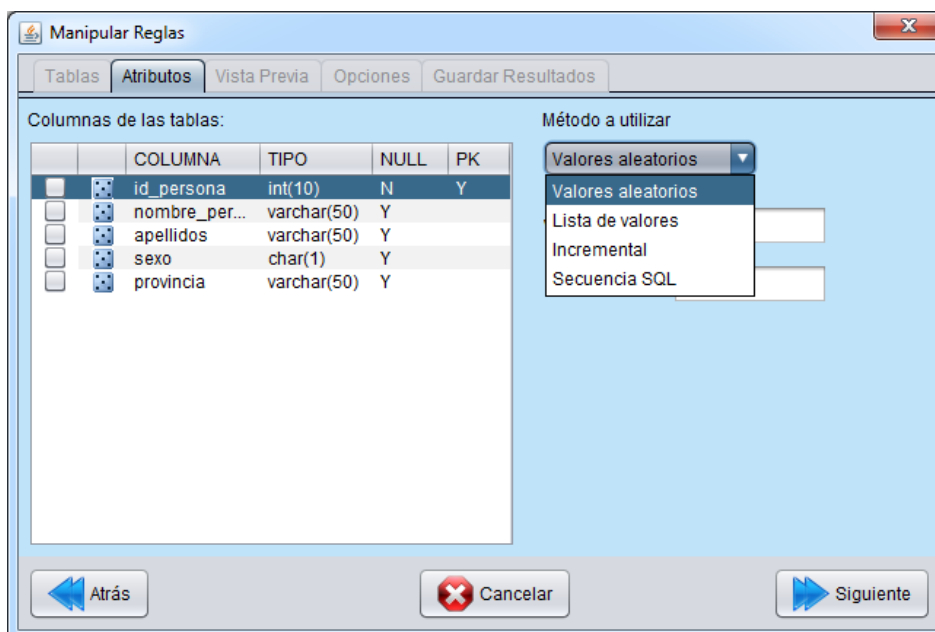


Figura 21. Atributos y Métodos

En esta ventana, se puede observar que hay dos zonas principales: *Columnas de las tablas* y *Método a utilizar*.

En la primera zona, se puede observar claramente una lista de columnas de la tabla seleccionada que contienen el nombre de columna, el tipo de atributo con su longitud, y específica si el atributo es NULL o no y cuál es la llave primaria de la tabla. Al lado de los nombres de columnas aparecen los botones a marcar y los íconos que se cambian en correspondencia con el método a trabajar en ese atributo.

En la segunda zona, se puede observar una lista de métodos: Valores aleatorios, Lista de valores, Incremental y Secuencia SQL.

- **Valores aleatorios:** Se permite seleccionar los valores aleatoriamente. (**Figura 22**)

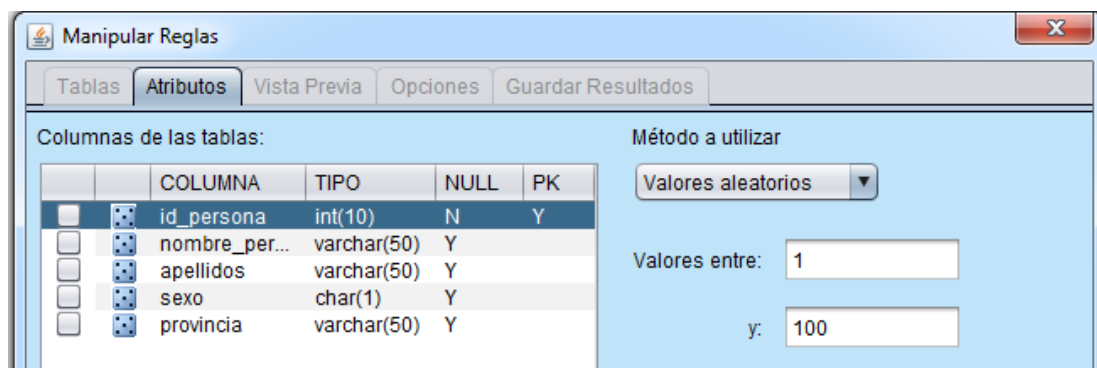


Figura 22. Valores aleatorios

- **Lista de valores:** Es una lista de valores agregada por el usuario. (**Figura 23**)
En lugar de “Valor” permite introducir datos necesarios.
“Agregar” permite obtener datos registrados que se han introducidos en “Valor”.
“Eliminar” permite eliminar un valor en la lista de datos.
“Limpiar” permite vaciar la lista de valores de entrada.

