



UNIVERSIDAD TÉCNICA ESTATAL DE QUEVEDO

FACULTAD DE CIENCIAS DE LA INGENIERÍA

ESCUELA DE INFORMÁTICA

Tema de tesis:

**“CREACIÓN DE COMPONENTES DATASETS PARA NETBEANS
IDE”**

Previo a la obtención del título de

Ingeniero en Sistemas

Autor:

Daniel Andrés De Los Santos Villao

Director de tesis:

Ing. Gleiston Cicerón Guerrero Ulloa

Quevedo – Ecuador

2014

DECLARACIÓN DE AUTORÍA Y CESIÓN DE DERECHOS

Yo, **DANIEL ANDRÉS DE LOS SANTOS VILLAO**, declaro que el trabajo aquí descrito es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento. **“CREACIÓN DE COMPONENTES DATASETS PARA NETBEANS IDE”** es de mi exclusiva responsabilidad.

La Universidad Técnica Estatal de Quevedo, puede hacer uso de los derechos correspondientes a este trabajo, según lo establecido por la Ley de Propiedad Intelectual, por su Reglamento y por la normatividad institucional vigente.

Daniel Andrés De los Santos Villao
C.I. 092691871-5

CERTIFICACIÓN DEL DIRECTOR DE TESIS

El suscrito, **ING. GLEISTON GUERRERO ULLOA**, Docente de la Universidad Técnica Estatal de Quevedo, certifica que el Egresado **DANIEL ANDRÉS DE LOS SANTOS VILLOA**, realizó la tesis de grado previo a la obtención del título de Ingeniero en Sistemas titulada “**CREACIÓN DE COMPONENTES DATASETS PARA NETBEANS IDE**”, bajo mi dirección, habiendo cumplido con las disposiciones reglamentarias establecidas para el efecto.

Ing. Gleiston Guerrero Ulloa
Director de Tesis

CERTIFICACIÓN DE REDACCIÓN

Yo, **LCDA. MARJORI TORRES BOLAÑOS** con C.I. **070182756-0**, Docente de la Facultad de Ciencias de la Ingeniería de la Universidad Técnica Estatal de Quevedo, certifico que he revisado la tesis de grado del Egresado **DANIEL ANDRÉS DE LOS SANTOS VILLO** con C.I. **092691871-5** previo a la obtención del título de Ingeniero en Sistemas, titulada “**CREACIÓN DE COMPONENTES DATASETS PARA NETBEANS IDE**”, habiendo cumplido con la redacción y corrección ortográfica que se ha indicado.

Lcda. Marjori Torres Bolaños
C.I. 070182756-0



UNIVERSIDAD TÉCNICA ESTATAL DE QUEVEDO

FACULTAD DE CIENCIAS DE LA INGENIERÍA

CARRERA DE INGENIERÍA EN SISTEMAS

TRIBUNAL DE TESIS

Presentado al Consejo Directivo como requisito previo a la obtención del título de Ingeniero en Sistemas.

Aprobado:

Dr. PHD Amilkar Puris
Presidente del tribunal de tesis

Ing. Nancy Rodríguez
Miembro del Tribunal de tesis

Ing. Cristian Zambrano
Miembro del Tribunal de tesis

Quevedo - Los Ríos – Ecuador

2014

AGRADECIMIENTO

Agradezco primeramente a Dios el Dador de la vida, quién me ha concedido de su sabiduría e inteligencia para lograr mis metas y aspiraciones.

A mis padres, por demostrarme su apoyo incondicional en todas las áreas de mi vida.

A mis maestros, por impartir sus conocimientos y valores profesionales.

Y a mis compañeros de clase y amigos, quienes me han regalado su afecto y confianza en toda la etapa universitaria.

Daniel

DEDICATORIA

A toda mi familia, especialmente a mis padres, en quienes siempre encontré palabras de superación, ánimo, e ímpetu para hacer frente a los obstáculos que se han presentado a lo largo de mi vida.

A la nueva generación estudiantil que vive en busca de éxitos para el progreso y bienestar de nuestro país.

Daniel

CÓDIGO DUBLIN

Título/Title	“CREACIÓN DE COMPONENTES DATASETS PARA NETBEANS IDE”
Creador/Creator	Daniel Andrés De los Santos Villao
Materia/Subject	Base de Datos, Diseño y Análisis de Sistemas
Descripción/Description	El presente trabajo consiste en la creación de un asistente gráfico que facilite el manejo de datos al momento de desarrollar aplicaciones con conexión a base de datos en Java usando Netbeans IDE.
Editor/Publisher	FCI; Carrera de Ingeniería en Sistemas, Daniel De los Santos Villao
Colaborador/Contributor	NINGUNO
Fecha/Date	Viernes, 21 de Marzo del 2014
Tipo/Type	PROYECTO DE INVESTIGACIÓN
Formato/Format	Microsoft Office Word 2010

