

**HERRAMIENTA SOFTWARE PARA LA GENERACIÓN DE DATOS DE
PRUEBAS PARA APLICACIONES A PARTIR DE UNA BASE DE DATOS
RELACIONAL.**

FANNY MARCELA MONTAÑEZ GÉLVEZ

**UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERÍAS FISICOMECAÑICAS
ESCUELA INGENIERÍA DE SISTEMAS E INFORMÁTICA
BUCARAMANGA**

2015

**HERRAMIENTA SOFTWARE PARA LA GENERACIÓN DE DATOS DE
PRUEBAS PARA APLICACIONES A PARTIR DE UNA BASE DE DATOS
RELACIONAL.**

FANNY MARCELA MONTAÑEZ GÉLVEZ

**TESIS DE GRADO: TRABAJO DE INVESTIGACIÓN PARA OPTAR AL TITULO
DE INGENIERO DE SISTEMAS**

**DIRECTOR: CARLOS ADOLFO BELTRÁN
INGENIERO DE SISTEMAS**

**UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERÍAS FISICOMECÁNICAS
ESCUELA INGENIERÍA DE SISTEMAS E INFORMÁTICA
BUCARAMANGA**

2015

AGRADECIMIENTOS

Este trabajo no sería posible sin la influencia directa de Dios, él me brindó la oportunidad de realizar el actual proyecto, ya que cuando sentí caer me levanto, me dio ánimo para seguir y salir adelante, al igual que me presento las personas indicadas para responder dudas e impulsar mi proyecto a su finalización.

Agradezco todo el apoyo de mi profesor, asesor y amigo Carlos Felipe Reyes Contreras sin el cual no habría sido posible llegar a la culminación de mi trabajo de grado de una manera satisfactoria a un nivel académico y personal. Gracias por todo su tiempo, sus consejos, su comprensión y su trabajo.

Gracias a mis amigos, compañeros y a la comunidad universitaria por todas las lecciones de vida y trabajo.

A mi director de proyecto por su tiempo, dedicación y disposición en todo momento.

A mis grandes amigos Dayana, Marcela, Ingrid, Alba, Carlos, Eduardo, Johare, Ferney Pico, Katherine, Luna, Jeison, Didiana, en fin a todos, por su gran apoyo incondicional, sus maravillosas palabras cuando las necesite en esta bonita etapa de mi vida, su tiempo y las miles de sonrisas que siempre despertaron en mí, fue el impulso para llegar hasta el fin.

Por último a mis padres y mi familia, por su apoyo, su confianza, su amor, su gran entrega, su trabajo constante por cultivar todo lo bueno que hay en mí. Por ser personas admirables llenas de entusiasmo, muy buena energía, honestas, en fin personas exitosas.

Ahora solo queda cumplir mis sueños, ya que gracias a todos ustedes tengo las bases necesarias para un día muy cercano decir lo que hoy con mucho orgullo declaro “estoy logrado lo que más quiero”.

TABLA DE CONTENIDO

<u>INTRODUCCIÓN</u>	<u>16</u>
<u>1. DESCRIPCIÓN DEL PROYECTO</u>	<u>18</u>
1.1 DESCRIPCIÓN DE LA PROBLEMÁTICA.....	18
1.2 OBJETIVOS	20
1.2.1 OBJETIVO GENERAL	20
1.2.2 OBJETIVOS ESPECÍFICOS	20
<u>2. MARCO TEÓRICO</u>	<u>21</u>
2.1 FUNDAMENTOS TEÓRICOS	21
2.1.1 MODELADO DE BASE DE DATOS.	21
2.1.2 MODELADO RELACIONAL.	21
2.1.3 CLAVES PRIMARIAS.	22
2.1.4 CLAVES FORÁNEAS.....	23
2.1.5 DOMINIOS.....	24
2.1.6 REGLAS DE INTEGRIDAD.....	24
2.1.7 NULOS.....	25
2.1.8 REGLA DE INTEGRIDAD DE ENTIDADES.....	25
2.1.9 REGLA DE INTEGRIDAD REFERENCIAL.	25
2.2 PROTOCOLO DE CONECTIVIDAD DE BASE DE DATOS.....	26
2.2.1 CONECTIVIDAD ABIERTA DE BASES DE DATOS (ODBC)..	26
<u>3. DESARROLLO E IMPLEMENTACIÓN DE LA METODOLOGÍA DEL PROYECTO</u>	<u>29</u>
3.1 METODOLOGÍA SCRUM	29
3.1.1 CARACTERIZACIÓN DEL PROCESO SOFTWARE.	31
3.1.2 FASES Y ACTIVIDADES.....	33
3.1.3 DISEÑO E IMPLEMENTACIÓN.....	34
3.1.3.1 DESCRIPCIÓN DE LOS MÓDULOS DEL SISTEMA.....	35
3.1.3.1.1 MÓDULO DE CONCEPCIÓN DEL PROYECTO.	37
3.1.3.1.2 MÓDULO DE GESTIÓN DE PATRONES	39
3.1.3.1.3 MÓDULO DE GESTIÓN DE LA BASE DE DATOS.....	40

3.1.3.1.4	MÓDULO DE GESTIÓN DE PROYECTOS	41
3.1.3.1.5	GENERACIÓN DE LOS SCRIPTS PARA LAS SENTENCIAS SQL PARA LA ALIMENTACIÓN DE LA BASE DE DATOS	41
3.1.4	ENTREGABLES DEL PROYECTO.	42
3.1.4.1	SPRINT 1. CONCEPCIÓN DEL PROYECTO.	42
3.1.4.2	SPRINT 2. MÓDULO DE PATRONES.	43
3.1.4.3	SPRINT 3. MÓDULO DE BASE DE DATOS.	43
3.1.4.4	SPRINT 4. MÓDULO DE PROYECTOS.	44
3.1.4.5	SPRINT 5. MÓDULO DE GENERACIÓN DE LOS SCRIPTS PARA LAS SENTENCIAS SQL PARA LA ALIMENTACIÓN DE LA BASE DE DATOS.	44
3.1.4.6	PRODUCT BACKLOG.	45
4.	EL PRODUCTO Y SU DOCUMENTACIÓN.....	47
4.1	CARACTERÍSTICAS DEL PRODUCTO DURANTE SU DESARROLLO.....	47
4.1.1	CONSTRUIR UN PATRÓN.	48
4.1.1.1	APLICAR DETALLES A LOS PATRONES.	49
4.1.2	IDENTIFICAR EL GESTOR DE BASE DE DATOS.	49
4.1.3	REALIZAR CONEXIÓN A UN PROYECTO.	50
4.1.4	GENERACIÓN DE SCRIPTS PARA LAS SENTENCIAS SQL PARA LA ALIMENTACIÓN DE DATOS CON DATOS PRUEBA.	50
4.1.4.1	FASE DE PREPARACIÓN	50
4.1.4.2	FASE DE GENERACIÓN	50
4.2	RESULTADOS OBTENIDOS	51
4.2.1	ESTRUCTURA DE LA BASE DE DATOS.	51
4.2.1.1	REGISTRO DE PATRONES DE DATOS.	52
4.2.1.2	DETALLE DE VALORES PARA LOS PATRONES	53
4.2.1.3	BASE DE DATOS DISPONIBLES	54
4.2.1.4	REGISTRO DE PROYECTOS	55
4.2.1.5	DETALLE DE ATRIBUTOS POR TABLA DE PROYECTO	56
4.2.1.6	CONJUNTO DE TABLAS DE UNA BASE DE DATOS DE PROYECTO	57
4.2.2	ESTRUCTURA FUNCIONAL DEL PROYECTO.	58
4.2.2.1	DISEÑO DE VENTANAS.	58
4.2.2.1.1	VENTANA PRINCIPAL.....	58
4.2.2.1.2	VENTANA DE DATOS	60
4.3	ESTRUCTURA FUNCIONAL Y RECORRIDO DEL SISTEMA.....	62
4.3.1	INGRESO AL SISTEMA.....	62
4.3.2	MÓDULO DE PATRONES.....	63
4.3.3	MÓDULO DE BASE DE DATOS	65
4.3.4	MÓDULO DE PROYECTOS.....	66

4.3.5	MÓDULO DE GENERACIÓN DE LOS SCRIPTS PARA LAS SENTENCIAS SQL PARA LA ALIMENTACIÓN DE LA BASE DE DATOS.....	67
4.3.6	AYUDA DE LA APLICACIÓN.....	70
4.3.7	SALIR DE LA APLICACIÓN.....	71
<u>5. EL LENGUAJE SQL: SENTENCIA PARA LA ADICIÓN DE DATOS A LA BASE DE DATOS</u>		<u>72</u>
5.1	ADICIÓN DE DATOS A LA BASE DE DATOS.....	73
5.1.1	LA SENTENCIA INSERT DE UNA FILA.....	74
5.1.2	LA SENTENCIA INSERT MULTIFILA.....	75
5.1.3	UTILIDADES DE CARGA MASIVA.....	76
<u>6. CONCLUSIONES</u>		<u>77</u>
<u>7. RECOMENDACIONES.....</u>		<u>79</u>
<u>BIBLIOGRAFIA.....</u>		<u>81</u>

LISTA DE TABLAS

	Página
Tabla 1. Descripción Módulos del Sistema	36
Tabla 2. Listado de Producto	45
Tabla 3. Dominios de los atributos de la relación que almacena los datos de la entidad PERSONA.....	48
Tabla 4. Ejemplos de patrones y valores que se pueden crear en un atributo.	49
Tabla 5. Descripción Tabla: patron	52
Tabla 6. Descripción Tabla: dtllepatron.....	53
Tabla 7. Descripción Tabla: base.....	54
Tabla 8. Descripción Tabla: proyecto.....	55
Tabla 9. Descripción Tabla: dtlletblaproy	56
Tabla 10. Descripción Tabla: tablasproy.....	57

LISTA DE FIGURAS

	Página
Figura 1. Clave Foránea	23
Figura 2. Conectividad con ODBC	27
Figura 3. Modelo ciclo de vida del desarrollo aplicando SCRUM.....	30
Figura 4. Metodología SCRUM adaptada al proyecto de desarrollo.	32
Figura 5. Módulos del Sistema.....	36
Figura 6. Línea de Tiempo del Proyecto	42
Figura 7. Diagrama Relacional.....	51
Figura 8. Script para la creación de la Tabla: patrón.....	52
Figura 9. Script para la creación de la Tabla: dtlletpatron	53
Figura 10. Script para la creación de la Tabla: base	54
Figura 11. Script para la creación de la Tabla: proyecto	55
Figura 12. Script para la creación de la Tabla: dtlletblaproy	56
Figura 13. Script para la creación de la Tabla: tablasproy	57
Figura 14. Ventana Principal	58
Figura 15. Ventana de Datos	60
Figura 16. Íconos de Navegación.....	61
Figura 17. Íconos de gestión de registros.....	61
Figura 18. Ícono de Ingreso al Sistema.....	62
Figura 19. Ventana Principal de la Aplicación	62
Figura 20. Ingreso Módulo Patrones.....	63
Figura 21. Ventana Gestión de Patrones.....	64
Figura 22. Ingreso Módulo Base de Datos.....	65
Figura 23. Ventana de Gestión de Base de Datos	65
Figura 24. Ingreso Módulo de Proyectos.....	66
Figura 25. Ventana de Gestión de Proyectos.....	67
Figura 26. Ingreso Módulo Generador.....	68
Figura 27. Ventana de Generador de datos	68
Figura 28. Generación de Sentencias	70
Figura 29. Ícono de Ayuda.....	70
Figura 30. Botón Salir de la Aplicación.....	71
Figura 31. Diagrama sintáctico de la sentencia INSERT de una fila.	74
Figura 32. Diagrama sintáctico de la sentencia INSERT multifila.	75

GLOSARIO

Atributo: Son aquellas características, propiedades o cualidades de la entidad para las cuales es importante registrar su valor o alguna actualización en ellas. Por ejemplo la entidad persona tendrá los atributos nombre, domicilio, población.

Base de Datos: Es una colección o depósito de datos organizados y relacionados entre sí, los cuales son recolectados y explotados por los Sistemas de Información de una empresa o negocio en particular.

Campo: Un campo es la unidad básica de una base de datos. El campo puede ser, por ejemplo, el nombre de una persona. Un campo debe cumplir con ciertos requisitos: no puede empezar con espacios en blanco y caracteres especiales, no puede llevar puntos, ni signos de exclamación o corchetes.

Clave Primaria: La clave primaria es la clave candidata escogida para identificar un registro de modo único.

Clave Foránea: Atributo o conjunto de atributos de un relación cuyos valores coinciden con los valores de la clave primaria de otra relación. Representan relaciones entre datos.

DML: Lenguaje de Manipulación de Datos. Lenguaje que permite manipular los datos en las base de datos, es decir, realizar consultas, inserciones, eliminaciones, entre otras.

Dominio: Es el conjunto de valores que puede tomar cada atributo.

Entidad: Son objetos concretos, reales o abstractos que presentan interés para el sistema y sobre los que se recoge información que será representada en un sistema de base de datos.

Integridad de datos: Grado hasta el cual los datos en cualquier archivo individual deben ser exactos.

Integridad referencial: Propiedad imprescindible en toda base de datos. Garantiza que un conjunto de datos (registros) siempre se relacionen con otros conjuntos válidos que existan en la base de datos.

Metodología: Conjunto de procedimientos, técnicas y ayudas para la documentación y desarrollo de un producto software.

Modelo de datos relacional: Una colección o depósito de relaciones normalizadas en la que cada relación tiene un nombre distintivo.

Módulo: Estructura que forma parte de un sistema; su operación es independiente de otros módulos, pero su implementación puede requerir de datos o información proporcionada de otros módulos del sistema.

ODBC: La Conectividad Abierta de Base de Datos Abierta (Open DataBase Connectivity, ODBC) es un estándar de acceso a las bases de datos utilizado por los sistemas Microsoft.

Propietario del producto (Product Owner): Persona encargada de la dirección y control del Product Backlog, es decir, las historias de usuario que debe cumplir el sistema.

Registro: Conjunto de información referida a una misma unidad.

Script: Un script es un tipo de programa que consiste de una serie de instrucciones que serán utilizadas por otra aplicación.

Scrum: Término que describe una forma para desarrollar proyectos iniciada en Japón. Metodología iterativa e incremental en los procesos de gestión, mejora y control del producto.

Sentencia INSERT: Esta sentencia permite insertar filas múltiples en una tabla.

Sprint (Iteración): Etapas de corta duración (no más de 30 días) donde se desarrolla un incremento iterativo del producto.

SQL: Lenguaje de consulta estructurado (Structured Query Lenguaje), es un lenguaje declarativo que permite acceder a las bases de datos relacionales y especificar diversos tipos de operaciones en ellas.

RESUMEN

Título: HERRAMIENTA SOFTWARE PARA LA GENERACIÓN DE DATOS DE PRUEBAS PARA APLICACIONES A PARTIR DE UNA BASE DE DATOS RELACIONAL.*

AUTOR: Fanny Marcela Montañez Gélvez**

PALABRAS CLAVE: Integridad Referencial, ODBC, generación de datos prueba, Scrum, base de datos.