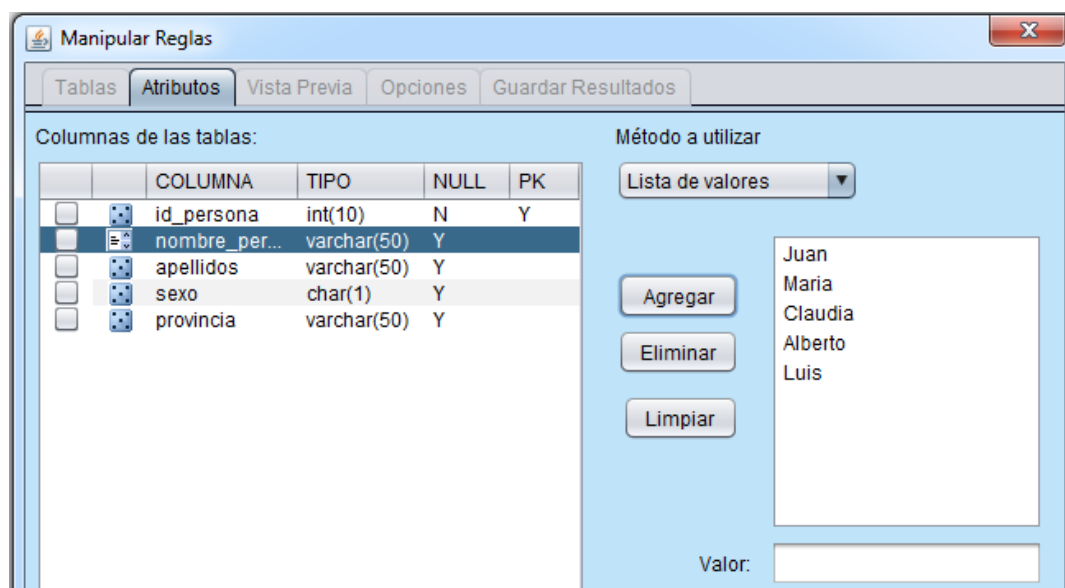


Figura 23. Lista de valores

- **Incremental:** Se generan valores incrementalmente según una distancia. (**Figura 24**)

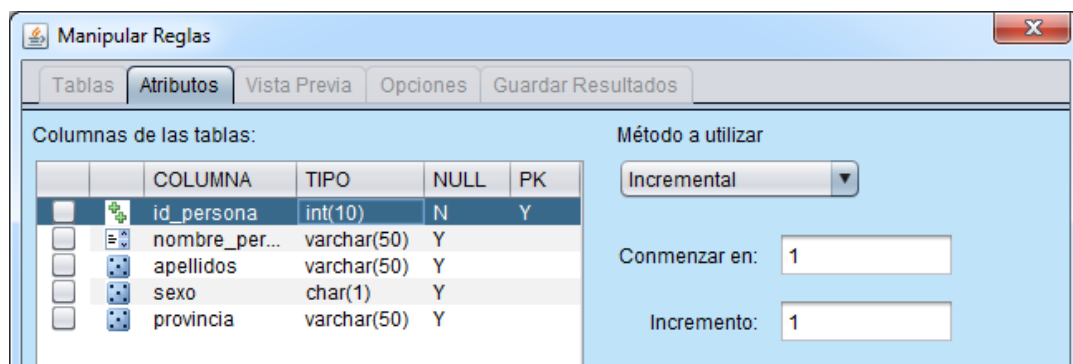


Figura 24. Incremental

- **Secuencia SQL:** Se obtiene una lista de valores a partir de una sentencia SQL local o con otro origen de datos.

La opción “Usar Conexión actual” permite trabajar por defecto en la misma base de datos conectada. Si se desmarca esta opción es porque el usuario deseará conectarse con otro origen de datos.

Se tuvo en cuenta que *Secuencia SQL* de la herramienta solo obtener un valor por atributo, no varios. Por eso, este valor se repite muchas veces en correspondencia con la cantidad de operaciones que introduce el usuario. Entonces si el objetivo es introducir una lista de valores sería mejor utilizar el método “Lista de valores”. La función de *Secuencia SQL* se muestra en la **Figura 25**.



Figura 25. Secuencia SQL

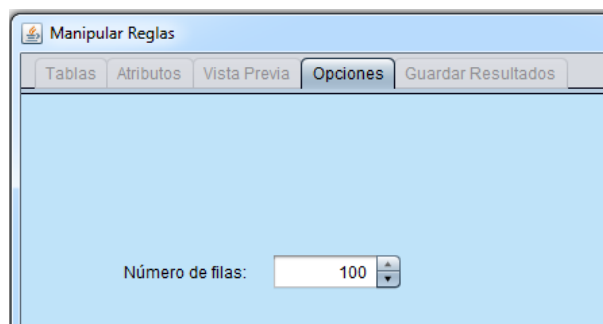
Cuando se introducen todos los datos necesarios usando los métodos adecuados, se da clic en el botón “Siguiente” para revisar el resultado de la ventana *Vista Previa* que se muestra en la **Figura 26**.



id_persona	nombre_persona	apellidos	sexo	provincia
1	Juan	Carlos	m	Santa Clara
2	Maria	Elena	f	Santa Clara
3	Claudia	Rodriguez	f	Santa Clara
4	Alberto	Cesar	m	Santa Clara
5	Luis	Daniel	m	Santa Clara

Figura 26. Vista Previa

Cuando el resultado es el deseado como la *Vista Previa* presenta, se pasa al próximo paso del asistente mediante la ficha *Opciones* como la **Figura 27** muestra. Aquí se puede introducir una gran cantidad de tuplas a insertar en la base de datos.



Número de filas:

Figura 27. Opciones

Después de introducir la cantidad de filas deseadas a insertar, se da clic en el botón “Siguiente”, se muestra la ficha *Guardar resultados* como se puede apreciar en la **Figura 28**. En esta ficha, hay tres opciones a salvar resultado:

- “Ambos” es para guardar en la base de datos y en ficheros.
- “Base Datos” es para guardar en la base de datos en la que se está trabajando.
- “Ficheros” es para guardar en uno o varios tipos de ficheros de texto (Texto, SQL, XML, JSON).