ÍNDICE

DECLARACIÓN DE AUTORÍA Y CESIÓN DE DERECHOS	ii
CERTIFICACIÓN DEL DIRECTOR DE TESIS	iii
CERTIFICACIÓN DE REDACCIÓN.....	iv
TRIBUNAL DE TESIS.....	v
AGRADECIMIENTO	vi
DEDICATORIA	vii
CÓDIGO DUBLIN	viii
ÍNDICE.....	ix
INDICE DE TABLAS	xiii
INDICE DE ILUSTRACIONES	xv
RESUMEN EJECUTIVO.....	xvii
ABSTRACT.....	xviii
CAPITULO I.....	1
MARCO CONTEXTUAL DE LA INVESTIGACIÓN	1
1.1. INTRODUCCIÓN.....	2
1.2. PROBLEMATIZACIÓN	3
1.2.1. Diagnóstico	3
1.2.2. Formulación	4
1.2.3. Sistematización.....	4
1.3. JUSTIFICACIÓN.....	5
1.4. OBJETIVOS.....	6
1.4.1. Objetivo General	6
1.4.2. Objetivos Específicos.....	6
1.5. HIPÓTESIS.....	6
1.5.1. Planteamiento	6
1.5.2. Matriz de Operacionalización.....	7

CAPITULO II.....	8
MARCO TEÓRICO DE LA INVESTIGACIÓN.....	8
2.1. FUNDAMENTACIÓN CONCEPTUAL.....	9
2.1.1. IDE (Entorno de Desarrollo Integrado)	9
2.1.2. DataSet (concepto)	9
2.1.3. Tiempo de desarrollo	9
2.1.4. Plataforma	9
2.1.5. API (Interfaz de Programación de Aplicaciones)	10
2.1.6. Modularidad	10
2.1.7. Plugin	11
2.1.7. Método Deductivo	11
2.1.8. Método Inductivo	11
2.2. FUNDAMENTACIÓN TEÓRICA	12
2.2.1. Módulos en Netbeans	12
2.2.2. Extreme Programming XP	16
2.2.3. Documentos XML	19
2.3. MARCO REFERENCIAL.....	21
2.3.1. DataSet de .NET	21
2.3.2. Hibernate	24
2.3.3. Proyecto AppPaint	26
2.4. FUNDAMENTACIÓN LEGAL.....	27
CAPITULO III.....	28
METODOLOGÍA DE LA INVESTIGACIÓN.....	28
3.1. MATERIALES Y METODOS.....	29
3.1.1. Materiales y presupuesto	29
3.1.2. Factibilidad.....	29
3.1.3. Métodos de investigación.....	30

3.1.4. Metodología de desarrollo.....	31
3.2. TIPO DE INVESTIGACIÓN.....	53
3.2.1. Investigación de Campo.....	53
3.2.2. Recolección de datos.....	53
3.2.3. Presentación de datos	57
3.3. DISEÑO DE LA INVESTIGACIÓN.....	66
3.3.1. Diseño cuasi-experimental.....	67
3.3.2. Pasos del diseño cuasi-experimental	67
3.4. POBLACIÓN Y MUESTRA	70
3.4.1. Población finita.....	70
3.4.2. Cálculo de la muestra	71
CAPITULO IV.....	73
RESULTADOS Y DISCUSIÓN	73
4.1 RESULTADOS.....	74
4.1.1. Pruebas de software	74
4.1.2. Análisis general de las encuestas.....	77
4.1.3. Análisis del cuasi-experimento.....	80
4.1.4. Comprobación de Hipótesis	83
4.1.5. Resultados de los objetivos planteados.....	84
4.2 DISCUSIÓN	86
4.2.1. Tema de la investigación	86
4.2.2. Requisitos de usuario no existentes	87
CAPITULO V.....	88
CONCLUSIONES Y RECOMENDACIONES	88
5.1 Conclusiones	89
5.2 Recomendaciones	90
CAPITULO VI.....	91

BIBLIOGRAFÍA.....	91
6.1. Literatura Citada	92
6.2. Linkografía	92
CAPITULO VII.....	93
ANEXOS.....	93
MANUAL DE USUARIO.....	94
TABLA T-STUDENT	138
FICHAS DE REGISTRO DEL CUASI-EXPERIMENTO	141
ENCUESTAS.....	150

INDICE DE TABLAS

TABLA 1.- MATRIZ DE OPERACIONALIZACIÓN.....	7
TABLA 2.- VALORES DEL XP.....	17
TABLA 3.- ACTIVIDADES DEL XP.....	18
TABLA 4.- MATERIALES Y PRESUPUESTO	29
TABLA 5.- NOMBRE DEL PROYECTO Y ARCHIVO NBM.....	33
TABLA 6.- HISTORIA DE USUARIO CONECTAR CON LA BD.....	34
TABLA 7.- HISTORIA DE USUARIO VISUALIZAR EL MODELO ESTRUCTURAL	34
TABLA 8.- HISTORIA DE USUARIO CREAR MÉTODOS CON SENTENCIAS SQL.....	35
TABLA 9.- HISTORIA DE USUARIO EJECUTAR MÉTODOS DENTRO DEL PROYECTO.....	36
TABLA 10.- ANALISIS DE PRIORIDADES Y ESTIMACIÓN	36
TABLA 11.- RESPUESTAS DE ENCUESTADOS DEL 4TO SEMESTRE (2DO AÑO).....	57
TABLA 12.- RESPUESTAS DE ENCUESTADOS DEL 6TO SEMESTRE (3ER AÑO).....	57
TABLA 13.- RESPUESTAS DE ENCUESTADOS DEL 8VO SEMESTRE (4TO AÑO).....	58
TABLA 14.- RESULTADOS DE LA PRIMERA PROPOSICIÓN DE LA ENCUESTA	59
TABLA 15.- RESULTADOS DE LA SEGUNDA PROPOSICIÓN DE LA ENCUESTA	60
TABLA 16.- RESULTADOS DE LA TERCERA PROPOSICIÓN DE LA ENCUESTA	61
TABLA 17.- RESULTADOS DE LA CUARTA PROPOSICIÓN DE LA ENCUESTA	62