DESCRIPCIÓN: Los desarrolladores en el momento de verificar sus aplicaciones o programas software deben realizar pruebas donde detecten los errores potenciales en todas las posibles situaciones por las que atraviesa el sistema; para ello es fundamental el uso de datos, donde dichos datos sean muy similares al producto deseado por el usuario final y aquella base de datos contenga un gran volumen de datos para realizar un mayor número de pruebas con las cuales pueda realizar verificaciones y validaciones y así comprobar el buen funcionamiento del producto y las expectativas del cliente. La presente herramienta software permite mediante la generación de datos prueba ser un instrumento para los desarrolladores en la alimentación de sus bases de datos. Con la funcionalidad que contempla la integridad referencial, los datos generados serán más consistentes para la base de datos. La herramienta permite conectarse a diferentes bases de datos mediante la tecnología ODBC, recurso de Microsoft que conecta la herramienta con la base de datos del desarrollador. Adicionalmente la herramienta puede utilizarse como recurso de apoyo para los docentes y para los estudiantes como mecanismo de aprendizaje. Para el desarrollo de la herramienta software se utilizó la metodología SCRUM por ser una metodología ágil, práctica e incremental. En la documentación se describe el diseño de las interfaces de usuario, los módulos que conforman la herramienta y los resultados obtenidos luego de integrar los módulos. Además se presenta la definición y la historia de SQL, además de la explicación de las sentencias usadas para la creación de la base de datos.

*Trabajo de Grado

**Facultad de Ingeniería Fisicomécanicas. Escuela Ingeniería de Sistemas E Informática.
Director: Carlos Adolfo Beltrán

ABSTRACT

Title: SOFTWARE TOOL FOR THE GENERATION OF TEST DATA FOR APPLICATIONS FROM A RELATIONAL DATABASE. *

AUTHOR: Fanny Marcela Montañez Gélvez **

KEYWORDS: Referential Integrity, ODBC, test data generation, Scrum, database technology ODBC.

DESCRIPTION: The developers at the time of check their software applications or programs should make testing to detect potential errors in all possible situations that passes through the system; for it is essential to use data, where such data are very similar to the desired by the end user and that database contains a large volume of data for a greater number of tests which can perform verification and validation and thus verify the proper operation of the product and customer expectations. This tool allows software by generating test data become a tool for developers in feeding their databases. With the functionality that provides referential integrity, data generated will be more consistent for the database. The tool allows to connect to different databases using ODBC technology, Microsoft resource tool that connects with the database developer. Additionally, the tool can be used as a support resource for teachers and students as a learning mechanism. For the development of the software tool the SCRUM methodology was used to be a flexible, practical and incremental methodology. The documentation described designing user interfaces, modules as part of the tool, and the results obtained after integrating modules. Also presents the definition and history of SQL, in addition, the explanation of the statements used to create the database.

*Work Degree

**Faculty of Engineering Physicomechanical. School E Compute Systems Engineering
Director: Carlos Adolfo Beltrán

INTRODUCCIÓN

La industria del software ha crecido significativamente; en la actualidad existe un gran número de desarrolladores; los cuales continuamente están mejorando sus creaciones, como también incorporando nuevas herramientas, haciendo más fácil la labor del desarrollo del software; minimizando tiempos de producción, buscando calidad, verificando y probando resultados.

La herramienta desarrollada permite a los profesionales desarrolladores de software, a docentes y estudiantes de carreras relacionadas con programación en computadores, a administradores y gerentes de sistemas; contar con un punto de referencia basado en información previamente generada, según los propósitos de un sistema en particular con el fin de evaluar de una forma más precisa su comportamiento; realizando pruebas para comprobar el aceptable funcionamiento del software.

La herramienta acumulará datos asociados por diferentes proyectos. Se contará con una base creciente que el desarrollador puede ir alimentando continuamente; creando un punto de referencia útil para generar datos de prueba aceptablemente apreciables por el usuario final de un proyecto en particular.

Una funcionalidad a destacar que contendrá la herramienta, es que para la contención de los datos se tendrá en cuenta la característica de integridad referencial.

Aprovechando la tecnología ODBC (Open Data Base Connectivity); estándar de acceso a la base de datos; la herramienta podrá conectarse a diferentes bases de datos como por ejemplo MySql, PostgreSql, Oracle y Microsoft SQL Server

En este libro se presenta la documentación del proceso de desarrollo de la herramienta, la cual se describe mediante el comportamiento de la metodología ágil SCRUM; la herramienta es un entorno software por la cual los usuarios alimentarán sus bases de datos con datos prueba y de este manera verificar la funcionalidad de sus proyectos software.

1. DESCRIPCIÓN DEL PROYECTO

1.1 DESCRIPCIÓN DE LA PROBLEMÁTICA

Para los desarrolladores de software puede resultar una actividad tediosa poblar la base de datos con información significativa para mostrar sus aplicaciones ante el usuario final (clientes). Generalmente esta actividad se hace alimentando manualmente la base de datos o ingresando registros desde módulos implementados específicamente.

Estos datos en la mayoría de los casos no llegan a ser adecuados para que el usuario final tenga una mejor apreciación de la funcionalidad de la aplicación; al tampoco ofrecen características equiparables con datos reales en la veracidad de los datos, en la generación de grandes volúmenes de datos y en la categorización de éstos.

Existen varias herramientas desde la web que pueden generar parcialmente datos de prueba, los cuales se pueden generar por patrones, categorías y/o combinación de datos; sin embargo, sólo operan a nivel de tablas individuales y no tienen en cuenta la integridad referencial. Es poca la ventaja que ofrecen si no se considera este aspecto.

La propuesta es desarrollar un generador de datos que considere la integridad referencial y de esta manera se pueda alimentar una base de una forma más integral que es como el usuario realmente lo requiere para probar sus aplicaciones.

Esta herramienta sería muy útil para desarrolladores de aplicaciones al momento de realizar pruebas de la funcionalidad y resultados de sus proyectos con el objetivo de mejorar la productividad ya que el desarrollador contará con una gran variedad de patrones y gran volumen de datos personalizados. Además es una herramienta útil en la academia en las áreas de ingeniería del software, base de datos ya que facilita y apoya el proceso de enseñanza y aprendizaje, gracias al gran volumen de datos se pueden realizar distintos tipos de pruebas ampliando los conocimientos de los estudiantes. Respecto al usuario final (cliente) le ofrece una idea más adecuada de la información que almacena y gestiona el producto desarrollado con el fin de que éste obtenga una noción más clara del aplicativo que va a adquirir.

1.2 OBJETIVOS

1.2.1 Objetivo general

Implementar una herramienta software que permita generar automáticamente datos de prueba para poblar una base de datos relacional a partir de características o patrones de datos previamente configurados.

1.2.2 Objetivos específicos

- Implementar un módulo para la gestión integral de categorías y patrones de datos.
- Implementar una funcionalidad que mediante conexión ODBC importe el esquema de creación de la base de datos y ofrezca al usuario las interfaces adecuadas de asociación de patrones y categorías de datos para la generación datos de prueba.
- Generar los scripts SQL con las sentencias correspondientes a la alimentación de la base de datos, con los datos de prueba.
- Elaborar la documentación correspondiente de uso y operación de la herramienta desarrollada.

2. MARCO TEÓRICO

2.1 FUNDAMENTOS TEÓRICOS

Dentro del éxito de la fase de desarrollo de software el diseño, la implementación y prueba son actividades confiables para el usuario final debido a que en ella puede predecir cómo será el comportamiento de la aplicación desarrollado cuando esté trabajando con volúmenes de datos. La herramienta software desarrollada en es este proyecto se cimentó en la teoría expuesta a continuación y la metodología SCRUM.

2.1.1 Modelado de base de datos. Los modelos de datos son una serie de conceptos los cuales describen la estructura y las operaciones de una base de datos, dentro de la estructura se puede se encuentran los conceptos de los datos, las relaciones, las restricciones y dentro de las operaciones, proceso tales como la recuperación, actualización o modificación de los datos, así como la especificación de comportamiento y la capacidad de auto-modificarse.

Un esquema de la base de datos es la descripción de una base de datos mediante un modelo de datos, se especifica durante el diseño, y se espera se generen pocos cambios en un futuro cercano.

2.1.2 Modelado relacional. Una base de datos relacional es un conjunto de tablas, cada una de las cuales se le asigna un nombre exclusivo o diferenciador. Cada fila de la tabla, presenta una relación entre un conjunto de valores, los

atributos también llamados columnas es la estructura en la que está representada la base de datos.

La estructura fundamental del modelo relacional es la relación, es decir, una tabla bidimensional constituida por filas y columnas.

Las bases de datos relacionales son un depósito de datos en que se mantiene su independencia; las ventajas más importantes que ofrece un modelo relacional son las siguientes:

- Diseño, ejecución y uso más fácil de la base de datos: gracias a que este modelo cuenta con independencia de datos e independencia estructural, hace que la creación del diseño sea mucho más fácil y la administración de su contenido también.
- Poderoso sistema de base de datos: realiza tareas más complejas disminuyendo la complejidad física del sistema tanto para el diseñador como para los usuarios finales (clientes).
- Garantiza la integridad referencial: al eliminar un registro todos los registros relacionados dependientes se eliminarán en cascada.

2.1.3 Claves primarias. En cualquier base de datos relacionales, las claves juegan un papel muy importante, ya que son las encargadas de proporcionar de una manera rápida y eficiente de buscar datos en una tabla, además permite preservar la integridad de los datos.

Antes de conocer el concepto de clave primaria, se debe definir y comprender el concepto de clave candidata ya que es ella la encargada de identificar de manera

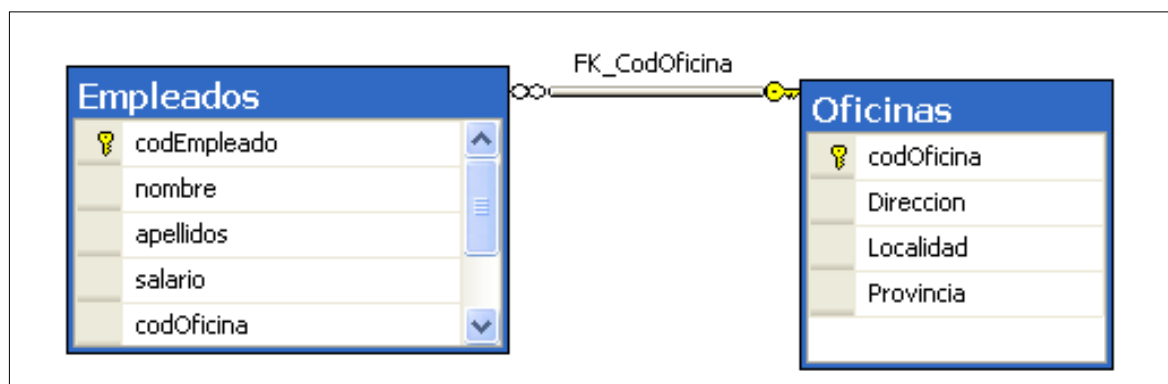
única un registro en una tabla, éstas no pueden ser nulas y su valor debe ser único.

La clave primaria es una clave candidata que tiene como propósito identificar de manera única a los registros de una tabla a través de toda la estructura de datos; su selección es muy importante en el diseño de una base de datos, ya que es un elemento fundamental de los datos que facilita la unión de tablas y el concepto total de una base de datos relacional.

La clave primaria tiene como característica principal que debe ser única y no nula, de esta manera garantiza que una fila de una tabla pueda ser siempre referenciada a través de su clave primaria.

2.1.4 Claves foráneas. Una clave foránea es un atributo de una entidad o relación que a su vez es la clave principal de otra entidad. Las claves foráneas permiten establecer uniones entre entidades. En la figura 1 se muestra un ejemplo de clave foránea.

Figura 1. Clave Foránea



Fuente: 2-Lenguaje de Definición de datos (DDL). [En línea] Disponible en internet: <http://www.adrformacion.com/cursos/aspnet35av/leccion2/tutorial2.html>

[Consultado: 02 de diciembre de 2014]

2.1.5 Dominios. *Un dominio es el conjunto de valores legales para uno o varios atributos.* Los dominios constituyen una característica muy importante del modelo relacional. Cada atributo de una base de datos relacional se define sobre un dominio, se puede presentar el caso en que varios atributos se encuentren definidos sobre el mismo dominio.

El concepto de dominio es importante porque permite que el usuario defina, en un lugar común, el significado y la fuente de los valores que los atributos pueden tomar. Esto hace que haya más información disponible para el sistema cuando éste va a ejecutar una operación relacional, de modo que las operaciones que son semánticamente incorrectas, se puedan evitar. ^[1]

2.1.6 Reglas de integridad. Una vez concretada la estructura de la bases de datos relacional, debemos garantizar mediante las reglas de integridad que los datos almacenados en dicha estructura estén bien definidos y cumplan los requisitos para que sean datos correctos.

Definido cada atributo sobre un dominio o patrón se impone una restricción sobre el conjunto de valores permitidos para cada atributo, a este tipo de restricciones se les denomina *restricciones de dominios*.

Las bases de datos relacionales deben cumplir dos reglas de integridad muy importantes las cuales se deben aplicar todo el tiempo. Estas reglas son *la regla de integridad de entidades* y *la regla de integridad referencial*; antes de comprenderlas es preciso conocer el concepto de *nulo*.

[1]UNIVERSITAT JAUME Mercedes Marqués. Bases de Datos. Estructura de datos relacional, Definiciones informales [En línea] Disponible en: <http://www.uji.es/bin/publ/edicions/bdatos.pdf> [Consultado 12 de Noviembre de 2014].

2.1.7 Nulos. Los registros pueden tener algunos atributos que no requieren que un valor sea suministrado, en estos casos se habla de valores desconocidos o nulos. Es preciso aclarar que un nulo no representa el valor cero ni la cadena vacía ya que éstos son valores que tienen significado. El nulo implica ausencia de información, bien porque al insertar el registro se desconocía el valor del atributo, o bien porque para dicho registro, el atributo no tiene sentido.

2.1.8 Regla de integridad de entidades. La primera regla de integridad se aplica a las claves primarias de las relaciones base: *ninguno de los atributos que componen la clave primaria puede ser nulo.* ^[2]

Una clave primaria por definición es una clave irreducible su principal uso es identificar de modo único las filas. Irreducible significa que ningún subconjunto de la clave primaria sirve para identificar las filas de modo único.

La parte de la clave primaria no debe ser nula, ya que si se permite se estaría indicando que no todos sus atributos son necesarios para distinguir las filas, por tal razón se está contradiciendo la irreducibilidad.

2.1.9 Regla de integridad referencial. La regla de integridad referencial se aplica a las claves ajenas: *si en una relación hay alguna clave ajena, sus valores deben coincidir con valores de la clave primaria a la que hace referencia, o bien, deben ser completamente nulos.* ^[3]

[2] UNIVERSITAT JAUME Mercedes Marqués. Bases de Datos. Regla de Integridad de Entidades [En línea] Disponible en: <http://www.uji.es/bin/publ/edicions/bdatos.pdf> [Consultado 12 de Noviembre de 2014].