Luego se da clic en el botón “Finalizar” para obtener una regla creada.

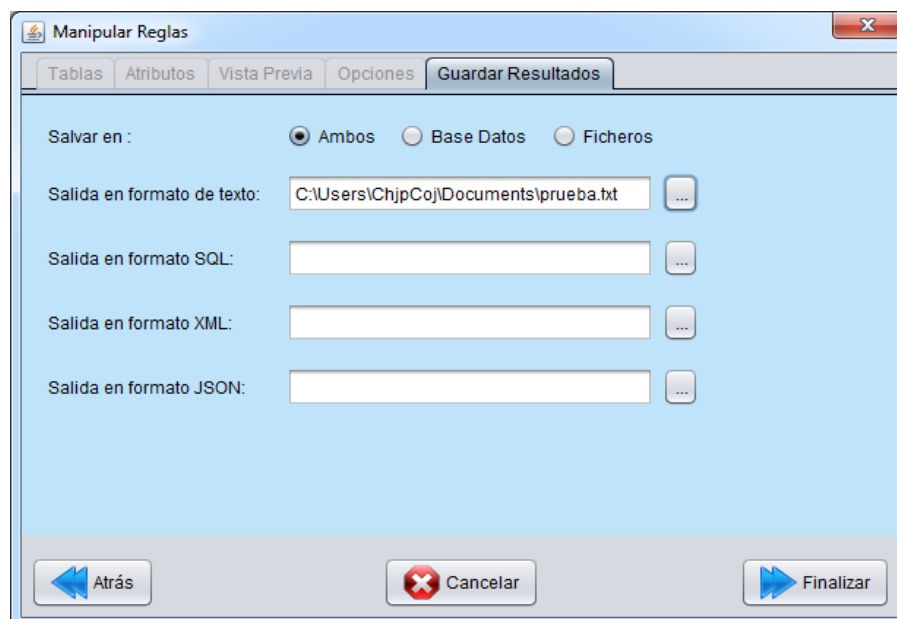



Figura 28. Guardar resultados

Se ejecuta la regla creada y se obtiene el resultado deseado.

3.3.3 Crear regla de Actualización

Para realizar este objetivo, se presiona el botón  “Añadir regla” o también se puede seleccionar dentro del menú “Reglas” la opción “Añadir regla”. Inmediatamente se muestran en una ventana con la ficha *Tablas* como la primera opción dentro de un asistente para crear una regla que contiene, además, las fichas: *Atributos*, *Vista Previa*, *Opciones* y *Guardar Resultados*. Esta opción inicial permite seleccionar la tabla deseada para insertar los datos. Se selecciona la tabla necesaria a actualizar y se elige *Actualizar*, como la **Figura 29** muestra, con el fin de culminar el objetivo deseado.

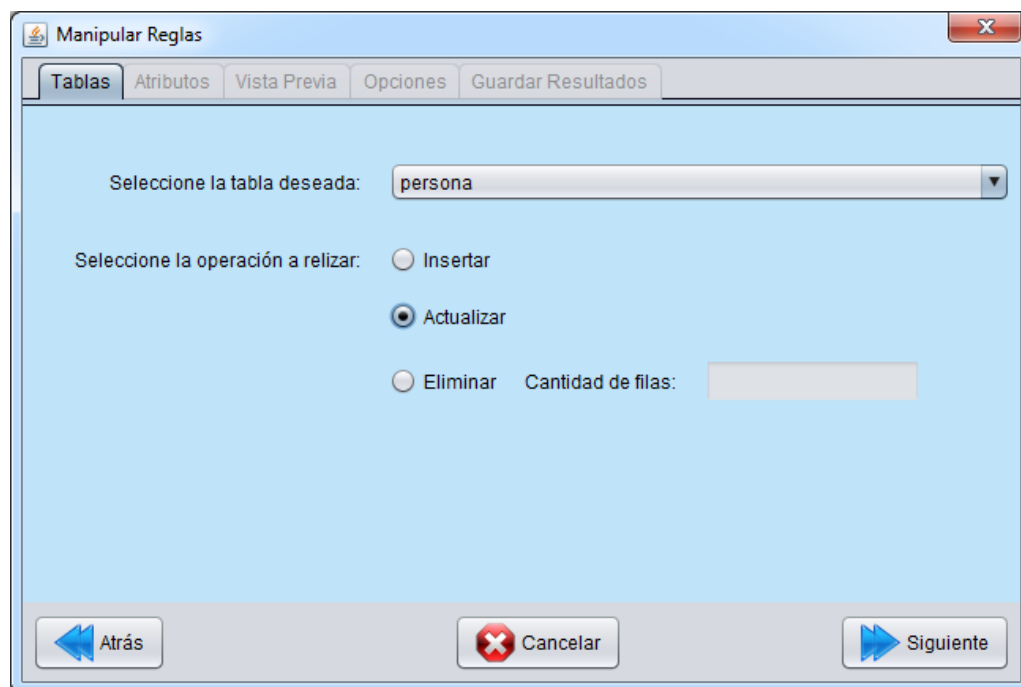


Figura 29. Seleccionar la tabla y la operación de Actualización

Cuando se da clic en el botón “Siguiente” se muestra una ventana que permite seleccionar campos para editar datos. Se pueden actualizar todos los campos de la tabla, pero en este ejemplo se editan los datos de la columna “*provincia*”, que se muestra en la **Figura 30**.