TABLA 18.- RESULTADOS DE LA QUINTA PROPOSICIÓN DE LA ENCUESTA	63
TABLA 19.- RESULTADOS DE LA SEXTA PROPOSICIÓN DE LA ENCUESTA	64
TABLA 20.- RESULTADOS DE LA SÉPTIMA PROPOSICIÓN DE LA ENCUESTA	65
TABLA 21.- FICHA DE REGISTRO DE DATOS DEL DISEÑO CUASI-EXPERIMENTAL	68
TABLA 22.- POBLACIÓN FINITA CARRERA DE SISTEMAS.....	71
TABLA 23.- RESULTADOS GENERALES DE LAS ENCUESTAS.....	78
TABLA 24.- FICHA DE LOS RESULTADOS GENERALES DEL DISEÑO CUASI-EXPERIMENTAL	81

INDICE DE ILUSTRACIONES

ILUSTRACIÓN 1.- CUADRO DE DIÁLOGO DE MUESTRAS DE PROYECTOS NETBEANS MODULES.....	12
ILUSTRACIÓN 2.- PROYECTO NETBEANS MODULE	14
ILUSTRACIÓN 3.- ALCANCE DEL XP	16
ILUSTRACIÓN 4.- ESTRUCTURA DE UN DOCUMENTO XML	20
ILUSTRACIÓN 5.- DATAADAPTER Y DATASET DE .NET	22
ILUSTRACIÓN 6.- DISEÑADOR DEL DATASET DE .NET.....	23
ILUSTRACIÓN 7.- MAPEO DE HIBERNATE	24
ILUSTRACIÓN 8.- CATEGORÍA HIBERNATE EN NETBEANS IDE	25
ILUSTRACIÓN 9.- APLICACIÓN APPPAINT	26
ILUSTRACIÓN 10.- DIFERENCIAS ENTRE WINDOWS Y LINUX.....	27
ILUSTRACIÓN 11.- DIAGRAMA DINÁMICO DEL XP	32
ILUSTRACIÓN 12.- DIAGRAMA DE CLASES DEL PAQUETE TYPEFILE.....	38
ILUSTRACIÓN 13.- DIAGRAMA DE CLASES DEL PAQUETE DBSTRUCTURE	39
ILUSTRACIÓN 14.- DIAGRAMA DE CLASES DEL PAQUETE ACTIONS.....	40
ILUSTRACIÓN 15.- DIAGRAMA DE CLASES DEL PAQUETE WIZARDS	41
ILUSTRACIÓN 16.- DIAGRAMA DE CLASES DEL JDATASET-LIB.JAR	42
ILUSTRACIÓN 17.- DIAGRAMA DE SECUENCIA - CONECTAR CON LA BD	43
ILUSTRACIÓN 18.- DIAGRAMA DE SECUENCIA - VISUALIZAR EL DIAGRAMA DE LA BD	43
ILUSTRACIÓN 19.- DIAGRAMA DE SECUENCIA - CREAR MÉTODOS CON SENTENCIAS SQL.....	44
ILUSTRACIÓN 20.- DIAGRAMA DE SECUENCIA - EJECUTAR MÉTODOS DENTRO DEL PROYECTO JAVA.....	44
ILUSTRACIÓN 21.- ELEMENTO RAÍZ DEL ARCHIVO XML	45

ILUSTRACIÓN 22.- OTROS ELEMENTOS DEL ARCHIVO XML	46
ILUSTRACIÓN 23.- CUADRO DE DIÁLOGO NEW FILE	47
ILUSTRACIÓN 24.- FORMULARIO CONFIGURAR CONEXIÓN.....	47
ILUSTRACIÓN 25.- VENTANA EDITOR DEL JDATASET	48
ILUSTRACIÓN 26.- WIZARD PARA AGREGAR SENTENCIAS SQL	49
ILUSTRACIÓN 27.- FORMULARIO CONSTRUCTOR DE SENTENCIAS	50
ILUSTRACIÓN 28.- FORMULARIO DE REVISIÓN DE DATOS.....	51
ILUSTRACIÓN 29.- DIAGRAMA DE GANTT.....	52
ILUSTRACIÓN 30.- PORCENTAJES DE LA PRIMERA PROPOSICIÓN DE LA ENCUESTA	59
ILUSTRACIÓN 31.- PORCENTAJES DE LA SEGUNDA PROPOSICIÓN DE LA ENCUESTA	60
ILUSTRACIÓN 32.- PORCENTAJES DE LA TERCERA PROPOSICIÓN DE LA ENCUESTA	61
ILUSTRACIÓN 33.- PORCENTAJES DE LA CUARTA PROPOSICIÓN DE LA ENCUESTA	62
ILUSTRACIÓN 34.- PORCENTAJES DE LA QUINTA PROPOSICIÓN DE LA ENCUESTA	63
ILUSTRACIÓN 35.- PORCENTAJES DE LA SEXTA PROPOSICIÓN DE LA ENCUESTA	64
ILUSTRACIÓN 36.- PORCENTAJES DE LA SÉPTIMA PROPOSICIÓN DE LA ENCUESTA	65