[3] UNIVERSITAT JAUME Mercedes Marqués. Bases de Datos. Regla de Integridad Referencial [En línea] Disponible en: <http://www.uji.es/bin/publ/edicions/bdatos.pdf> [Consultado 15 de Noviembre de 2014].

Por lo tanto, para cualquier valor no nulo de la clave foránea existe un valor asociado en la clave primaria de la relación objetivo; la regla de integridad referencial se aplica a la relación existente entre la clave ajena y la clave primaria, y no entre la clave ajena y la clave candidata. En conclusión, la definición de clave ajena y de integridad referencial está estrechamente relacionada.

Las relaciones de claves foráneas se describen como relaciones padre/hijo y en el caso en el que un registro el padre no exista se le llama un registro huérfano. Al presentarse esta situación, se dice que se tiene una integridad referencial pobre, lo que significa que probablemente va ligado a un mal diseño, lo que puede conllevar a otros problemas en la base de datos, en la mayoría de los casos se debe evitar que estas situaciones se presenten con el fin de lograr una buena integridad referencial.

2.2 PROTOCOLO DE CONECTIVIDAD DE BASE DE DATOS.

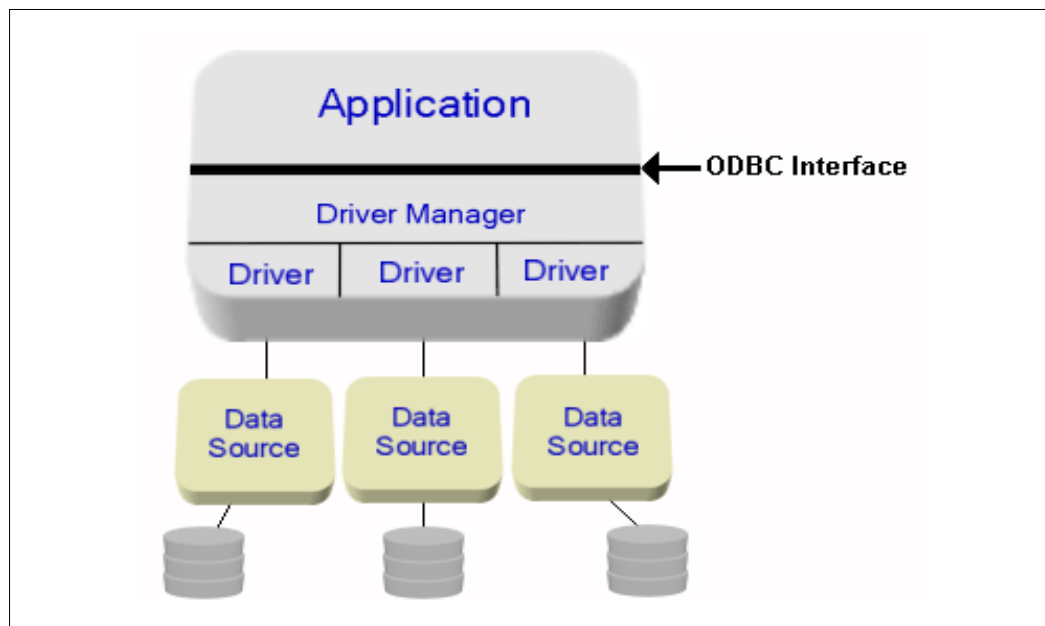
Los protocolos de conectividad de bases de datos tiene como objetivo principal conectarse a la base de datos SQL y en ella realizar funciones tales como importar, exportar o adjuntar los datos almacenados en dichas bases de datos.

2.2.1 Conectividad abierta de bases de datos (ODBC). Es el protocolo de conectividad de base de datos de Microsoft, se llevó a cabo en 1992 por Microsoft para sus sistemas operativos, con el tiempo por la intervención de otros editores se usa en sistemas operativos como Unix y en plataformas Java, tiene como objetivo obtener el acceso a datos almacenados en tanto modelos relacionales, en

sistemas de gestión de base de datos no relacionales y en sistemas de administración de bases de datos.

ODBC proporciona una forma abierta, independiente del proveedor de acceso a datos almacenados en una variedad de equipo propietario, minicomputadoras y base de datos de mainframe.

Figura 2. Conectividad con ODBC



Fuente: Conectividad con ODBC. [En línea] Disponible en internet: <http://www.web20.co/desarrollo/conectividad-con-odbc/> [Consultado: 06 de diciembre de 2014]

La arquitectura de ODBC se compone de cuatro componentes principales: la interfaz de cliente, el gestor de controladores ODBC, los controladores ODBC, y las fuentes de datos.

ODBC permite que un programa en un computador, mediante un proceso único maneje múltiples bases de datos en los sistemas de gestión de base de datos, sus principales características son:

- ODBC contiene una base de datos de registro, es decir, una fuente de datos, los cuales se encuentran disponibles en el computador del usuario.
- La interfaz gráfica permite al usuario agregar bases de datos para registrar.
- Los controladores de ODBC dan aplicaciones a una serie de métodos que pueden utilizar para acceder a base de datos, estos controladores también ofrecen a las herramientas la posibilidad de recuperación de datos y mayor nivel de actualización de información en la base de datos.
- El software de ODBC Microsoft incluye controladores para Access, FoxPro y DBMS de SQL Server.

3. DESARROLLO E IMPLEMENTACIÓN DE LA METODOLOGÍA DEL PROYECTO

3.1 METODOLOGÍA SCRUM

La metodología de desarrollo es un marco de trabajo con el objetivo de estructurar, planificar y controlar los procesos de un proyecto y así organizar los distintos procedimientos que se ejecutan a lo largo de un proyecto.

El presente proyecto se desarrolla a través de la metodología ágil SCRUM; ésta se basa en procesos iterativos e incrementales produciendo en espacios cortos de tiempo la constante retroalimentación y la recopilación de historias, aumentando la productividad y satisfaciendo las distintas necesidades del cliente en el menor tiempo posible, proporcionando un mayor valor al negocio.

Las historias de usuario presentadas desde la perspectiva del usuario describen las características esperadas del software que se va a desarrollar dando detalle a las funcionalidades y requisitos, definiendo sus prioridades y el tiempo en que se ejecutan.

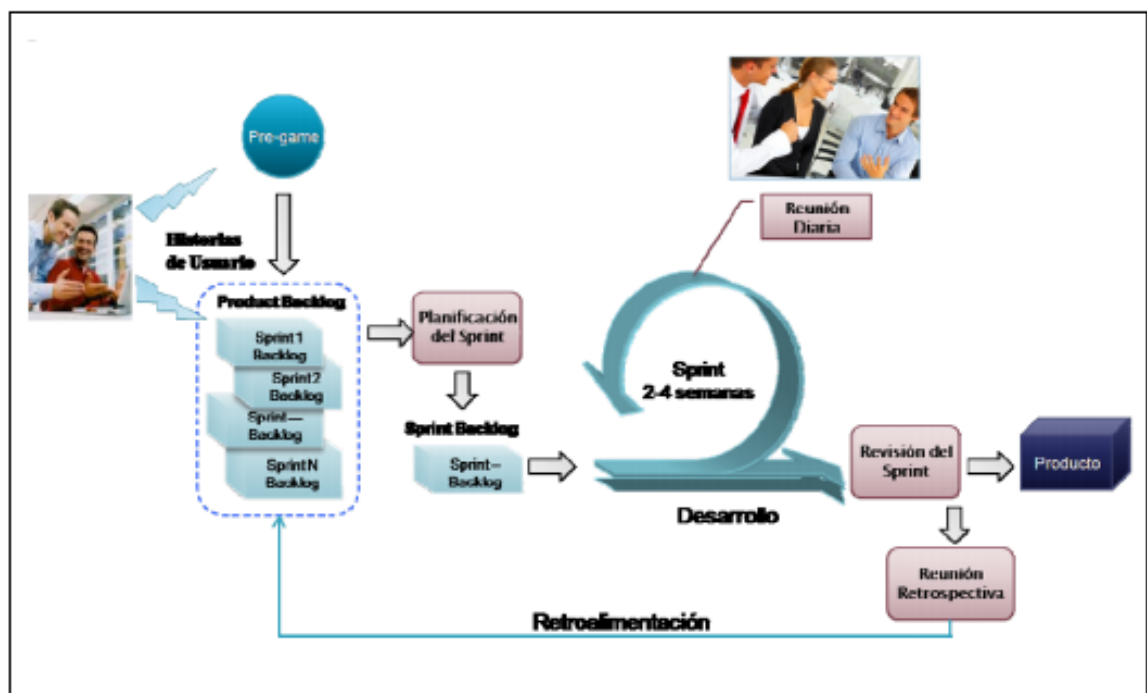
Scrum es un proceso para la gestión y el control del producto, su principal objetivo es que las personas que forman parte del equipo trabajen unidos y de manera eficiente, aportando sus habilidades individuales a la elaboración de un software de buena calidad.

Scrum tiene como característica más importante la entrega de porciones incrementales del producto al finalizar cada iteración (Sprint); así, el cliente

mediante reuniones periódicas puede solicitar se realicen modificaciones, verificaciones y pruebas los cuales dan evolución y continuación al desarrollo del software.

La siguiente figura presenta el modelo del ciclo de vida del desarrollo aplicando SCRUM en un producto software:

Figura 3. Modelo ciclo de vida del desarrollo aplicando SCRUM



Fuente: Rodríguez González Pilar. SCRUM [En línea] Disponible en: http://oa.upm.es/1939/1/TESIS_MASTER_PILAR_RODRIGUEZ_GONZALEZ.pdf página 17 [Consultado: 16 de Diciembre de 2014].

Con base a lo anterior, se evidenciará la aplicación de la metodología Scrum a la herramienta software desarrollada en este proyecto:

3.1.1 Caracterización del proceso software. El objetivo del proyecto es desarrollar una herramienta que pueda ayudar a desarrolladores de software a poblar sus bases de datos con datos de prueba y poder obtener mejores resultados en las pruebas de sus desarrollos software y en las presentaciones que se realizan frente al cliente o a los interesados en adquirir sus aplicaciones.

El proceso inicia definiendo las historias de usuario las cuales describen las características que el usuario desea que tenga la herramienta que se va a desarrollar, están relacionadas con las funcionalidades del sistema, las restricciones, el rendimiento, entre otras. Dichas características son necesarias para garantizar que cumplirá las expectativas del cliente y además la responsabilidad de satisfacer las necesidades del usuario desarrollador que utilice la herramienta.

De este proceso, se obtiene, lo que se denomina en SCRUM el “Product Backlog”, el cual consiste en listar las tareas que identifican todas las historias de usuario identificando las de mayor prioridad y asignándole una continuidad u orden; logrando así una planificación inicial del proyecto.

Como resultado de esta planificación, se realizó la separación de las tareas en cinco iteraciones de corta duración, no superiores a cuatro semanas, denominadas “Sprints”. En cada Sprint se realiza una reunión de planificación definiendo las tareas y el tiempo en que se deben llevar a cabo. El objetivo es establecer un conjunto de tareas a desarrollar denominado “Sprint Backlog” donde se incluye diseño, desarrollo, pruebas e integración y así determinar la funcionalidad de cada Sprint, garantizando la evolución del producto.

Finalizando cada Sprint se obtiene parte del producto, en este caso un módulo, este entregable muestra los avances realizados; en este punto se procede a una revisión, donde sí es necesario se incorporarán nuevas historias de usuario para

3.1.2 Fases y actividades. La realización de un Sprint tiene como objetivo conseguir un incremento del producto, el cual será productivo para el Product Owner (Persona que toma las decisiones y conoce el negocio del cliente y la visión del producto).

El desarrollo del proyecto se llevó a cabo a través de tres fases, dentro de cada fase se realizan ciertas tareas o actividades las cuales garantizan el incremento del producto y el cumplimiento de las necesidades del interesado.

Las fases y actividades se encuentran relacionadas con el estándar IEEE 1219.

1. Fase de planificación: Es la fase encargada de la planificación inicial, en ella se analiza el negocio, los requerimientos y es donde se consideran las actividades, las prioridades y los tiempos en que se van a realizar cada uno de los Sprint.

- **Identificación y clasificación del problema:** En esta actividad con la colaboración de las historias de usuario se identifica el problema que se desea solucionar y se determinan las diferentes tareas que se van a llevar a cabo durante la realización del producto; asignando a cada tarea un prioridad y estimando un tiempo de ejecución.
- **Análisis:** Realizando el análisis de la identificación del problema y los requisitos que requiere el desarrollo del producto para garantizar las funcionalidades, se desarrolla un plan preliminar de diseño, implementación, pruebas y liberación del producto.

2. Fase de desarrollo: En el cumplimiento de las reuniones diarias, se busca gestionar el riesgo en cada iteración y dar soluciones posibles a los problemas que puedan impedir un avance normal en cada uno de los Sprint.

- **Diseño:** El trabajo en el diseño es obtener el esquema que represente las características requeridas del producto.
- **Implementación:** En la implementación el interés principal es la integración de los Sprints dentro del producto, garantizando la evolución de la herramienta.
- **Pruebas del sistema:** Esta actividad consiste en realizar las pruebas que validarán el correcto funcionamiento de las funcionalidades implementadas. Esta actividad es de especial interés para el cliente del producto ya que garantizará que la herramienta está cumpliendo con los requerimientos solicitados.

3. Fase de finalización: Finalizado el proceso de construcción del producto, es decir, cuando la lista de tareas se hayan cumplido a cabalidad, se procede a realizar las siguientes tareas:

- **Pruebas de aceptación:** Esta actividad es la encargada de realizar pruebas sobre el sistema completamente integrado garantizando la funcionalidad de los distintos módulos que componen el producto.
- **Liberación del producto:** La liberación del producto consiste en un plan donde se desarrolla el manual de uso de la herramienta desde la instalación hasta una capacitación.

3.1.3 Diseño e implementación. La planificación inicial del proyecto permitió determinar las tareas que se van a ejecutar en los distintos Sprints, en este caso se decidió establecer cinco Sprints, el primero manifiesta los requerimientos del cliente, los tres siguientes módulos componen el paso a paso de la metodología de la herramienta y el último la generación de los scripts SQL con las sentencias correspondientes a la alimentación de la base de datos.

- Sprint 1: Concepción del Proyecto
- Sprint 2: Módulo Gestión de Patrones
- Sprint 3: Módulo Creación de la Base de Datos
- Sprint4: Módulo Creación de Proyectos
- Sprint 5: Módulo Generación de los scripts SQL con las sentencias correspondientes a la alimentación de la base de datos.

Las funcionalidades a implementar en cada Sprint describen los requisitos del producto a partir de las historias de usuario determinadas por el Propietario del Producto (Product Owner). El objetivo del proyecto es la generación de los scripts SQL con las sentencias correspondientes a la alimentación de la base de datos y se obtendrá a partir de la buena realización de los primeros cuatro Sprints.