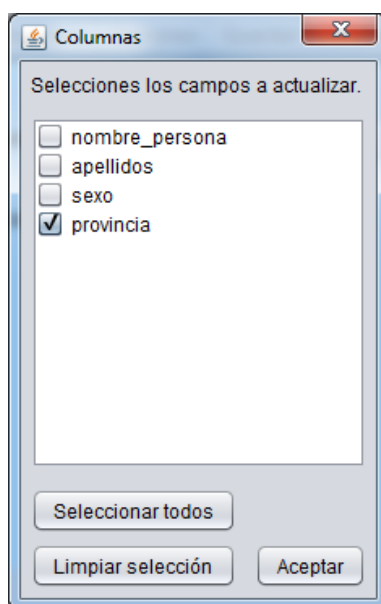


Figura 30. Seleccionar campos a actualizar


Cuando se selecciona la opción “Aceptar”, se muestra la ficha *Atributos* que solo presenta la columna seleccionada a actualizar. Se elige el método a utilizar y se modifican los datos. Se selecciona el método correspondiente al tipo de atributo. Este paso se muestra en la **Figura 31**.



Figura 31. Modificar datos

Luego se revisan los datos actualizados en la ficha *Vista Previa*, se introducen la cantidad de filas deseadas a actualizar y después se guarda el resultado. Cuando se ejecuta la regla, se obtiene el resultado deseado. Se tuvo en cuenta que los datos se generan de forma aleatoria en correspondencia con el número de filas necesarias a editar.

3.3.4 Crear regla de Eliminación

Para crear una regla de *Eliminación*, se da clic en el botón  “Añadir regla” o también se puede seleccionar dentro del menú “Reglas” la opción “Añadir regla”. Luego se muestra una ventana *Tablas* que contiene una lista de tablas, y se selecciona la tabla deseada a eliminar.

Una vez seleccionada la tabla, se elige la operación deseada que es “Eliminar” y se mostrarán la filas a eliminar como se muestra en la **Figura 32**. Luego se presiona el botón “Finalizar” para guardar la regla creada. Se tuvo en cuenta que en esta opción no se pueden eliminar todas las filas de la tabla.

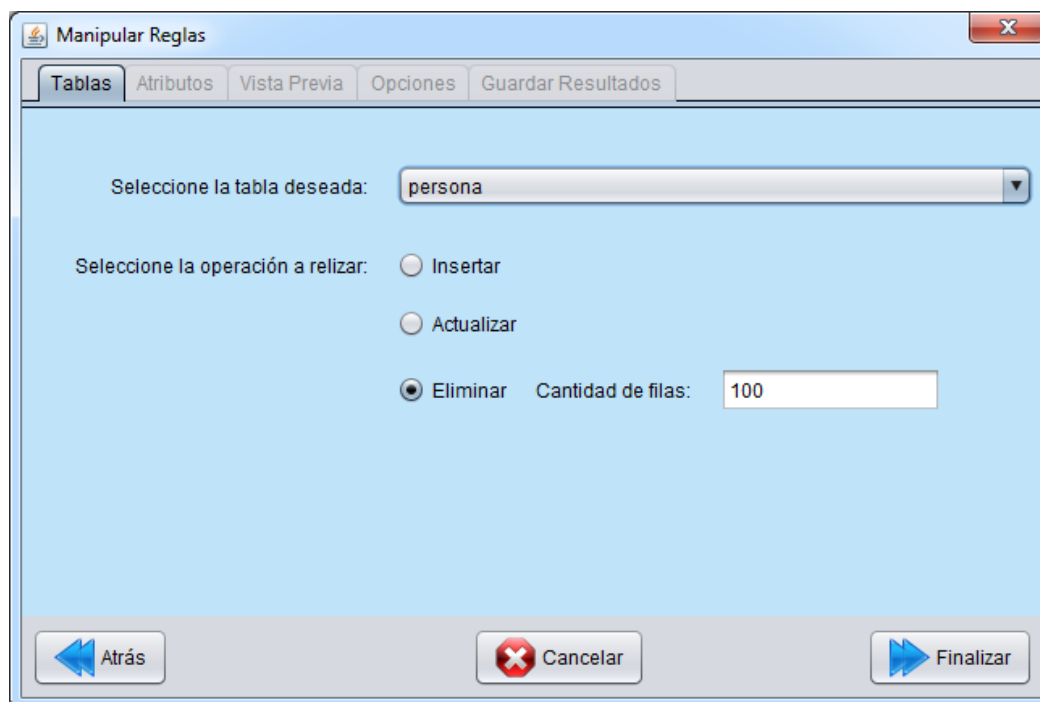



Figura 32. Crear regla de Eliminación

3.4 Modificar una regla

La herramienta es capaz de modificar una regla creada previamente por el usuario. Se tuvo en cuenta que solo se modifican las reglas de generar datos, pero no las reglas de vaciar tablas. Todos los datos de la tabla se pueden editar excepto la llave primaria.