RESUMEN EJECUTIVO

La investigación de tesis que se presenta a continuación, surge tras la observación evidente de las repetidas y monótonas tareas de escritura que el programador tiene que realizar para codificar los métodos que ejecutan las operaciones transaccionales hacia la base de datos, las cuales causan complejidad y lentitud en el desarrollo general de la aplicación. Mediante la consecución de encuestas ejercidas dentro de la comunidad informática de la UTEQ, se dan resultados que concluyen con la aprobación de la problemática expuesta, pero también se logra encontrar soluciones de acuerdo a las propuestas planteadas.

La hipótesis planteada define que el manejo de datos a través de un asistente gráfico de acceso a datos facilita el desarrollo de aplicaciones en Java con conexión a base de datos; de esa manera, se obtuvo como objetivo principal de la investigación crear un asistente gráfico de acceso a datos que facilite el procedimiento de manejo de datos al momento de desarrollar aplicaciones en Java usando Netbeans como IDE (entorno de desarrollo integrado).

La construcción de este asistente gráfico de acceso a datos se lleva a cabo mediante el desarrollo de un plugin para Netbeans, el cual se instale y permita extender nuevas funcionalidades de manejo de datos en el entorno de trabajo de Netbeans IDE; de esa manera el programador puede utilizar una interfaz gráfica que le permita abreviar el procedimiento para realizar los métodos de llamado y actualización de datos, e implemente un código fuente reducido; logrando aminorar la complejidad y lentitud de desarrollo, otorgando facilidad y agilidad por medio de componentes de asistencia gráfica para el programador.

Las conclusiones obtenidas tras evaluaciones y pruebas realizadas por programadores expertos e inexpertos dan un resultado favorable y permiten la comprobación de la hipótesis.

ABSTRACT

The thesis research that is presented below comes after the evident observation of repeated and monotonous writing tasks that the developer has to perform to encode methods that execute transactional operations to the database, which cause complexity and slowness in the overall development of the application. By means of achievement surveys exercised within the computing community of UTEQ, results were given that concluding with the approval of the issues exposed, but also manages to find solutions according to the proposals raised.

The hypothesis presented defines that the management of data through a graphical wizard of data access facilitates the development of Java applications with connection to database; thus, was obtained as the main objective of the research to create a graphical wizard of data access that facilitates the process of data management when developing applications in Java using Netbeans as IDE (Integrated Development Environment).

The construction of this graphical wizard of data access is accomplished through the development of a plugin for Netbeans, which installed and extend new functionalities of data management in the workplace Netbeans IDE; thus, the programmer can use a graphical interface that allows you to shorten the procedure for to perform methods to call and update of data, and implement a reduced source code; achieving lessen the complexity and slowness in development, providing facility and agility through components of graphic support for the programmer.

The conclusions obtained after evaluation and testing by experienced and inexperienced programmers give a favorable result according to the testing of the hypothesis.

CAPITULO I

MARCO CONTEXTUAL DE LA INVESTIGACIÓN

1.1. INTRODUCCIÓN

El presente trabajo consiste en la creación de un asistente gráfico que facilite el manejo de datos al momento de desarrollar aplicaciones en Java utilizando el entorno de trabajo de Netbeans IDE, este asistente gráfico se lo conoce más en el ambiente informático como DataSet o conjunto de datos.

Como lo especifica su nombre, un DataSet es un conjunto de objetos de datos que por lo general son transcritos o copiados desde una base de datos a un espacio de trabajo para la correspondiente manipulación y uso de los mismos. El componente DataSet permite que el programador visualice e interactúe con estos objetos con el fin de realizar operaciones transaccionales tales como indexar, ordenar, buscar, filtrar y modificar los datos, entre otras.

Una de las primeras interrogantes inciertas que surge en un programador de sistemas es el tiempo en que le tomará construir una aplicación, ya que la demora en el tiempo de construcción no sólo depende de los requerimientos de los clientes, sino también de las herramientas y métodos que se tengan disponibles para acelerar el desarrollo.

En la actualidad se han desarrollado muchas herramientas que favorecen la programación en Java que sustituyen las clases encargadas de las operaciones ligadas con la base de datos, tales herramientas son denominados como frameworks de persistencia de datos, que asisten al programador para que su aplicación contenga ficheros que configuren la conexión y operaciones hacia la base de datos, permitiendo constituir características robustas sobre la aplicación según los beneficios que la herramienta de persistencia plantee, aportando a un desarrollo ágil y seguro.

Estas herramientas de persistencia de datos generan su funcionalidad a base de ficheros de configuración que son agregados al proyecto en desarrollo, pero no establecen un ambiente gráfico en donde el programador pueda visualizar los objetos de la base de datos en el IDE en que se está trabajando.

En base a lo escrito previamente y estudiando todas las posibilidades de crear un asistente gráfico que facilite el manejo de datos en el desarrollo de aplicaciones de Java utilizando Netbeans como IDE, surge la motivación de realizar este trabajo de trabajo de tesis.