3.1.3.1 Descripción de los módulos del sistema

A continuación se describe el desarrollo de la herramienta software para la alimentación de datos pruebas para una base de datos. Finalmente la realización del producto consta de cinco Sprints; por un lado, el diseño de la herramienta y, por el otro, la evolución del producto tras la finalización de cada uno de los Sprints.

La siguiente figura refleja los módulos en que está compuesto el sistema, el objetivo principal del proyecto es la generación de los scripts SQL con las sentencias correspondientes a la alimentación de la base de datos.

Figura 5. Módulos del Sistema



Tabla 1. Descripción Módulos del Sistema

Módulo	Descripción
Concepción del Proyecto	En este módulo agrupa las historias de usuario con el objetivo de realizar un análisis preliminar y la identificación del problema
Gestión de Patrones	Este módulo agrupa las distintas funcionalidades de administración en la gestión de patrones, permite la creación, adición, modificación y/o eliminación de patrones.
Gestión de la Base de Datos	Este módulo agrupa las distintas funcionalidades de administración en la gestión de la base de datos, permite la creación, adición, modificación y/o eliminación de base de datos. Una vez creados los patrones inicia la población de datos en la base de datos con los datos prueba.

Gestión de Proyectos	Este módulo agrupa las distintas funcionalidades de administración en la gestión de proyectos, permite la creación, adición, modificación y/o eliminación de proyectos.
Generación de los Scripts para las sentencias SQL para la alimentación de la Base de Datos	El objetivo de la herramienta es la generación de los scripts para las sentencias SQL para la alimentación de la base de datos.

3.1.3.1.1 Módulo de concepción del proyecto. En esta primera etapa por medio de la metodología de Scrum se pretende resumir las historias de usuario. El objetivo es descubrir las expectativas, los alcances, las necesidades a satisfacer y las características de operación de funcionamiento.

Expectativas

- Una herramienta que facilite la población de datos prueba para una base de datos la cual el usuario asigne los datos según sus necesidades.
- Una herramienta que favorezca al desarrollador en la entrega de sus productos ya que con datos semejantes a los reales el cliente puede reconocer las bondades del software a adquirir y por lo tanto realizar mejoras óptimas
- Una herramienta que ayude a los estudiantes en la entrega de sus trabajos de clase ya que con la diversidad de datos podrán consultar, agregar, eliminar y modificar información.
- Una herramienta de apoyo en el proceso de enseñanza - aprendizaje para las asignaturas relacionadas con bases de datos.

Alcances

- Se inicia con la acumulación de información suministrada en las historias de usuario.
- Creación de Patrones, son los dominios para determinar la clasificación de los datos
- Definición a la conexión de la base de datos al gestor de datos, la cual se poblará con los datos establecidos previamente.
- Creación de Proyectos, este consiste en definir el tipo de proyecto para dar uso a los datos correctos especificados por el desarrollador.
- Generación de los scripts para la generación de datos de prueba.

Necesidades a Satisfacer

- **Para el Desarrollador**
 - Evitar continuamente alimentar una base de datos manualmente para pruebas de software.
 - Generar datos a volúmenes que requieran los desarrolladores en sus aplicaciones.
 - Categorizar datos para delimitar los dominios.
- **Para el Docente**
 - Una herramienta que apoye sus conocimientos y argumentos ante los estudiantes.
- **Para el Estudiante**
 - Una herramienta que le permita experimentar, verificar y comprobar.
- **Para el Cliente**
 - Información adecuada en el software que desea adquirir.
 - Visualizar con mayor claridad la gestión que realiza el producto desarrollado
 - Mayor optimización en los cambios que considere con el fin de obtener mejor rendimiento.

Características de Operación de Funcionamiento

- **Requisitos de Instalación**
 - Sistema Operativo: Windows 7 o Superior
 - Espacio Disco: 100 MB
 - RAM: 2 GB
- **Requisitos de Operación**
 - Se Requiere que el desarrollador defina previamente la cadena de conexión ODBC
 - Se Requiere que el administrador de la base de datos ejecute los scripts generados.

3.1.3.1.2 Módulo de gestión de patrones

Crear Patrones

- Esta funcionalidad permite al usuario crear la cantidad de patrones que desee y los datos que considere necesarios para la población de las bases de datos.

Adicionar Patrones

- Esta funcionalidad permite al usuario adicionar datos a los distintos patrones con el objetivo de aumentar los datos prueba.

Modificar Patrones

- Esta funcionalidad permite al usuario renombrar los patrones anteriormente creados.

Eliminar Patrones

- Esta funcionalidad permite al usuario eliminar patrones creados anteriormente. En el caso de eliminar el patrón el usuario eliminaría los datos existentes en dicho patrón.

3.1.3.1.3 Módulo de gestión de la base de datos

Crear Base de Datos

- Esta funcionalidad permite al usuario seleccionar la base de datos que tiene relación directa, a través de los patrones anteriormente creados poblarlas para realizar las tareas de pruebas correspondientes.

Adicionar Base de Datos

- Esta funcionalidad permite al usuario adicionar las tablas de las bases de datos seleccionada, con las actualizaciones de atributos creados a través de las mejoras existentes.

Modificar Base de Datos

- Esta funcionalidad permite al usuario renombrar las bases de datos y realizar cambios en los atributos para realizar modificaciones de los datos.

Eliminar Base de Datos

- Esta funcionalidad permite al usuario eliminar bases de datos anteriormente creadas. En este caso eliminaría los datos existentes en la base de datos, ya que para eliminar definitivamente la base de datos debe hacerse directamente desde el programa gestor de datos.

3.1.3.1.4 Módulo de gestión de proyectos

Crear Proyecto

- Esta funcionalidad permite al usuario crear el proyecto indicando el gestor de datos al que pertenece la base de datos que se poblará con los datos prueba.

Adicionar Proyecto

- Esta funcionalidad permite al usuario adicionar los proyectos que desee a la herramienta y así obtener gran cantidad de datos para posteriormente ser usados en otros proyectos.

Modificar Proyecto

- Esta funcionalidad permite al usuario renombrar el proyecto , cambiar el programa gestor de datos y por lo tanto sus características.

Eliminar Proyecto

- Esta funcionalidad permite al usuario eliminar el proyecto anteriormente creado.

3.1.3.1.5 Generación de los scripts para las sentencias SQL para la alimentación de la base de datos

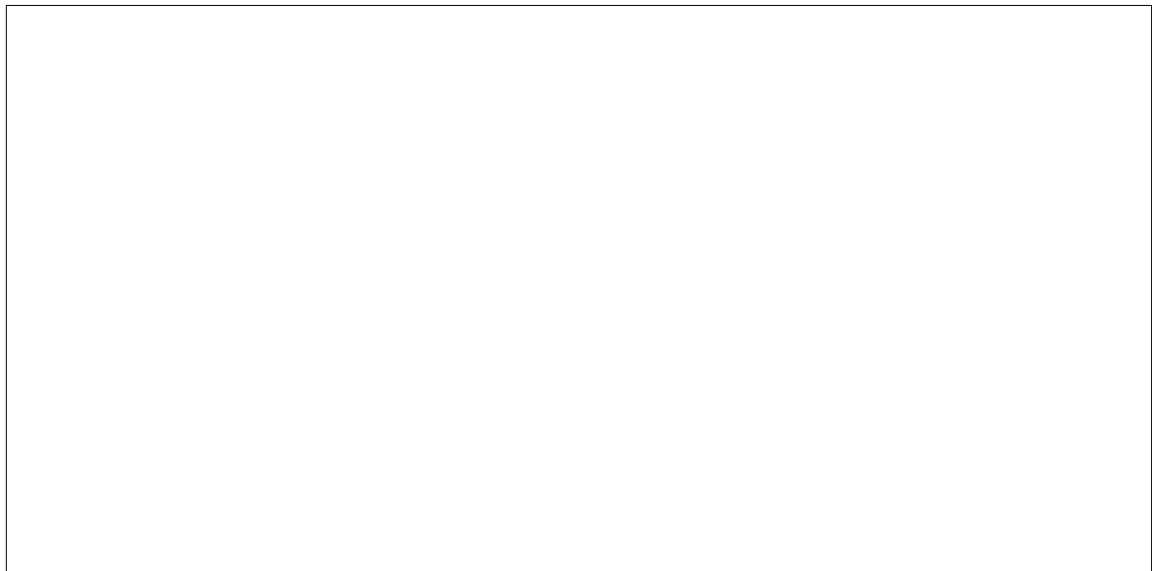
Generación de Scripts para las Sentencias SQL para la alimentación de la Base de Datos

- Esta funcionalidad permite generar las sentencias para la alimentación de la base de datos a diferentes generadores de bases de datos por ejemplo MySQL, PostgreSQL, Oracle, Microsoft SQL Server, es decir, los generadores que puedan usar la tecnología ODBC tendrán acceso para el uso de los scripts generados por la herramienta.

3.1.4 Entregables del proyecto. La herramienta consta de alrededor de cuatro (4) módulos, las cuales permiten gestionar los diferentes componentes del sistema.

En la figura 7 muestra los tiempos a partir del comienzo y finalización para los distintos sprints completados en el proyecto. El proyecto tuvo una duración de 17 semanas, es decir aproximadamente cuatro meses y medio.

Figura 6. Línea de Tiempo del Proyecto



A continuación se listará por módulo los entregables realizados al cliente durante la ejecución del proyecto.

3.1.4.1 Sprint 1. Concepción del proyecto. En este Sprint se analizaron las características del producto y se realizó un resumen de requerimientos donde se especifican las diferentes funcionalidades que ofrecerá el proyecto. En esta etapa se elaboró el diseño y la configuración del entorno de ejecución de la herramienta y se analizaron las validaciones iniciales.

3.1.4.2 Sprint 2. Módulo de patrones. En esta etapa, el objetivo es desarrollar el módulo según las características especificadas en la concepción del proyecto; para ello se realizaron los siguientes entregables:

Entregables:

- Diseñar y programar vistas
- Secuencia de Patrones
- Crear nuevo Patrón
- Modificar Patrón
- Eliminar Patrón
- Editar detalles de los Patrones
- Adicionar detalles en los Patrones existentes
- Modificar detalles en los Patrones existentes
- Elimiar detalles en los Patrones existentes
- Validar campos vacios en los detalles de los patrones
- Módulo funcionando según las características especificadas.

3.1.4.3 Sprint 3. Módulo de base de datos. En esta etapa, el requisito principal es realizar la conexión al gestor de datos con el objetivo de recuperar las tablas y los atributos. El módulo debe cumplir según las características especificadas en la concepción del proyecto; para ello se realizaron los siguientes entregables:

Entregables:

- Diseñar y programar vistas
- Realizar conexión de la base de datos mediante sentencias según gestor de Base de Datos.
- Recuperar mediante sentencias SQL los atributos de las tablas.
- Seleccionar los datos que van a contener dichos atributos con el fin de realizar las pruebas.
- Preparar los datos según patrones definidos anteriormente.
- Eliminar los atributos si corresponde.
- Validar campos vacíos en los atributos que contendrán datos de prueba.
- Módulo funcionando según las características especificadas.

3.1.4.4 Sprint 4. Módulo de proyectos. En esta etapa, el objetivo es desarrollar el módulo según las características especificadas en la concepción del proyecto. El usuario a medida que crea proyectos aumentará la cantidad de datos prueba para próximos proyectos; para para ello se implementarán los siguientes entregables:

Entregables:

- Diseñar y programar vistas
- Crear Proyecto (Consecutivo de Proyectos)
- Abrir Proyecto
- Adicionar Proyecto
- Modificar y/o eliminar proyecto
- Módulo funcionando según las características especificadas.

3.1.4.5 Sprint 5. Módulo de generación de los scripts para las sentencias SQL para la alimentación de la base de datos. La etapa principal del proyecto, ya que las sentencias SQL contienen los datos que el usuario usará en sus proyectos para realizar las pruebas; para para ello se implementaron los siguientes entregables:

Entregables

- Diseñar y programar vistas.
- Generar las sentencias SQL para la alimentación de la Base de Datos.
- Scripts SQL para el cargue de datos.
- Módulo funcionando según las características especificadas.

3.1.4.6 Product backlog. El product backlog o listado de producto, es el resultado que contiene las historias de usuario para convertirlas en elementos de una lista de objetivos del producto, son los requisitos que debe cumplir el sistema, se indicará el valor que tiene para el cliente y el coste estimado con el fin de priorizar las tareas, los cuales guiarán el proceso de desarrollo. En la siguiente tabla se describe el listado del producto a desarrollar, cada tarea se le asigna la prioridad y su tiempo para identificar la importancia y el tiempo que requerirá realizar cada tarea:

Tabla 2. Listado de Producto

Descripción de requisitos, prioridades y tiempos estimados			
Requisitos	Descripción	Prioridad de Desarrollo	Tiempo estimado (días)
Gestionar Patrón	El sistema permitirá crear, adicionar, modificar y eliminar patrones; el usuario podrá incluir las cantidades de patrones que desee.	ALTA	15
Crear detalles de patrones	El sistema permitirá crear, adicionar, modificar y eliminar detalles en los patrones, es decir, podrá especificar estos detalles a su elección.	ALTA	12
Gestionar Bases de Datos	El sistema permitirá crear, adicionar, modificar y eliminar las bases de datos.	ALTA	12
Gestionar Proyecto	El sistema permitirá crear, abrir, adicionar, modificar y eliminar proyectos.	ALTA	15
Recuperar Tablas	El sistema permitirá importar las tablas del gestor de la base de datos.	ALTA	10

Descripción de requisitos, prioridades y tiempos estimados			
Requisitos	Descripción	Prioridad de Desarrollo	Tiempo estimado (días)
Recuperar Atributos	El sistema permitirá importar los atributos de las tablas del gestor de la base de datos para iniciar la población de datos.	ALTA	12
Asociar patrones a los atributos	El sistema permitirá la asociación de patrones para especificar los requerimientos de los atributos.	ALTA	15
Generación los scripts para las sentencias SQL para la alimentación de las bases de datos.	El sistema permitirá generar los scripts de las sentencias SQL necesarias para la creación de las bases de datos, éste será el resultado de los pasos anteriormente descritos.	MEDIA	21

4. EL PRODUCTO Y SU DOCUMENTACIÓN

4.1 CARACTERÍSTICAS DEL PRODUCTO DURANTE SU DESARROLLO

Actualmente, los desarrolladores en la creación de sus herramientas necesitan realizar pruebas para verificar el buen funcionamiento de sus programas. Los datos necesarios para dichas pruebas deben ser de un volumen lo suficientemente amplio ya que el objetivo es garantizar en los distintos módulos o procesos que contenga la herramienta su correcto funcionamiento.

Los generadores de datos prueba ofrecidos por la web brindan grandes beneficios; además de ser gratuitos; permiten generar volúmenes de datos con hasta 200 registros; realizar exportación de datos en diferentes tipos de bases de datos y la posibilidad de utilizar diferentes tipos de datos de uso general, tales como nombres, direcciones de domicilio, números de teléfono, ciudades, países y muchas más. Estos registros son generados considerando independiente las tablas lo que genera inconvenientes cuando existe integridad referencial.