Para realizar este objetivo, se indica la regla necesaria a editar, luego se da clic en el botón  “Modificar regla” o también se puede elegir dentro del menú “Reglas” la opción “Modificar regla”. En este punto se muestra una ventana *Tablas* como en el primer paso de crear regla para generar datos. Desde allí se pueden revisar y modificar los datos deseados.

3.5 Conectar a otro origen de datos usando método Secuencia SQL

En el presente trabajo, la herramienta de generar datos no sólo permite trabajar con grandes volúmenes de datos de la base de datos conectada actualmente, sino que admite además, la conexión a otro origen de datos para tomar los datos desde ese origen e insertarlos o actualizarlos a la base de datos en la que se está trabajando, usando el método **Secuencia SQL**. Se tuvo en cuenta que solo se permite extraer los datos que no son llave primaria de la tabla.

Para comprender con más detalle, se presenta un ejemplo de cómo realizar este trabajo. Por ejemplo, con la base de datos “Bodega”, se necesita insertar datos desde *PostgreSQL* (o *MySQL*) al gestor *Microsoft SQL Server*. Para ello, se deben realizar los siguientes pasos:

Se conecta con el gestor *Microsoft SQL Server*. La tabla “bodega” de la base de datos “Bodega” que en este gestor es vacía. La tabla “bodega” en la misma base de datos “Bodega” en *PostgreSQL* tiene datos iniciales como la **Figura 33** muestra.

	id_bodega [PK] integer	nombre_bodega character(15)
1	1	Primavera
2	2	Braga
3	3	Amanecer
4	4	Siboney
5	5	Tu esperanza
*		

Figura 33. Inicio de la tabla “bodega” en PostgreSQL

Crear una regla de **Inserción** en la tabla “bodega” seleccionada como se explicó anteriormente sobre “Crear regla de Inserción”.

En la columna “nombre_bodega” se selecciona el método a utilizar “Secuencia SQL”, se desmarca la opción “Usar Conexión actual” para mostrar otra ventana que permite establecer

conexión con otro origen de datos. (**Figura 34**). Los pasos para establecer la conexión fueron explicados anteriormente.

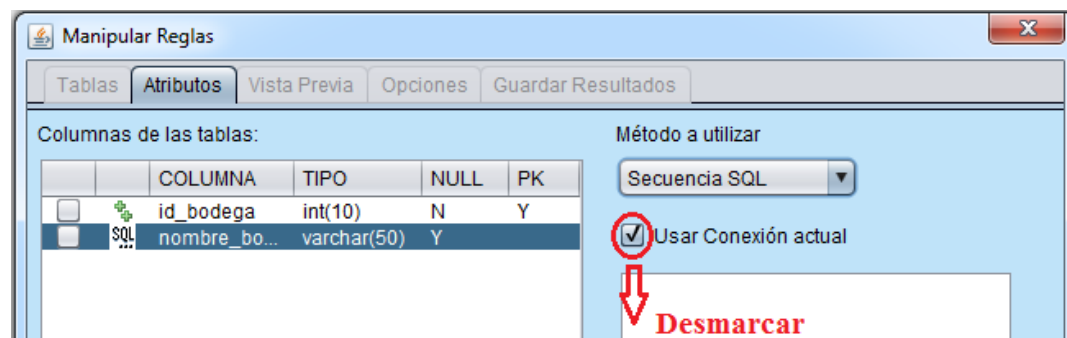


Figura 34. Desmarcar “Usar Conexión actual”

Se escribe la sentencia SQL para tomar datos desde *PostgreSQL* y luego insertarlos a *Microsoft SQL Server* como se ve en la **Figura 35** y se obtiene el resultado en *Microsoft SQL Server* cuando se ejecuta la regla.

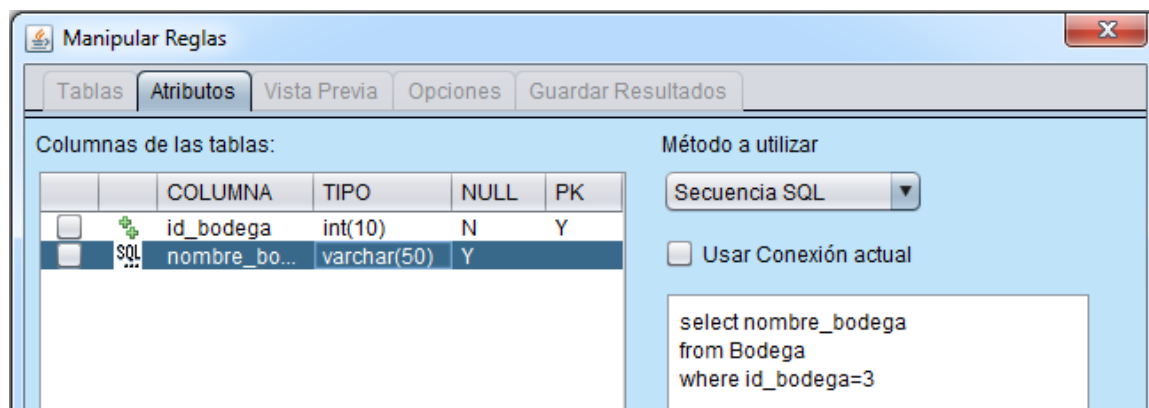


Figura 35. Tomar datos de otro origen de datos

3.6 Resultados de experimentos con la herramienta

En el presente trabajo se realizó un pequeño experimento para examinar el desempeño de la herramienta presentada en cada tipo de operación a realizar en correspondencia con la cantidad de tuplas de la base de datos.