Este trabajo fue realizado por medio de la creación de un módulo, desarrollado sobre una aplicación de **Netbeans Platform**. La principal característica de Netbeans es la modularidad, lo cual origina su funcionalidad mediante la combinación de módulos pequeños, simples y de fácil uso que encapsulan funciones específicas para el desarrollo de las aplicaciones que ofrece este IDE.

De esta manera Netbeans permite desarrollar un nuevo módulo en el mismo IDE en la categoría **Netbeans Module**, integrarlo e instalarlo en una instancia de Netbeans IDE, estos módulos que son un tipo de proyecto diferente, son más conocidos en la comunidad informática como plugins.

Como ejemplo y modelo para realizar este trabajo se tomará el entorno de desarrollo de .NET que ya tiene integrado el objeto DataSet como un componente para desarrollar aplicaciones, con el fin de realizar una ingeniería inversa para comprender qué lo hace funcionar y como está fabricado, pues es en la plataforma de .NET donde se creó y se origina el uso del componente DataSet como una herramienta de soporte para el manejo de datos.

1.2. PROBLEMATIZACIÓN

1.2.1. Diagnóstico

Por lo general, para administrar las operaciones que se realizan hacia una base de datos al momento de desarrollar una aplicación informática en el lenguaje de Java, es necesario que el programador realice las clases que contengan los métodos para el llamado y la actualización de datos.

Al analizar el procedimiento que el programador tiene que realizar para el acceso a datos en el lenguaje de Java se obtiene la siguiente resolución: El

procedimiento para realizar las operaciones transaccionales hacia la base de datos a través de código fuente escrito línea por línea, provoca que el desarrollo de una aplicación con conexión a base de datos sea monótona, repetitiva y a la vez compleja.

Esto es evidente cuando el programador de software tiene que codificar métodos en donde sea necesario realizar consultas complejas a una base de datos, por ejemplo una consulta que invoque a varias tablas relacionadas. Para esto el programador tiene que acudir al generador de consultas del DBMS, escribir la sentencia, armarla bien hasta que genere el resultado esperado, y luego copiarla al programa que se está codificando en el IDE. Finalmente el programador tiene que encargarse de que el método pueda leer los datos invocados por medio de la sentencia establecida, y darles el correspondiente uso en la aplicación.

Todas estas tareas de codificación provocan que el programador implemente un código fuente excesivo, repetido, y a la vez provoque lentitud en la construcción de la aplicación.

1.2.2. Formulación

¿De qué manera mejorar el procedimiento de manejo de datos, al momento de desarrollar una aplicación en Java utilizando Netbeans como IDE?

1.2.3. Sistematización

¿Cómo armar y probar una sentencia compleja directamente desde el IDE, sin tener que acudir al generador de consultas del DBMS?

¿Cómo visualizar desde el IDE, el modelo estructural de los objetos y datos que contiene una base de datos?

¿Cómo hacer que un componente gráfico de manejo de datos, complemente el proceso para desarrollar aplicaciones en Java?

1.3. JUSTIFICACIÓN

Java es una herramienta potencial en el desarrollo de aplicaciones. Una de las características más relevantes que posee Java está en ejecutar sus aplicaciones en cualquier máquina virtual Java, sin importar la arquitectura de la computadora, ya que está diseñado para no tener ninguna dependencia de implementación como fuera posible; de esa manera Java ha alcanzado una gran demanda en su uso como herramienta de desarrollo de software.

En el ámbito de manejo y persistencia de datos, Java cuenta con varias herramientas como Hibernate, Ibatis, TopLink, entre otros, aún Java se ha encargado de estandarizar esta capa de persistencia a través de la creación de un estándar JPA (Java Persistence API) basado en el modelo ORM (Object Relational Mapping) que realiza un mapeo relacional entre los objetos de la aplicación y las tablas de la base de datos¹; sin embargo estas herramientas generan ficheros sin ninguna opción de visualización de los objetos de las base de datos.

Pues es de mucha motivación como usuario de Java y Netbeans, elaborar una aplicación gráfica que beneficie al programador o desarrollador de sistemas, para realizar el procedimiento de manipulación de datos con más sencillez teniendo una interfaz gráfica que le asistirá por medio de una secuencia de pasos para generar las operaciones ligadas a la base de datos, siendo este cómodo y eficiente para el desarrollo de aplicaciones.

Siendo Java un lenguaje de programación con mucha acogida, no existe un IDE que contenga este complemento gráfico, ni un objeto similar que proporcione la funcionalidad de los componentes DataSet. Poner a disposición de los usuarios de Java un DataSet fabricado bajo la plataforma de Netbeans beneficiaría a la comunidad informática que elabora aplicaciones de código abierto, ya que al integrar esta herramienta en la etapa de desarrollo de software reduciría el tiempo de construcción de una aplicación informática.

¹ Caules, C.Á. (2012). Arquitectura Java Sólida. Pag. 198. Sun Certified Enterprise Architecture

1.4. OBJETIVOS

1.4.1. Objetivo General

- ❖ Crear un asistente gráfico de acceso a datos que facilite el procedimiento de manejo de datos en tiempo de desarrollo de aplicaciones en Java utilizando Netbeans como IDE.