La herramienta diseñada en este proyecto tiene como objetivo ofrecer las ventajas anteriormente mencionadas, además de una personalización según las necesidades del usuario. ¿Cuáles son las ventajas de esta personalización?; en ocasiones los programas a desarrollar deben contener datos dirigidos a universos más específicos, por ejemplo a universidades, ferreterías, farmacias, entre otros, y estos datos el usuario podrá de una manera sencilla acomodarlos al patrón de tipo de dato correspondiente y así alimentar la base de datos para trabajar con ella y

comprobar el correcto funcionamiento de sus creaciones. Adicionalmente la herramienta incorporará la funcionalidad para controlar la integridad referencial.

El objetivo principal de la herramienta la generación de los scripts para sentencias SQL y así lograr la alimentación de la base de datos con datos pruebas de características muy semejantes a los datos necesarios en dichos programas creados.

4.1.1 Construir un patrón. Un patrón es la descripción de un atributo que forma parte de una entidad. Es el conjunto de valores que puede tomar el atributo.

La tabla refleja los dominios de los atributos de la entidad PERSONA.

Tabla 3. Dominios de los atributos de la relación que almacena los datos de la entidad PERSONA

Entidad: PERSONA			
Atributo	Patrón	Descripción	Definición
IdPersona	CodIdent_dom	Posibles Números de identificación de la persona	Números de 7 a 10 dígitos
Nombre	Nombre_dom	Nombres de personas: apellido1 – Nombre	50 caracteres
Dirección	Dirección_dom	Domicilios: Casa, número	50 caracteres

El módulo de patrones permite adicionar, modificar y/o eliminar patrones; éste módulo adicionalmente a las descripciones permite igualmente gestionar los detalles de los patrones, éstos detalles son valores particulares de cada patrón, es decir, definir si serán números, direcciones, ciudades, nombres de personas o animales, entre otros más.

4.1.1.1 Aplicar detalles a los patrones. En este paso se busca lograr un detalle de los patrones escogido por el usuario; permitiendo identificar los tipos de dato y distintas opciones que podrían llenar dichos campos, es decir, para la misma entidad PERSONA, en el atributo de nombres, el usuario puede crear según su necesidad distintos valores de datos. En la siguiente tabla se muestran algunos posibles ejemplos que se pueden usar dependiendo el dato que se desee obtener.

Tabla 4. Ejemplos de patrones y valores que se pueden crear en un atributo.

Entidad: Persona		
Atributo	Patrón	Ejemplos
Nombre	Nombre_dom	James (Nombres de Hombres) Salomé (Nombres de Mujeres) Alex (Cualquier Sexo) James Rodriguez (Nombre Hombre – Apellido) Salomé Rodriguez (Nombre Mujer – Apellido) Alex Rodriguez (Cualquier Sexo – Apellido)

4.1.2 Identificar el gestor de base de datos. En este paso es muy importante que el desarrollador para los diferentes gestores de datos, ingrese correctamente las sentencias SQL correspondientes a las consultas de tablas y a las consultas de campos según el gestor de base de datos

Las bases de datos para las que se generarán valores en esta herramienta, corresponden a un servidor de datos específico o particular. La conexión se realiza a través del estándar de acceso a base de datos ODBC. Previamente el desarrollador debe configurar el ODBC correspondiente.

4.1.3 Realizar conexión a un proyecto. Un proyecto es la descripción de una conexión a la base de datos de un software existente mediante ODBC.

El proyecto recupera la estructura de la base de datos a nivel de tablas diferenciando características como clave principal, claves foráneas y valores nulos. Esta es la base de referencia para generar los datos con los que se poblará la base de datos.

4.1.4 Generación de scripts para las sentencias SQL para la alimentación de datos con datos prueba. En este módulo se realiza la generación de scripts para las sentencias SQL para la alimentación de datos prueba. Desde este módulo se generan los datos que van a poblar las bases de datos asociada a un proyecto registrada en el banco de proyectos. Este paso contempla las siguientes fases:

4.1.4.1 Fase de preparación: En esta fase el sistema recupera el esquema de la base de datos y le ofrece al usuario la posibilidad de personalizar cada atributo de las tablas mediante de la asociación del patrón más adecuado existente en el banco de patrones

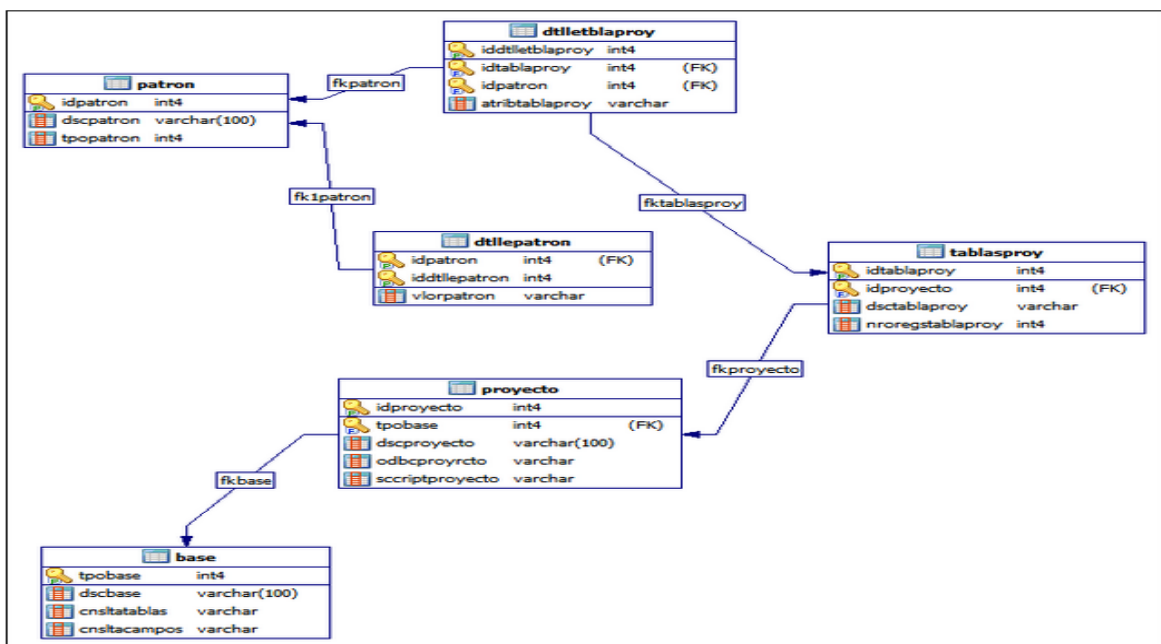
4.1.4.2 Fase de generación: Recorrerá las tablas en el orden de la integridad referencial, definido este orden con las características creadas por el usuario se generan los datos prueba para la alimentación de la base de datos.

4.2 RESULTADOS OBTENIDOS

La aplicación de la metodología de SCRUM permite en la finalización de cada Sprint observar los avances realizados y antes de continuar en la evolución de la herramienta verificar mediante un realimentación las posibles mejoras, validar las funcionalidades y de esta manera iterativa e incremental mantener la calidad del producto con el objetivo de satisfacer las necesidades del cliente.

4.2.1 Estructura de la base de datos. Una estructura de la Base de datos correctamente diseñada es una pieza vital para la gestión efectiva de la base de datos. Permite planificar la información relevante que se desea almacenar, referenciar los tipos de datos, las relaciones entre las tablas y las restricciones logrando la integridad de los datos, evitando la redundancia de los mismos. La estructura de la BD se diseñó empleando el Modelo Relacional de Datos en el motor de base de datos PostgreSQL.

Figura 7. Diagrama Relacional



4.2.1.1 Registro de patrones de datos.

Tabla 5. Descripción Tabla: patron

Columna	Tipo de Dato	Clave Principal	No Nulo	Descripción
idpatron	Int4	✓	✓	Identificador del patrón
dscpatron	Varchar(100)	-	-	Descripción del patrón
Tpopatron	Int4	-	-	Tipo de dato del patrón: 1. Texto 2. Numérico 3. Fecha 4. Otro

Figura 8. Script para la creación de la Tabla: patrón

```

CREATE TABLE "patron" (
    "idpatron" SERIAL NOT NULL,
    "dscpatron" varchar(100),
    "tpopatron" int4,
    CONSTRAINT "pkpatron" PRIMARY KEY("idpatron")
)
WITH (
    OIDS = False
);

ALTER TABLE "patron" OWNER TO "postgres";

COMMENT ON TABLE "patron" IS 'Registro de patrones de datos';

COMMENT ON COLUMN "patron"."idpatron" IS 'Identificador del patrón';

COMMENT ON COLUMN "patron"."dscpatron" IS 'descripción del patrón';

COMMENT ON COLUMN "patron"."tpopatron" IS 'Tipo de dato del patrón:
1. Texto
2. Numérico
3. Fecha
4. Otro';

```

4.2.1.2 Detalle de valores para los patrones

Tabla 6. Descripción Tabla: dtllepatron

Columna	Tipo de Dato	Clave Principal	No Nulo	Descripción
Idpatron	Int4	✓	✓	Identificación del patrón
Iddtllepatron	Int4	✓	✓	Identificación del detalle patrón
Vlorpatron	Varchar	-	-	Valor del detalle del patrón
Referencia				
Tabla Padre			Enlace	
patron			idpatron = idpatron	

Figura 9. Script para la creación de la Tabla: dtllepatron

```

CREATE TABLE "dtllepatron" (
    "idpatron" int4 NOT NULL,
    "iddtllepatron" int4 NOT NULL,
    "vlorpatron" varchar,
    CONSTRAINT "pkdtlle" PRIMARY KEY("idpatron","iddtllepatron"),
    CONSTRAINT "fkpatron" FOREIGN KEY ("idpatron")
        REFERENCES "patron"("idpatron")
    MATCH SIMPLE
    ON DELETE NO ACTION
    ON UPDATE NO ACTION
    NOT DEFERRABLE
)
WITH (
    OIDS = False
);

ALTER TABLE "dtllepatron" OWNER TO "postgres";

COMMENT ON TABLE "dtllepatron" IS 'Detalle de valores para los patrones';

COMMENT ON COLUMN "dtllepatron"."idpatron" IS 'Identificación del patrón';

COMMENT ON COLUMN "dtllepatron"."iddtllepatron" IS 'Identificación del detalle patrón';

COMMENT ON COLUMN "dtllepatron"."vlorpatron" IS 'Valor del detalle del patrón';

```

4.2.1.3 Base de datos disponibles

Tabla 7. Descripción Tabla: base

Columna	Tipo de Dato	Clave Principal	No Nulo	Descripción
tpobase	Int4	✓	✓	Identificador de la base de datos
dscbase	Varchar(100)	-	-	Descripción de la base de datos
cnsltatablas	Varchar	-	-	Instrucción SQL para recuperar las tablas
cnsltacampos	Varchar	-	-	Instrucción SQL para recupera atributos

Figura 10. Script para la creación de la Tabla: base

```
CREATE TABLE "base" (  
    "tpobase" SERIAL NOT NULL,  
    "dscbase" varchar(100),  
    "cnsltatablas" varchar,  
    "cnsltacampos" varchar,  
    CONSTRAINT "pkbase" PRIMARY KEY("tpobase")  
)  
WITH (  
    OIDS = False  
);  
  
ALTER TABLE "base" OWNER TO "postgres";  
  
COMMENT ON TABLE "base" IS 'Bases de datos disponibles';  
  
COMMENT ON COLUMN "base"."tpobase" IS 'Identificador de la base de datos';  
  
COMMENT ON COLUMN "base"."dscbase" IS 'Descripción de la base de datos';  
  
COMMENT ON COLUMN "base"."cnsltatablas" IS 'Instrucción sql para recuperar las tablas';  
  
COMMENT ON COLUMN "base"."cnsltacampos" IS 'Instrucción sql para recuperar los atributos de cada tabla';
```

4.2.1.4 Registro de proyectos

Tabla 8. Descripción Tabla: proyecto

Columna	Tipo de Dato	Clave Principal	No Nulo	Descripción
Idproyecto	Int4	✓	✓	Identificador del proyecto
tpobase	Int4	-	-	Base de datos asociada al proyecto
dscproyecto	Varchar(100)	-	-	Descripción del proyecto
odbcproyrcto	Varchar	-	-	Cadena de conexión ODBC
sccriptproyecto	Varchar	-	-	Script para la generación de datos.
Referencia				
Tabla Padre		Enlace		
base		tpobase=tpobase		

Figura 11. Script para la creación de la Tabla: proyecto

```

CREATE TABLE "proyecto" (
    "idproyecto" SERIAL NOT NULL,
    "tpobase" int4,
    "dscproyecto" varchar(100),
    "odbcproyrcto" varchar,
    "sccriptproyecto" varchar,
    CONSTRAINT "pkproyecto" PRIMARY KEY("idproyecto"),
    CONSTRAINT "fkbase" FOREIGN KEY ("tpobase")
        REFERENCES "base"("tpobase")
        MATCH SIMPLE
        ON DELETE NO ACTION
        ON UPDATE NO ACTION
        NOT DEFERRABLE
)
WITH (
    OIDS = False
);

ALTER TABLE "proyecto" OWNER TO "postgres";

COMMENT ON TABLE "proyecto" IS 'Registro de proyectos';
COMMENT ON COLUMN "proyecto"."idproyecto" IS 'Identificación del proyecto';
COMMENT ON COLUMN "proyecto"."tpobase" IS 'Base de datos asociada al proyecto';
COMMENT ON COLUMN "proyecto"."odbcproyrcto" IS 'Cadena de conexión ODBC';
COMMENT ON COLUMN "proyecto"."sccriptproyecto" IS 'Script para la generación de datos';

```

4.2.1.5 Detalle de atributos por tabla de proyecto

Tabla 9. Descripción Tabla: dtlletblaproy

Columna	Tipo de Dato	Clave Principal	No Nulo	Descripción
iddtlletblaproy	Int4	✓	✓	Identificador del detalle de tabla de tabla para proyecto
idtablaproy	Int4	-	-	Identificador de la tabla dentro del proyecto
idpatron	Int4	-	-	Identificación del patrón asociado al atributo
atribtablaproy	Varchar	-	-	Nombre del atributo de la tabla
Referencia				
Tabla Padre			Enlace	
patron			idpatron = idpatron	
tablasproy			Idtablaproy = idtablaproy	