A continuación se presentan los experimentos realizados sobre una conexión en *MySQL*:

- **Experimento para 5000 tuplas:** Este experimento se realiza sobre una base de datos con una cantidad fija de 5000 tuplas. A esta base de datos inicial se le realizaron varias cantidades de operaciones desde 100 hasta 1000. Los tipos de operaciones fueron de *Inserción*, *Actualización* y *Eliminación*. En tales operaciones se midió el tiempo de ejecución en segundos. Se puede ver claramente el resultado obtenido en la **Figura 36**. El tiempo aumenta a medida que aumenta la cantidad de operaciones que se realizan. Las reglas de *Inserción* siempre se generan un poco más rápido que las demás operaciones mientras que las de *Actualización* generalmente son las que más tiempo toman.

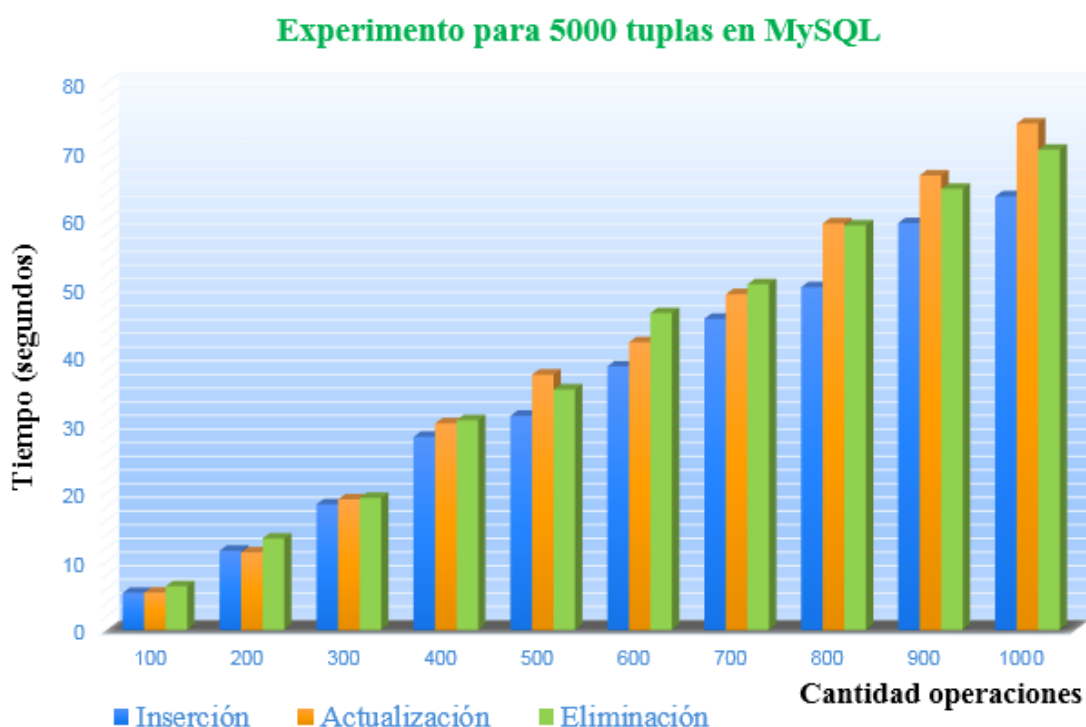


Figura 36. Experimento para 5000 tuplas en MySQL

- **Experimento con 300 operaciones:** En este experimento se comienza con 500 tuplas en una base de datos. Cada vez se realiza el experimento de cada tipo de operación *Inserción*, *Actualización* y *Eliminación* con 300 operaciones, y se aumenta a 500 tuplas en la base de datos para investigar las diferencias entre ellas. el resultado

obtenido se presenta en la **Figura 37**. Las reglas de *Inserción* tomaron menos tiempo de ejecución, mientras que las de *Eliminación* demoraron más.



Figura 37. Experimento con 300 operaciones

- **Experimento de insertar datos en la base de datos y en formatos de texto:** se puede ver claramente que los datos insertados en los formatos de texto se ejecutan mucho más rápido que con los que se ejecutan en la base de datos. El tiempo aumenta en correspondencia con la cantidad de operaciones necesarias a insertar. (**Figura 38**)