1.4.2. Objetivos Específicos

- ❖ Visualizar los objetos que contiene una base de datos directamente desde un editor gráfico de Netbeans IDE para su respectivo manejo.
- ❖ Abreviar el procedimiento de manejo de datos, por medio de una secuencia de pasos a seguir utilizando una interfaz gráfica.
- ❖ Complementar el desarrollo de aplicaciones, empleando este asistente gráfico como un componente esencial en un proyecto Java usando Netbeans IDE.

1.5. HIPÓTESIS

1.5.1. Planteamiento

Realizar el manejo de datos a través de un asistente gráfico de acceso a datos, facilita el desarrollo de aplicaciones con conexión a base de datos.

Variable Independiente

Asistente gráfico de acceso a datos

Variable Dependiente

Desarrollo de aplicaciones en Java con conexión a base de datos

1.5.2. Matriz de Operacionalización

TABLA 1.- MATRIZ DE OPERACIONALIZACIÓN

	VARIABLES	DEFINICIÓN	DIMENSIÓN	INDICADORES
VARIABLE INDEPENDIENTE	Asistente gráfico de acceso a datos	Se trata de un componente gráfico al servicio del programador, que abrevia la secuencia de pasos a seguir para realizar las operaciones ligadas a la base de datos.	Entorno de desarrollo de aplicaciones informáticas	Interacción con el usuario Comodidad y facilidad de uso Tiempo de desarrollo
VARIABLE DEPENDIENTE	Desarrollo de aplicaciones en Java con conexión a base de datos	Se refiere a la serie de métodos y operaciones que se realizan en el lenguaje de Java para realizar las conexiones y operaciones hacia la base de datos.	Desarrollo de aplicaciones informáticas	Seguridad en las operaciones internas de la aplicación Tiempo de desarrollo

CAPITULO II

MARCO TEÓRICO DE LA INVESTIGACIÓN

2.1. FUNDAMENTACIÓN CONCEPTUAL

A continuación los conceptos más relevantes a tomar en cuenta para comprender el desarrollo de la presente investigación de tesis.

2.1.1. IDE (Entorno de Desarrollo Integrado) ²

Un entorno de desarrollo integrado llamado también IDE por sus siglas en inglés (Integrated Development Environment), es un programa informático compuesto por un conjunto de herramientas de programación que han sido encapsulados para desarrollar aplicaciones informáticas; es decir, un IDE consta de un editor de código, un compilador, un depurador, y un constructor de interfaz gráfica (GUI). Un IDE puede disponer exclusivamente de un solo lenguaje de programación o bien puede disponerse para varios.

2.1.2. DataSet (concepto) ³

El DataSet es una representación de los objetos de la base de datos en un espacio de trabajo del IDE, con los que se puede realizar el manejo de los objetos independientemente de la base de datos que los contenga. El DataSet copia o transcribe un conjunto de objetos desde una base de datos a un espacio de trabajo.

2.1.3. Tiempo de desarrollo

Tiempo de desarrollo o modo de diseño, es el momento en que se codifica o se construye una aplicación informática en el editor de código de un IDE; lo contrario es el tiempo de ejecución, en donde el programa que se está codificando es puesto en marcha después de ser compilado y depurado.

2.1.4. Plataforma ⁴

² http://es.wikipedia.org/wiki/Entorno_de_desarrollo_integrado

³ <http://olgeferrer.wordpress.com/2011/03/18/definicion-de-dataset-dataview-y-datatable/>

⁴ [http://es.wikipedia.org/wiki/Plataforma_\(inform%C3%A1tica\)](http://es.wikipedia.org/wiki/Plataforma_(inform%C3%A1tica))

En el ambiente informático, una plataforma es un sistema que sirve como fundamento o base, sobre el cual se ponen a funcionar determinados módulos de hardware y software con propiedades similares que son compatibles. Este sistema se define por un estándar, que en torno a este se determina una arquitectura de hardware y a una plataforma de software. Una vez definida las propiedades de una plataforma se establecen: los tipos de arquitectura, el sistema operativo, el interfaz de usuario o el lenguaje de programación con los que son compatibles.

2.1.5. API (Interfaz de Programación de Aplicaciones) ⁵

Se denomina a una API como un conjunto de funciones y procedimientos que ofrece desde una biblioteca, los cuales son utilizados en el desarrollo de otro sistema o aplicación como una capa de abstracción. Es decir, desde el sistema en desarrollo se hace el llamado a ciertas bibliotecas que ofrecen acceso a servicios de uso muy frecuente, por ejemplo, para dibujar figuras geométricas en una pantalla. Los programadores se favorecen de todas las ventajas que ofrece la API, se evitan el trabajo de programar todo desde el principio reduciendo esfuerzos y tiempo de desarrollo.

2.1.6. Modularidad ⁶

Cuando se habla de modularidad, se refiere a la capacidad que posee un sistema para que este sea estudiado o entendido como la unión y fusión de distintas partes, al mismo tiempo interactúan entre sí, pero tales partes funcionan independientemente, cada parte realiza una tarea concreta con el fin de alcanzar en conjunto un objetivo en común. Cada parte en que se desintegra o divide el sistema se le da el nombre de módulo, el mismo que puede ser independiente del resto de los otros módulos. Estos módulos pueden tener la cualidad de comunicarse con otros módulos, a través de la definición de parámetros de entrada y salida.