Figura 12. Script para la creación de la Tabla: dtlletblaproy

```

CREATE TABLE "dtlletblaproy" (
    "iddtlletblaproy" int4 NOT NULL,
    "idtablaproy" int4,
    "idpatron" int4,
    "atribtablaproy" varchar,
    CONSTRAINT "pkdtlletblaproy" PRIMARY KEY("iddtlletblaproy"),
    CONSTRAINT "fkpatron" FOREIGN KEY ("idpatron")
        REFERENCES "patron"("idpatron")
        MATCH SIMPLE
        ON DELETE NO ACTION
        ON UPDATE NO ACTION
        NOT DEFERRABLE,
    CONSTRAINT "fktablasproy" FOREIGN KEY ("idtablaproy")
        REFERENCES "tablasproy"("idtablaproy")
        MATCH SIMPLE
        ON DELETE NO ACTION
        ON UPDATE NO ACTION
        NOT DEFERRABLE
)
WITH (
    OIDS = False
);

ALTER TABLE "dtlletblaproy" OWNER TO "postgres";

COMMENT ON TABLE "dtlletblaproy" IS 'Detalle de atributos por tabla de proyecto';
COMMENT ON COLUMN "dtlletblaproy"."iddtlletblaproy" IS 'Identificador del detalle de tabla para proyecto';
COMMENT ON COLUMN "dtlletblaproy"."idtablaproy" IS 'Identificador de la tabla dentro del proyecto';
COMMENT ON COLUMN "dtlletblaproy"."idpatron" IS 'Identificación del patrón asociado al atributo';
COMMENT ON COLUMN "dtlletblaproy"."atribtablaproy" IS 'Nombre del atributo de la tabla';

```

4.2.1.6 Conjunto de tablas de una base de datos de proyecto

Tabla 10. Descripción Tabla: tablasproy

Columna	Tipo de Dato	Clave Principal	No Nulo	Descripción
idtablaproy	Int4	✓	✓	Identificador de la tabla
idproyecto	Int4	-	-	Identificador del proyecto al que pertenece la tabla
dsctablaproy	Varchar	-	-	Nombre de la tabla
nrregstablproy	Int4	-	-	Número de registros a generar para la tabla correspondiente
sccriptproyecto	Varchar	-	-	Script para la generación de datos.
Referencia				
Tabla Padre			Enlace	
proyecto			idproyecto=idproyecto	

Figura 13. Script para la creación de la Tabla: tablasproy

```

CREATE TABLE "tablasproy" (
    "idtablaproy" int4 NOT NULL,
    "idproyecto" int4,
    "dsctablaproy" varchar,
    "nrregstablproy" int4 DEFAULT 0,
    CONSTRAINT "pktablaproy" PRIMARY KEY("idtablaproy"),
    CONSTRAINT "fkproyecto" FOREIGN KEY ("idproyecto")
        REFERENCES "proyecto"("idproyecto")
    MATCH SIMPLE
    ON DELETE NO ACTION
    ON UPDATE NO ACTION
    NOT DEFERRABLE
)
WITH (
    OIDS = False
);

ALTER TABLE "tablasproy" OWNER TO "postgres";

COMMENT ON TABLE "tablasproy" IS 'Conjunto de tablas de una base de datos de proyecto';
COMMENT ON COLUMN "tablasproy"."idtablaproy" IS 'Identificador de la tabla';
COMMENT ON COLUMN "tablasproy"."idproyecto" IS 'Identificador del proyecto al que pertenece la tabla';
COMMENT ON COLUMN "tablasproy"."dsctablaproy" IS 'Nombre de la tabla';
COMMENT ON COLUMN "tablasproy"."nrregstablproy" IS 'Número de registros a generar para esta tabla';

```

4.2.2 Estructura funcional del proyecto. Acorde al planteamiento del proyecto y a la ejecución realizada, a continuación se estructuran las funcionalidades iniciales del software.

4.2.2.1 Diseño de ventanas. El diseño de ventana se basa en el diseño general de Windows.

4.2.2.1.1 Ventana principal. La ventana principal es una interfaz de múltiples documentos. Es un contenedor de documentos, en este caso son ventanas de datos.

Figura 14. Ventana Principal

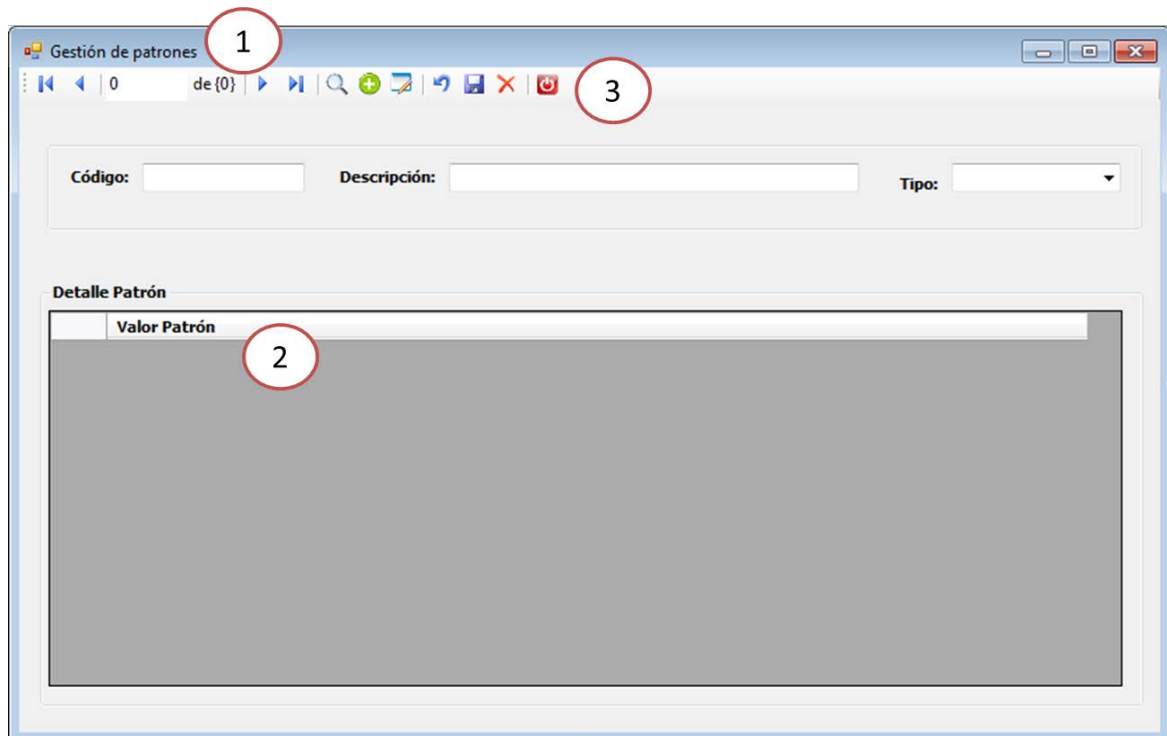


Las siguientes son las funcionalidades y características de la ventana principal:

1. **Menú de control:** Menú que ofrece las mismas funciones de los botones de control.
2. **Título de ventana:** Texto descriptivo de la funcionalidad de la ventana.
3. **Botones de control:** Botones que ofrecen la funcionalidad de minimizar, maximizar y cerrar la ventana actual.
4. **Menú principal:** Menú de la parte superior de la ventana superior que contiene acceso a las principales funcionalidades del sistema.
5. **Barra de íconos principal:** Presenta botones de acceso a las funcionalidades más importantes o de más frecuencia de uso de la aplicación.
6. **Área de datos:** Corresponde al área de la ventana principal donde se mostrarán las ventanas de datos contenidas en la ventana principal.

4.2.2.1.2 Ventana de datos. La ventana de datos es el área donde el usuario gestionará los datos correspondientes a una entidad o relación.

Figura 15. Ventana de Datos

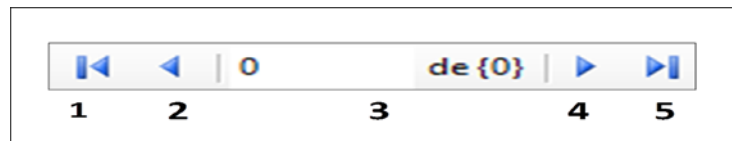


Las siguientes son las funcionalidades y características de la ventana principal:

- 1. Título de ventana:** Texto descriptivo de la funcionalidad de la ventana.
- 2. Área de datos:** Corresponde al área de la ventana principal donde se mostrarán las ventanas de datos contenidas en la ventana principal.
- 3. Barra de íconos de navegación:** Presenta botones de navegación y gestión de registros dentro de una ventana de datos.

1. **Íconos de navegación:** Los íconos de navegación permiten desplazarse a destinos de registros.

Figura 16. Íconos de Navegación



- (1) Ir al primer registro
- (2) Ir al anterior registro respecto al actual
- (3) Ir al registro de la posición dada
- (4) Ir al siguiente registro respecto al actual
- (5) Ir al último registro

2. **Íconos de gestión de registros:** Los íconos de gestión de registros permiten realizar funcionalidades sobre los registros.

Figura 17. Íconos de gestión de registros.



- (1) Buscar un registro
- (2) Adicionar registro
- (3) Editar registro actual
- (4) Deshacer cambios
- (5) Guardar cambios
- (6) Eliminar registro actual
- (7) Salir de ventana de datos actual

4.3 ESTRUCTURA FUNCIONAL Y RECORRIDO DEL SISTEMA

De acuerdo al planteamiento del proyecto y al desarrollo de la metodología SCRUM, la estructura funcional del software se realizó satisfaciendo las necesidades del cliente. A continuación la estructura funcional y el recorrido del sistema.

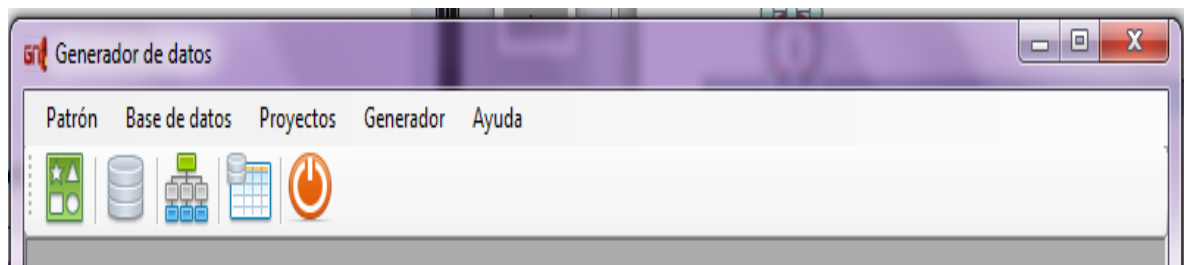
4.3.1 Ingreso al sistema. Para ingresar al sistema, ubique en el escritorio el siguiente ícono:

Figura 18. Ícono de Ingreso al Sistema



Pulse dos veces sobre él con el botón izquierdo del mouse. Aparece inicialmente la ventana principal de la aplicación. En la parte superior se aprecian en su orden: el título Generador de Datos, el menú principal y la barra principal de íconos.

Figura 19. Ventana Principal de la Aplicación



Desde el menú principal o desde la barra principal de íconos se puede acceder directamente a los cuatro módulos. Adicionalmente desde barra de íconos se puede finalizar la aplicación.

4.3.2 Módulo de patrones. Desde este módulo se gestionan los diferentes patrones y sus detalles. Gestionar efectuar

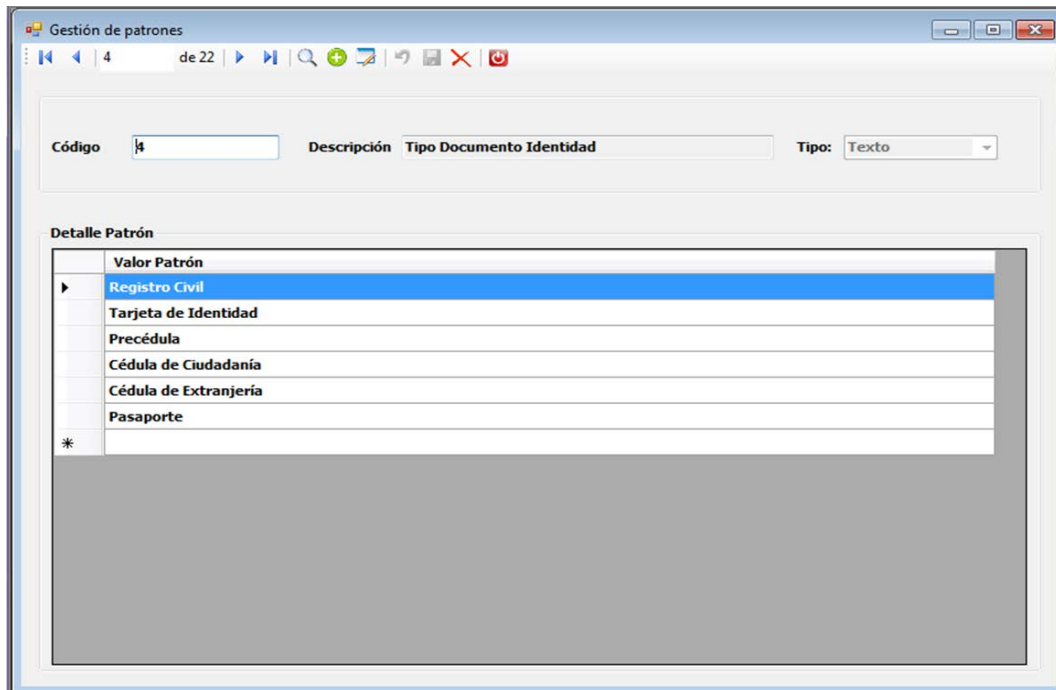
4.3.2.1 Acceso. Desde el menú principal pulse la opción Patrón o desde la barra de íconos pulse el primero.

Figura 20. Ingreso Módulo Patrones



4.3.2.2 Ventana de Gestión de Patrones. Desde esta ventana se puede adicionar, modificar o eliminar registros asociados a patrones. Adicionalmente es posible navegar a través de los registros existentes en la base de datos, o buscar un registro en particular.

Figura 21. Ventana Gestión de Patrones



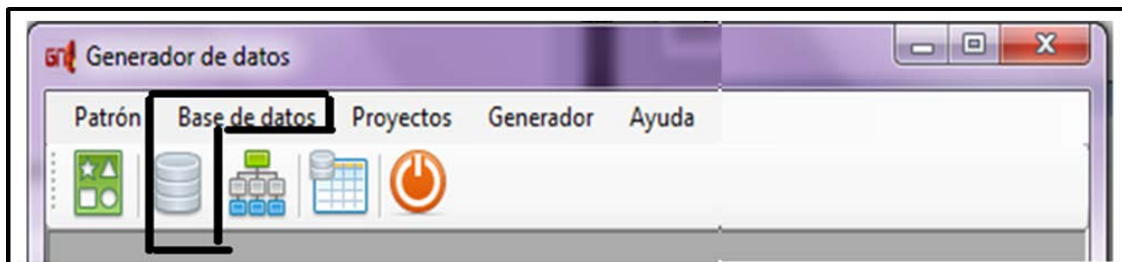
4.3.2.2.1 Descripción de campos.

- **Código:** Identificador único del patrón de datos.
- **Descripción:** Descripción breve de la aplicación del patrón.
- **Tipo:** Tipo de dato asociado al patrón de datos.
- **Detalle Patrón:** Banco de valores particulares para el patrón y que servirán de base para la generación de datos.