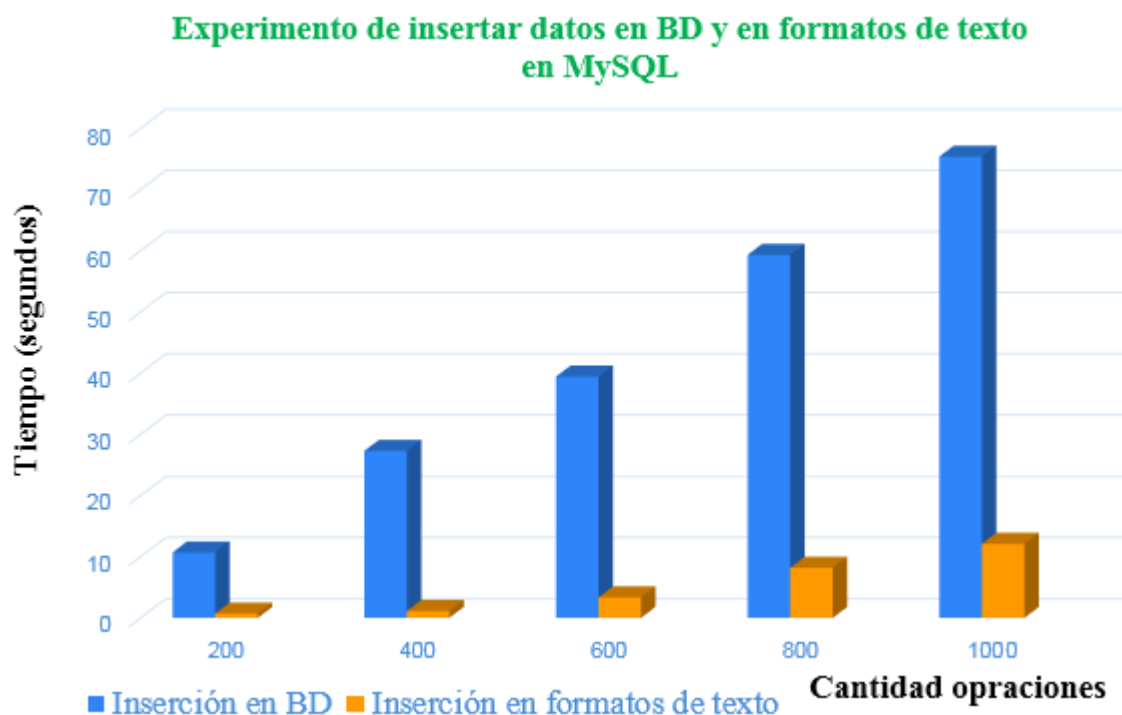


Figura 38. Experimento de insertar datos en BD y en formatos de texto

3.7 Conclusiones parciales del capítulo

En el presente capítulo se presentó un caso de estudio sobre la factibilidad del uso del sistema para los desarrolladores. Se ofrecen detalles sobre las características y uso de la herramienta. En este se pudo confirmar las ventajas que brinda la herramienta para manipular datos y obtener información necesarias y satisfechas de los usuarios.

Conclusiones

Con el resultado de esta investigación se implementó DaGenSQL, una nueva herramienta de generación de datos para el desarrollo de Sistemas de Información y el análisis de grandes volúmenes de datos, cumpliéndose el objetivo general planteado. Específicamente, se cumplió con los siguientes aspectos:

1. Se extrajeron los metadatos de diferentes gestores de base de datos desde su catálogo, específicamente SQL Server, MySQL y PostgreSQL.
2. Se establecieron y desarrollaron cuatro formas para generar valores de un atributo.
3. Se crearon diferentes reglas para poblar las tablas del sistema y se permitió ordenarlas para su ejecución.
4. Se diseñó e implementó DaGenSQL una herramienta capaz de generar datos automáticamente de interfaz sencilla y amigable.

Recomendaciones

Como parte de posibles aplicaciones del sistema de generar datos en Sistemas de Información se recomienda:

- Obtener una nueva versión más rápida para generar datos en el desarrollo de Sistemas de Información.
- Que el sistema sea capaz de realizar la conexión con tres tipos de gestor de la Bases de Datos SQL específicos (Microsoft SQL Server, MySQL y PostgreSQL), además se toman los datos desde otros orígenes de datos, para poder manipular las informaciones necesarias en las empresas como las universitarias.

Referencias Bibliográficas

- ADRIAN MARTINEZ, O. 2015. *Tipos de gestores de bases de datos* [Online]. . Available: <http://gestoresadrian.blogspot.com/> [Accessed 12-06-2015 2015].
- ALAIN, P. T. 2014. *La integridad en el modelado ER de los Sistemas de Información*. Tesis Maestría, Universidad Central "Marta Abreu" de las Villas.
- ANGIE MERCEDES, C. G. 2014. *Sistemas de Información y Toma de Decisiones* [Online]. Available: <http://angiemercedesc.blogspot.com/> [Accessed 12-6-2015].
- BERGMANN, V. 2015. *Databene Benerator* [Online]. Available: <http://databene.org/databene-benerator> [Accessed 12-6-2015].
- BERNARDO. 2015. *Catálogo de un sistema de base de datos* [Online]. Available: <http://programmingbernie.blogspot.com/2009/11/catalogo-de-un-sistema-de-base-de-datos.html> [Accessed 12-6-2015 2015].
- BERZAL, F. 2015. *Ciclo de vida de los sistemas de información*. Available: <http://elvex.ugr.es/idbis/db/docs/lifecycle.pdf>.
- BLOG.CLASSORA. 2015. *Metadatos: definición, aplicaciones y estándares* [Online]. Available: <http://blog.classora.com/2013/02/28/metadatos-definicion-aplicaciones-y-estandares/> [Accessed 12-6-2015].
- CANTALAPIEDRA LLERENA, M. L. A. 2006. *Diseño de un Sistema de Información de Marketing para el Centro de Información Comercial Científico y Tecnológica del Grupo Empresarial CUBALSE*. Ponencia presentada en Congreso Internacional de Información, INFO 2006.Taller Internacional sobre Inteligencia Empresarial y Gestión del Conocimiento, Ciudad de la Habana.
- CAÑAVATE, A. M. 2003. *Sistemas de información en las empresas*. *Hipertext. net*, 1, 251.
- CÉSAR JULIO, B. M. 2015. *Bases de Datos usando Java : JDBC* [Online]. Available: <http://www.slideshare.net/eimher/jdbc-14307886> [Accessed 12-6-2015].
- COBO, Á. 2005. *PHP y MySQL: Tecnología para el desarrollo de aplicaciones web*, Editorial Díaz de Santos, S.A.
- CURSOS.ITAM.MX. 2015. *El catálogo del sistema*. Available: <http://cursos.itam.mx/akuri/2002/S12002/BasesDeDatos/CHAPTER6.PDF> [Accessed 12-6-2015].
- DATE, C. J. 2001. *Introducción a los sistemas de bases de datos*, Pearson Educación.