⁵ http://es.wikipedia.org/wiki/Interfaz_de_programaci%C3%B3n_de_aplicaciones

⁶ <http://es.wikipedia.org/wiki/Modularidad>

2.1.7. Plugin ⁷

Un plugin (su nombre procede del inglés plug-in que significa enchufable) es una aplicación informática especializada en añadir funcionalidades o características específicas a un sistema principal. La adaptación de plugins es muy habitual en reproductores de música, en sistemas de gestión de contenidos, y sobre todo en los navegadores web. Por lo general, un plugin es producido por una compañía diferente a la que originó el programa principal.

2.1.7. Método Deductivo ⁸

Al hablar de deducción, las ideas se enfocan de lo general hacia lo particular. El método deductivo se basa en acumular todos los datos generales aceptados como válidos, para con ello poder deducir a través del razonamiento lógico varias suposiciones; es decir, el método deductivo parte de axiomas ya establecidas como principios generales, para posteriormente aplicarlas a procesos individuales y comprobar así su validez. El razonar deductivamente formaliza en la investigación una de las principales características del proceso de análisis cuantitativo.

2.1.8. Método Inductivo ⁹

La palabra inductivo procede del latín “inducere”, viene del verbo inducir que es un antónimo de concluir o deducir. Al hablar de inducción, las ideas son enfocadas de lo particular a lo general. Se emplea el método inductivo cuando del análisis de los casos particulares se obtienen proposiciones generales, esto quiere decir, que aplicando el método inductivo se puede establecer un principio general, una vez que se hayan realizado los estudios y análisis de los hechos o fenómenos particulares. Al usar el razonamiento inductivo en una investigación, se logra constituir los cimientos sobre el cual descansa el proceso de análisis cualitativo.

⁷ <http://www.saberia.com/2010/01/que-es-un-plugin/>

⁸ http://colbertgarcia.blogspot.com/2008_04_01_archive.html

⁹ http://colbertgarcia.blogspot.com/2008_04_01_archive.html

2.2. FUNDAMENTACIÓN TEÓRICA

La fundamentación teórica se basa en la selección de la información más importante que el investigador debe tomar en cuenta para la ejecución del proyecto.

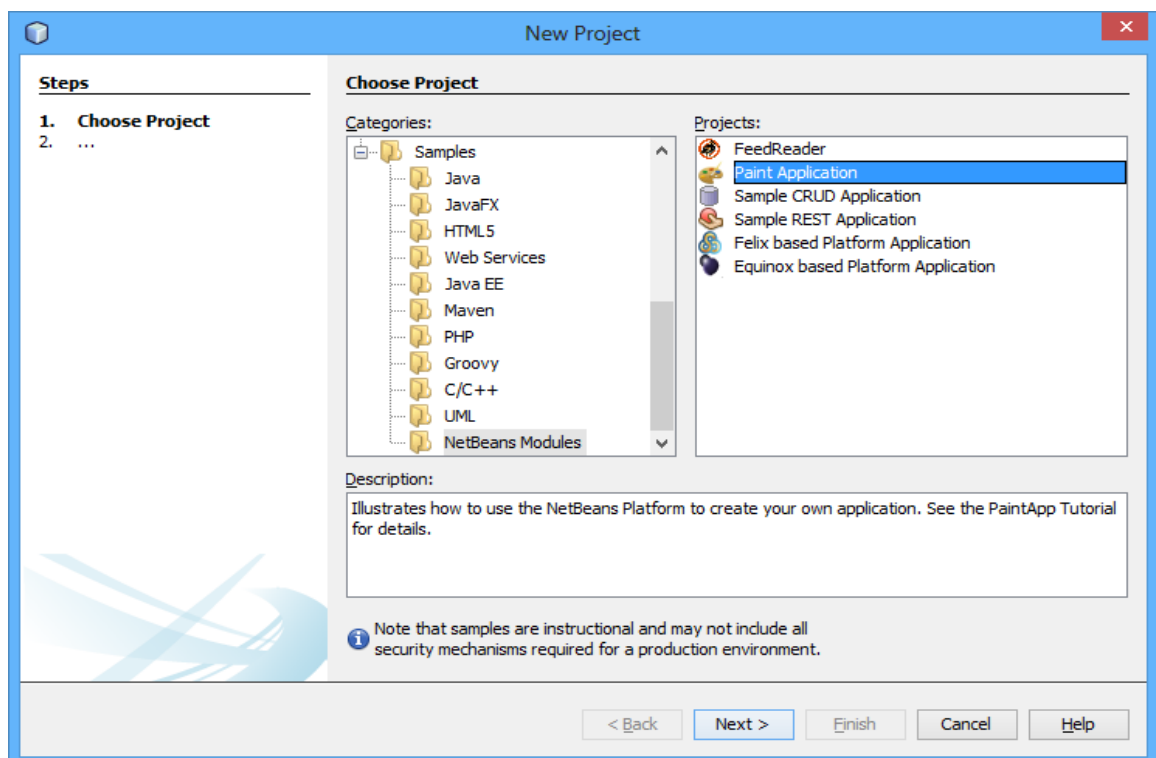
2.2.1. Módulos en Netbeans ¹⁰

Una excelente introducción para el desarrollo de módulos es ofrecida por las aplicaciones de ejemplos que ya vienen contenidas en Netbeans IDE.