4.3.3 Módulo de base de datos. Desde este módulo se gestionan la información asociada a las bases de datos.

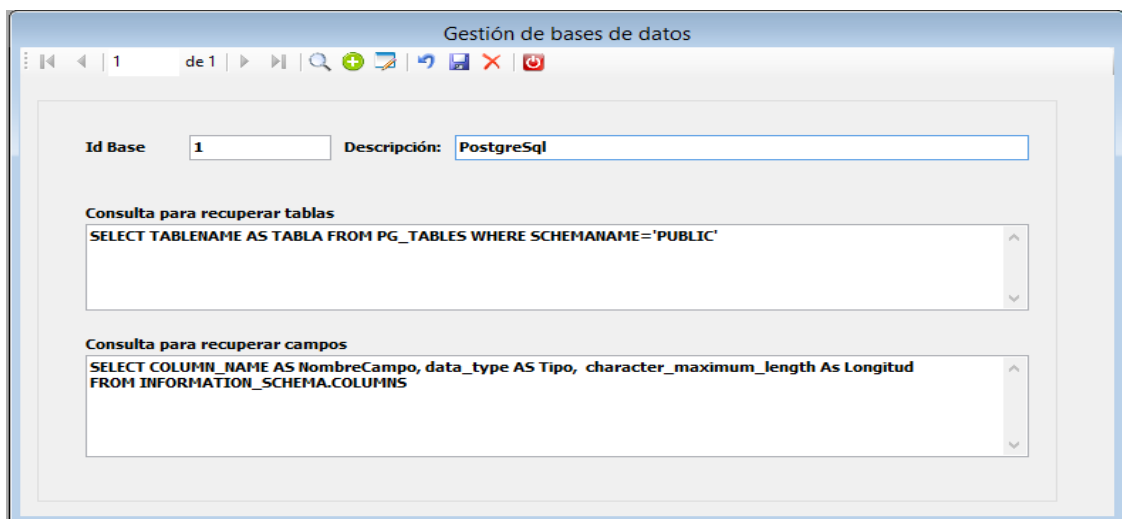
4.3.3.1 Acceso: Desde el menú principal pulse la opción Base de datos o desde la barra de íconos pulse el segundo.

Figura 22. Ingreso Módulo Base de Datos



4.3.3.2 Ventana de gestión de base de datos. Desde esta ventana se puede adicionar, modificar o eliminar registros asociados a Bases de Datos. Adicionalmente es posible navegar a través de los registros existentes en la base de datos, o buscar un registro en particular.

Figura 23. Ventana de Gestión de Base de Datos



4.3.3.2.1 Descripción de campos.

- **Id base:** Identificador único para la base de datos
- **Descripción:** Descripción breve de la base de datos
- **Consulta para recuperar tablas:** Sentencia SQL para recuperar tablas según la base de datos referida.
- **Consulta para recuperar atributos:** Sentencia SQL para recuperar atributos de tablas según la base de datos referida.

4.3.4 Módulo de proyectos. Desde este módulo se gestionan la información asociada a proyectos.

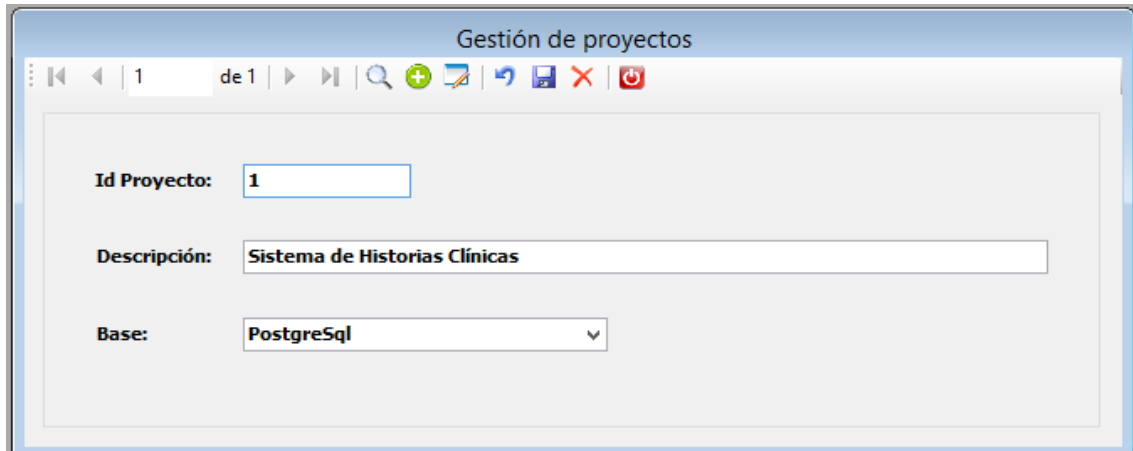
4.3.4.1 Acceso. Desde el menú principal pulse la opción Proyectos o desde la barra de íconos pulse el Tercero.

Figura 24. Ingreso Módulo de Proyectos



4.3.4.2 Ventana de gestión de proyectos. Desde esta ventana se puede adicionar, modificar o eliminar registros asociados a Proyectos. Adicionalmente es posible navegar a través de los registros existentes en la base de datos, o buscar un registro en particular.

Figura 25. Ventana de Gestión de Proyectos



The screenshot shows a window titled "Gestión de proyectos". At the top, there is a navigation bar with icons for back, forward, search, add, edit, delete, and refresh. Below the navigation bar, there is a form with three fields: "Id Proyecto" with the value "1", "Descripción" with the value "Sistema de Historias Clínicas", and "Base" with a dropdown menu showing "PostgreSql".

4.3.4.2.1 Descripción de campos:

- **Id Proyecto:** Identificador único para el proyecto
- **Descripción:** Descripción breve del proyecto
- **Base:** Base de datos utilizada por el proyecto.

4.3.5 Módulo de Generación de los scripts para las sentencias SQL para la alimentación de la Base de Datos.. Desde este módulo se generan las instrucciones SQL para cargar los datos de prueba al proyecto seleccionado.

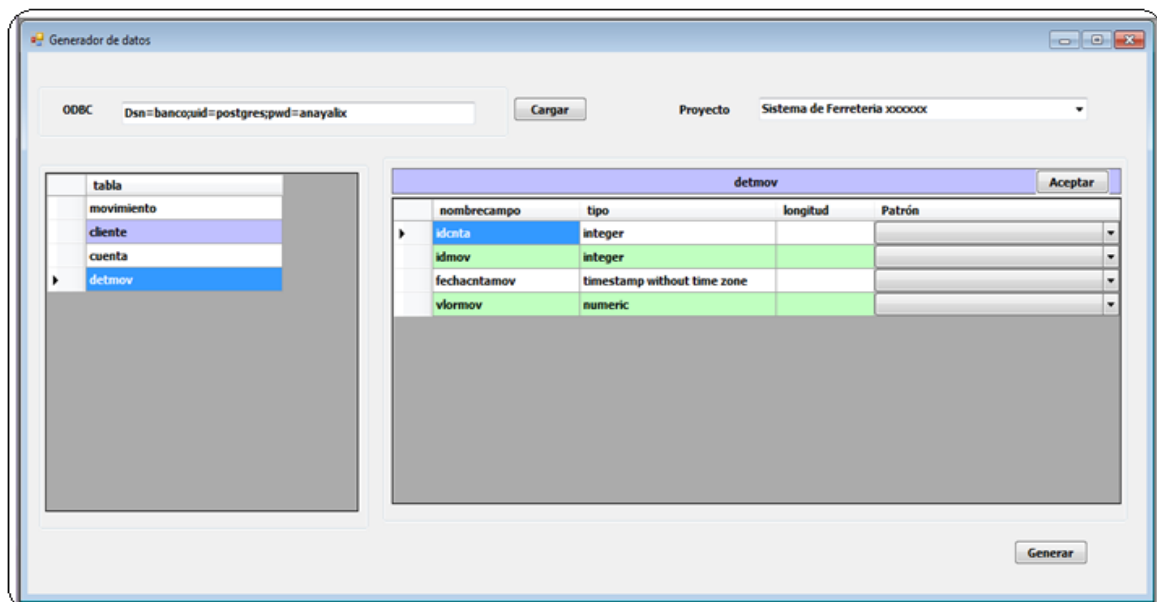
4.3.5.1 Acceso. Desde el menú principal pulse la opción **Generador** o desde la barra de íconos pulse el Cuarto.

Figura 26. Ingreso Módulo Generador



4.3.5.2 Ventana de generador de datos. Desde esta ventana se cumple con el objetivo principal de la herramienta: generar las sentencias SQL para alimentar la base del proyecto elegido con datos de prueba. Adicionalmente es posible navegar a través de los registros existentes en la base de datos, o buscar un registro en particular.

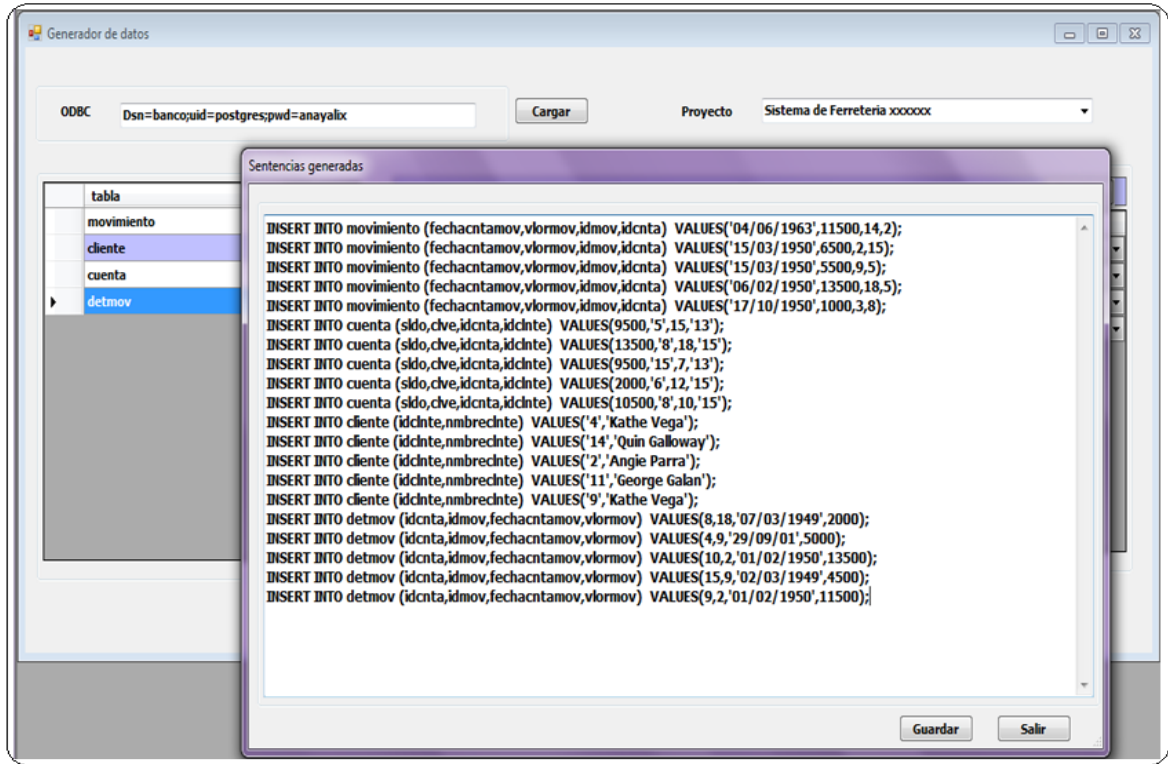
Figura 27. Ventana de Generador de datos



4.3.5.2.1 Descripción de campos

- **ODBC:** Cadena de conexión ODBC a la base de datos.
- **Proyecto:** Contiene la lista de proyectos registrados en la base de datos y para el cual se hará la generación.
- **Botón cargar:** Seleccionado el proyecto se debe pulsar este botón para cargar la lista de tablas que componen la base de datos del proyecto seleccionado.
- **Tablas:** Contiene la lista de tablas que componen la base de datos del proyecto.
- **Detalle de atributos:** Al pulsar dos veces sobre una tabla, en la ventana adjunta aparecen los atributos de cada tabla. Desde acá se asociará a cada atributo el patrón a utilizar.
- **Botón generar.** Genera las sentencias SQL correspondientes a los datos de prueba para el proyecto seleccionado. Al finalizar la generación se despliega una ventana mostrando las sentencias generadas. La siguiente figura muestra la generación de scripts para las sentencias SQL para la alimentación de la Base de Datos.

Figura 28. Generación de Sentencias



4.3.6 Ayuda de la aplicación

Desde este ícono el usuario podrá verificar toda la ayuda de la herramienta.

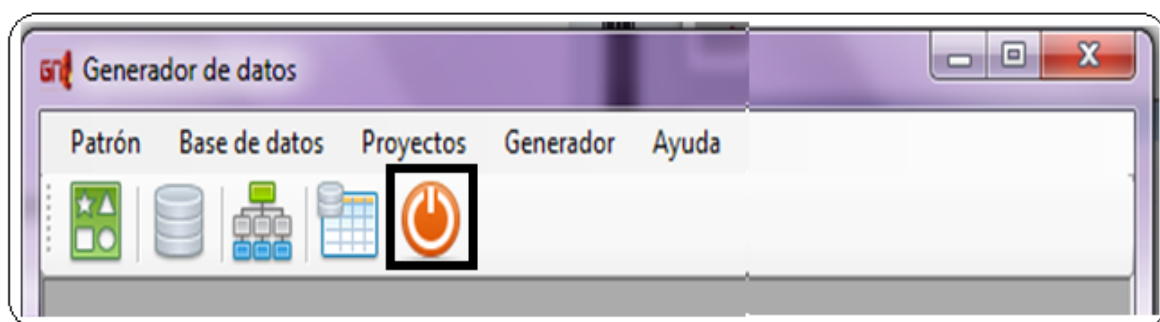
Figura 29. Ícono de Ayuda



4.3.7 Salir de la aplicación.

Desde la barra de íconos seleccione el último ícono.

Figura 30. Botón Salir de la Aplicación



5. EL LENGUAJE SQL: SENTENCIA PARA LA ADICIÓN DE DATOS A LA BASE DE DATOS

SQL, el Lenguaje de Consulta Estructurado, es un lenguaje de base de datos normalizado, usado por los diferentes gestores de base de datos con el objetivo de realizar determinadas operaciones sobre los datos o la estructura de los mismos.

SQL es una herramienta cuyas funciones son organizar, gestionar y recuperar datos almacenados en una base de datos con el fin de realizar tareas de datos básicas, tales como agregar, modificar y eliminar, así como de realizar operaciones no básicas procesando los datos de manera muy útil.

El lenguaje SQL es un lenguaje el cual está compuesto por comandos, cláusulas, operadores y funciones de agregado. Dichos elementos se combinan en las instrucciones permitiendo crear, actualizar y manipular las bases de datos.

Los comandos SQL satisfacen los requerimientos de un lenguaje de datos ya que sus funciones ejecutan objetivos particulares: Los comandos DDL significan Lenguaje de Definición de Datos permiten crear y definir nuevas bases de datos como tablas, vistas, campos y elementos; los comandos DML que en español sus siglas traducen a Lenguaje de Manipulación de Datos permiten generar consultas de selección, modificación, filtración, extracción y demás operaciones en la base de datos y los comandos DCL (Lenguaje de Control de Datos), permiten controlar las funciones de administración que realiza el DBMS, como la atomicidad y la seguridad de los datos. Las cláusulas son condiciones de modificación para definir los datos que se desean seleccionar o manipular. En SQL encontramos los operadores lógicos y de comparación los cuales evalúan las condiciones con el

objetivo de devolver un valor y las funciones de agregado que se usan en la mayoría de los casos para devolver un valor único en grupo de registros.