Referencias Bibliográficas

- DE BLANCO, A. 1994. *Sistema de información gerencial para la toma de decisiones*. Tesis de Licenciatura. UPIICSA. México DF.
- ECURED.CU. 2015a. *Integridad de las Bases de Datos* [Online]. Available: [http://www.ecured.cu/index.php/Integridad de las Bases de Datos](http://www.ecured.cu/index.php/Integridad_de las Bases de Datos) [Accessed 12-6-2015].
- ECURED.CU. 2015b. *Sistema Gestor de Base de datos* [Online]. Available: [http://www.ecured.cu/index.php/Sistema Gestor de Base de Datos](http://www.ecured.cu/index.php/Sistema_Gestor_de Base de Datos) [Accessed 12-6-2015].
- GENERATEDATA. 2015. *Generatedata* [Online]. Available: <http://www.generatedata.com/> [Accessed 12-6-2015].
- ITERAMOS. 2015. *Herramientas para la generación de datos Mock?* [Online]. Available: <http://www.iteramos.com/pregunta/5257/herramientas-para-la-generacion-de-datos-mock> [Accessed 12-6-2015].
- LAPUENTE, M. J. L. 2012. Hipertexto: El nuevo concepto de documento en la cultura de la imagen. http://www.hipertexto.info/documentos/web_tecnolog.htm. Consultado em, 19, 06-07.
- LÁZARO JESÚS, B. E. 2012. *Sistemas de información para el economista y el contador*, Editorial Académica Española (July 24, 2012).
- LEANDRO, A. 2015. ¿Cuál es la definición de integridad de datos (informática)? [Online]. Available: <http://www.alegsa.com.ar/Dic/integridad%20de%20datos.php> [Accessed 12-6-2015].
- LOCKHART, T. 2001. Tutorial de PostgreSQL. *The PostgreSQL Administrator's Guide*, 1-2.
- PERSONAL.LOBOCOM. 2015. *Introducción* [Online]. Available: <http://personal.lobocom.es/claudio/sql001.htm> [Accessed 12-6-2015].
- PETKOVIC, D. 2005. *Microsoft SQL Server 2005: A Beginner's Guide*, McGraw-Hill, Inc.
- PRESSMAN, R. S., MARTÍN, R. O. & AGUILAR, L. J. 2003. *Ingeniería del software: un enfoque práctico*, McGraw-Hill.
- PROFESORES.ELO.UTFSM.CL. 2015. *JDBC API* [Online]. Available: <http://profesores.elo.utfsm.cl/~agv/elo330/2s09/lectures/JDBC/JDBC.html> [Accessed 12-6-2015].
- RED GATE, S. L. 2015. *Red Gate SQL Data Generator* [Online]. Available: <http://www.sigmasd.com/product/Red-Gate-Software-Ltd/Red-Gate-SQL-Data-Generator/396583> [Accessed 12-6-2015].

Referencias Bibliográficas

- REDGATE. 2015. *Redgate SQL Data Generator* [Online]. Available: <http://www.red-gate.com/products/sql-development/sql-data-generator/> [Accessed 12-6-2015].
- RODRÍGUEZ, J. R. 2005. *Gestión de proyectos informáticos: métodos, herramientas y casos*, UOC.
- SAROKA HERNÁNDEZ, R. 2002. *Sistemas de Información en la era digital*. [sl]: Fundación OSDE.
- SQLEDIT. 2015. *DTM Data Generator* [Online]. Available: <http://www.sqledit.com/dg/> [Accessed 12-6-2015].
- STAIR, R. & REYNOLDS, G. 2011. *Principles of information systems*, Cengage Learning.
- TECHNET.MICROSOFT. 2015. *Restricciones* [Online]. Available: <https://technet.microsoft.com/es-es/library/ms189862%28v=sql.105%29.aspx> [Accessed 12-6-2015].
- WIKIPEDIA. 2015a. *Metadato* [Online]. Available: <http://es.wikipedia.org/wiki/Metadato> [Accessed 12-6-2015].
- WIKIPEDIA. 2015b. *Sistema de gestión de bases de datos* [Online]. Available: http://es.wikipedia.org/wiki/Sistema_de_gestión_de_bases_de_datos [Accessed 12-6-2015].
- WIKIPEDIA. 2015c. *SQL* [Online]. Available: <http://es.wikipedia.org/wiki/SQL> [Accessed 12-6-2015].