SQL es un lenguaje universal, se utiliza en cualquier sistema gestor de base de datos relacional, ya sea PostgreSQL, MySQL, SQL Server, Oracle y muchos más, cada uno maneja un estándar definido a su propia versión.

5.1 ADICIÓN DE DATOS A LA BASE DE DATOS

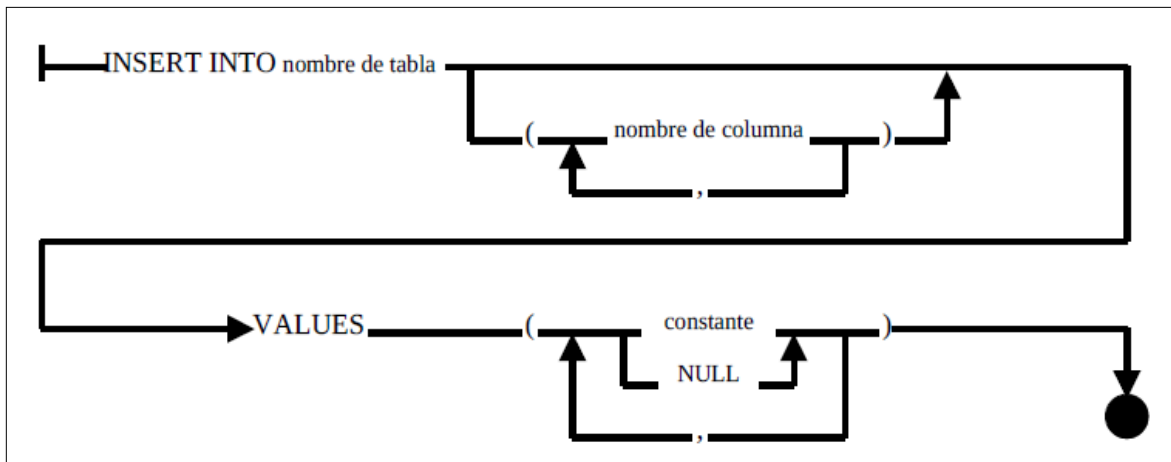
El lenguaje de manipulación de datos (DML), son un grupo de sentencias SQL con la finalidad de manipular los datos que se están almacenando en la base de datos, a nivel de filas y columnas. Estas sentencias permiten según se requiera que los datos se puedan modificar, eliminar, consultar o adicionar nuevas filas a las tablas de las bases de datos.

La sentencia INSERT, es la encargada de agregar nuevas filas de datos a una tabla. SQL proporciona tres maneras de adicionar nuevas filas de datos a una base de datos:

1. Una sentencia INSERT de una fila que agrega una única nueva fila de datos a una tabla.
2. Una sentencia INSERT multifila, encargada de extraer filas de datos de otra parte en la base de datos y agregarla a una tabla.
3. Una utilidad de carga masiva, agrega datos a una tabla o a varias tablas desde un fichero externo a la base de datos.

5.1.1 La sentencia INSERT de una fila. La sentencia INSERT de una fila agrega una nueva fila a una tabla; junto a la cláusula INTO especifican el destino de los datos y la cláusula VALUE especifica los valores de datos que la nueva fila contendrá. La siguiente figura explica el diagrama sintáctico de la sentencia INSERT de una fila.

Figura 31. Diagrama sintáctico de la sentencia INSERT de una fila.



Fuente: El Lenguaje SQL II: Introducción, modificación y borrado de datos. [En línea] Disponible en internet: <http://informatica.uv.es/estguia/ATD/apuntes/teoria/documentos/SQL-II.pdf>[Consultado 21 de enero de 2014].

Un ejemplo, es la actualización de un nuevo estudiante a la base de datos, el cual ingresa a la carrera de ingeniería de sistemas:

Nombre:	Santiago Alvarado Jaimes
Tipo de Documento:	CC
Número de Identificación:	1098765465
Edad:	19
Código:	2152345
Carrera:	Ingeniería de Sistemas
Fecha inicio:	22 de Enero de 2015
Valor Semestre:	189.876

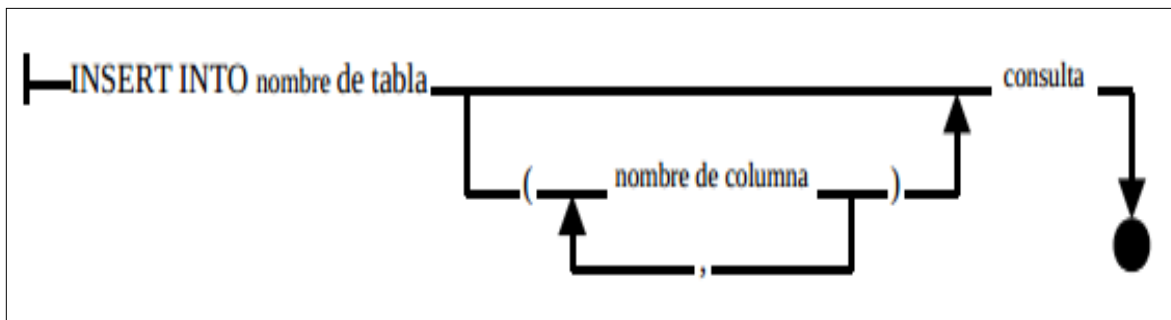
La sentencia INSERT para agregar el nuevo estudiante a la base de datos es:

```
INSERT INTO estudiante (nombre, tipoID, numID, edad, codest, nomCarrera,
feclng, valSem) VALUES ('Santiago Alvarado Jaimes', 'CC', 1098765465, 19,
2152345, 'Ingeniería de Sistemas', '22/01/2015', 189876)
```

5.1.2 La sentencia INSERT multifila. La siguiente forma de la sentencia INSERT permite agregar múltiples registros de datos a una tabla destino. Se debe realizar la combinación de las sentencia INSERT junto a la sentencia SELECT. El resultado la adición de registros cuyos valores provienen de la misma base de datos insertados mediante una consulta.

La siguiente figura explica el diagrama sintáctico de la sentencia INSERT para el segundo caso multifila.

Figura 32. Diagrama sintáctico de la sentencia INSERT multifila.



Fuente: El Lenguaje SQL II: Introducción, modificación y borrado de datos. [En línea] Disponible en internet: <http://informatica.uv.es/estguia/ATD/apuntes/teoria/documentos/SQL-II.pdf>[Consultado 21 de enero de 2014].

El estándar SQL especifica las siguientes normas para poder utilizar la inserción múltiple de filas sobre la consulta dentro de la sentencia INSERT:

- La cláusula ORDER BY no puede estar contenida en la consulta.
- La consulta no debe ser la UNIÓN de varias sentencias SELECT diferentes.
- La tabla destino de la sentencia INSERT no puede aparecer en la cláusula FROM ni en la consulta ni en ninguna subconsulta que ésta contenga.
- La lista de campos de las sentencias INSERT y SELECT deben coincidir en número de columnas y tipo de datos.
- Ninguna de las filas devueltas en la consulta debe infringir las reglas de integridad de la tabla en la que va a ocurrir la inserción.

Un ejemplo en este caso, siguiendo el ejemplo del estudiante es cuando se desea crear una nueva tabla con datos de estudiantes que hayan iniciado sus estudios antes del 31 de diciembre de 2013. Se crea una tabla llamada ANTESTUDIANTE2013, copiamos la información de las columnas que describan la información del estudiante, recordando que deben coincidir el número de columnas invocadas tanto en INSERT como en el SELECT, es decir, esta sería la sintaxis:

```
INSERT INTO antestudiante (nombre, tipoID, numID, edad, codest, nomCarrera,
    feclng, valSem) SELECT nombre, tipoID, numID, edad, codest, nomCarrera,
    feclng, valSem FROM estudiante WHERE feclng<31/12/13
```

5.1.3 Utilidades de Carga Masiva. Los datos a insertar en una base de datos provienen con mucha frecuencia de otros sistemas informáticos o son recolectados de otros lugares y almacenados en un archivo secuencial.

Los productos DBMS comerciales incluyen una capacidad de carga masiva que carga los datos desde un archivo a una tabla a alta velocidad. Sin embargo, pueden realizarse habilitaciones o inhabilitaciones de restricciones en estos productos para gestionar de manera satisfactoria la carga masiva de datos.

6. CONCLUSIONES

- La importancia de garantizar la relación entre las entidades radica en la forma correcta de utilizar la integridad referencial; esta asegura que la relación entre dos o más tablas permanezca sincronizada durante las operaciones de actualización y eliminación. La clave foránea es la encargada de que ocurra la integridad referencial, ya que ésta representa una referencia del registro que contiene el mismo valor en su clave primaria. Los datos en los campos de la clave foránea y la clave primaria deben coincidir, aunque los nombres de los campos puedan que nos sean los mismos.
- Un buen diseño de base de datos contribuye a información confiable, consistente y precisa. Al final la empresa gestionará su información de una forma más adecuada y esto se reflejará prácticamente en todas sus operaciones.
- SCRUM es una metodología que permite desarrollar progresivamente un producto software, aboliendo las potenciales dificultades que se presenten en la integración de funcionalidades; permitiendo la flexibilidad a los posibles cambios con el propósito de lograr la satisfacción del cliente. Sin embargo, para aplicarla, es necesario tener un buen conocimiento del producto que se desea obtener, dominio de la aplicación donde se va a desarrollar la herramienta, y un gran manejo en las situaciones donde se presenten complicaciones para avanzar.
- Gran parte del éxito de la metodología SCRUM es la participación del propietario del producto o Product Owner, es la persona encargada de la dirección y control del Product Backlog, es decir, la descripción de las historias

de usuario que debe cumplir el sistema. Su colaboración elimina posibles confusiones, su objetividad ayuda a marcar las prioridades del proyecto y a proporcionar detalles importantes en las diferentes evoluciones que se desarrollen en cada uno de los Sprint.

- Es importante resaltar el aprendizaje obtenido durante esta experiencia, ya que afianzó muchas de las conocimientos brindados por los docentes de la Escuela de Ingeniería de Sistemas durante mi carrera, ya que me permitió extraer información importante para el levantamiento de requisitos y detección de necesidades, adquirir habilidades para proponer soluciones, llevar una buena documentación para facilitar el diseño y la implementación y ratifico la importancia de desarrollar habilidades comunicativas efectivas para el buen desarrollo del producto.

7. RECOMENDACIONES

- El proyecto desarrollado corresponde a la primera versión de la herramienta funcionando correctamente acorde a los objetivos inicialmente requeridos. Se recomienda continuar el desarrollo de la herramienta con funcionalidades adicionales como: generación de los datos directamente sobre la base de datos, visualización parcial del diagrama de relaciones para ofrecer una mayor claridad de la integridad de los datos, creación de rango de valores y una funcionalidad para determinar la relación entre departamentos y ciudades.
- Se recomienda para una próxima versión el programa sea implementado en la web, con un administrador el cual mantenga actualizada la base de datos con datos prueba suministrado por desarrolladores y de esta manera mantener un buen volumen de datos.
- Garantizar la calidad de un producto software, implica establecer sistemas de evaluación y medición; los cuales determinen en qué medida se está actuando correctamente, documentar los procesos de desarrollo mediante una metodología existente o personal, estandarizar los procesos para proporcionar mejores resultados en la actividades; entrevistar cuidadosamente a los usuarios verificando que se ha comprendido las necesidades que se desean abordar y constantemente realizar pruebas que comprueben el buen funcionamiento del producto. Se recomienda a los estudiantes de Ingeniería de Sistemas, aumentar el nivel educativo y adquirir experiencias profesionales para lograr el éxito en la producción de software demostrando su calidad.
- Es importante durante la formación de pregrado prestarle más atención al inglés, tanto como contenido curricular como uso cotidiano. Hoy en día, el

inglés, es parte fundamental en la vida de un desarrollador de software ya que la mayoría de los lenguajes de programación usan en su sintaxis palabras en este idioma; sus manuales de referencia y gran cantidad de recursos están en inglés.

- El internet es un gran repositorio de información. Actualmente millones de personas buscan información por este medio. Obtener información que resulte significativa, útil, interesante y confiable depende de hacer uso de unos buenos criterios de búsqueda y de evaluación; los cuales ayudarán a definir la confiabilidad y validez de la información encontrada. Se recomienda evaluar los documentos en la red en su exactitud, autoría, objetividad, actualidad y cubrimiento con el objetivo de encontrar información valiosa y realizar buenas investigaciones.
- El desarrollador debe indagar sobre las sentencias para la recuperación de tablas y de datos según el motor de base de datos en el que esté realizando su producto software. Por ejemplo, para PostgreSQL funciona de la siguiente manera:

- Recuperación Tabla:

```
select tablename AS Tabla from pg_tables where schemaname='public'
```

- Recuperación Campos:

```
SELECT COLUMN_NAME AS NombreCampo, data_type AS Tipo,  
character_maximum_length As Longitud FROM  
INFORMATION_SCHEMA.COLUMNS WHERE TABLE_NAME
```

BIBLIOGRAFIA

ALFONZO, Pedro L. MARIÑO, Sonia I. y GODOY, María V. Propuesta de aplicación de SCRUM para gestionar el proceso de mantenimiento software: estudio preliminar. {En línea}. {21 de Noviembre de 2014}. Disponible en: (<http://www.cyta.com.ar/ta1101/v11n1a4.htm>)

CEBALLOS, Francisco Javier. Enciclopedia de Microsoft Visual C#. México: AlfaOmega Grupo Editor S.A de C.V, 2006. 784p.

DEITEL, Harvey. DEITEL Paul. Cómo programar en C#. Cuarta Edición. Pearson Educación de México, S.A de C,V, 2007. 1214p.

EL LENGUAJE SQL II: Introducción, modificación y borrado de datos. {En línea} {21 de enero de 2014}. Disponible en internet: (<http://informatica.uv.es/estguia/ATD/apuntes/teoria/documentos/SQL-II.pdf>).

KROENKE. David M. Procesamiento de Bases de Datos. Fundamentos, diseño e implementación. Octava Edición. México: Pearson Educación de México S.A de C.V, 2003. 689p.

MARQUÉS, Mercedes. Bases de Datos. {En línea}. {12 de Noviembre de 2014}. Disponible en: (<http://www.uji.es/bin/publ/edicions/bdatos.pdf>).

POSTGRESQL. PostgreSQL-es: Portal en español sobre PostgreSQL. {En línea} {14 de Noviembre de 2014}. Disponible en: (<http://www.postgresql.org.es>).

RODRÍGUEZ GONZALEZ, Pilar. Estudio de la aplicación de metodologías ágiles para la evolución de productos software. Madrid. 2008. 143p. Tesis de Máster. (Máster en tecnologías de la información). Universidad Politécnica de Madrid. Facultad de Informática.