Ejemplos de los módulos contenidos en el IDE

Para tener una idea más definida de la consistencia de un módulo, podemos ir al menú principal de Netbeans IDE, escogemos la opción File, seleccionamos New Project, del cual aparecerá un cuadro de diálogo en el cual nos da diferentes opciones para escoger un tipo de proyecto.

ILUSTRACIÓN 1.- CUADRO DE DIÁLOGO DE MUESTRAS DE PROYECTOS NETBEANS MODULES



¹⁰ Böck, Heiko. (2012). *The Definitive Guide to Netbeans Platform 7*. Pág. 30. Apress

En este cuadro de diálogo seleccionamos la categoría Netbeans Modules, del cual se desplegarán una lista de proyectos tipos Module, encapsulados dentro de un proyecto tipo Netbeans Platform. Para entender mejor esta parte vamos a especificar los siguientes conceptos.

Netbeans Platform ¹¹

Netbeans Platform es un extenso framework de Java en el que se pueden establecer grandiosas aplicaciones de escritorio. El mismo Netbeans IDE es uno de los cientos de aplicaciones formados en Netbeans Platform. Netbeans Platform o la plataforma de Netbeans contiene APIs que simplifican la manipulación de ventanas, acciones, archivos y muchos otros elementos característicos en la construcción de aplicaciones.

Cada característica distintiva de una aplicación Netbeans Platform puede ser provista por distintos módulos Netbeans, los cuales se los conocen como plugins.

Módulo Netbeans

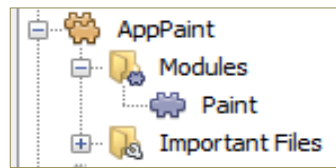
Un módulo Netbeans es un grupo de clases de Java proporcionada por una aplicación con una característica específica.

Incluso podemos crear nuevos módulos para el mismo Netbeans IDE. Por ejemplo, puedes escribir módulos que cumplan con los nuevos conocimientos según el avance de la tecnología requerida por el usuario de Netbeans IDE. Puedes crear un módulo que proporcione una característica de editor adicional.

Con los conceptos ya mencionados ahora se puede abrir un proyecto Netbeans Platform contenido en el IDE de Netbeans. Como ejemplo de demostración se selecciona el proyecto Paint Application, allí se muestra la siguiente vista del proyecto AppPaint en la pestaña Projects.

¹¹ <https://netbeans.org/features/platform/all-docs.html>

ILUSTRACIÓN 2.- PROYECTO NETBEANS MODULE



Donde el ícono de color naranja es una aplicación Netbeans Platform, y dentro de este se encuentra un ícono de color morado, el cual vendría a ser un módulo Netbeans que contiene características específicas, en este caso, el módulo Paint, que contiene clases de Java elaboradas con un objetivo en común para la aplicación Netbeans Platform AppPaint.

Creación de módulos

Netbeans IDE provee un asistente de ayuda (wizard) para la creación de aplicaciones de Netbeans Platform.

- 1.- En el entorno de trabajo de Netbeans IDE selecciona *File* → *New Project* en el menú principal. Diferentes categorías de proyectos son mostrados en el lado izquierdo del cuadro de dialogo. Aquí selecciona *Netbeans Modules*. Luego escoge el tipo de proyecto *Netbeans Platform Application* sobre el lado derecho.
- 2.- En la siguiente página, requiere un nombre para la aplicación y se escoge el directorio donde el proyecto será guardado. Los demás campos pueden ser dejados en blanco.
- 3.- Dale clic en el botón *Finish* para crear una nueva aplicación Netbeans Platform.
- 4.- Ahora podemos crear un primer módulo. Otro asistente de ayuda está disponible para realizar esta tarea. Dale clic en *File* → *New Project* en el menú. En el cuadro de dialogo mostrado escoge la categoría *Netbeans Modules* del lado izquierdo, y luego el tipo de proyecto *Module* sobre el lado derecho.
- 5.- Dale clic en el botón *Next* para ir a la siguiente página e ingrese un nombre para el módulo. Luego seleccione la opción *Add to Module Suite*, y seleccione

la aplicación Netbeans Platform creada anteriormente desde la lista desplegable.

6.- En la siguiente página defina el Code Name Base y el nombre a mostrar del módulo.

7.- Finalmente se da clic en el botón *Finish*, y el nuevo módulo será generado por Netbeans IDE.

Ahora ya podemos comenzar a desarrollar un nuevo módulo utilizando las diferentes opciones que el IDE proporciona para el desarrollo de módulos.

Tutorías para el desarrollo de módulos Netbeans

Para comenzar a desarrollar módulos para Netbeans IDE nos podemos instruir con el libro *The Definitive Guide to Netbeans Platform 7*; este libro está detallado en la sección de Bibliografía de la presente tesis de grado.

Este libro ilustra las numerosas innovaciones que Netbeans IDE origina desde la versión 7 en adelante.

Para la construcción del módulo que fue realizado en el presente trabajo de tesis, se hace mención de los capítulos del libro que ocuparon mayor importancia.

- ✓ The NetBeans Module System
- ✓ Actions
- ✓ Data and Files
- ✓ Dialogs and Wizards
- ✓ Visual Library

Alternativamente, también podemos acudir al sitio web de Netbeans, ubicándolo con el link <https://netbeans.org/>. Dentro de este sitio se selecciona el apartado Netbeans Platform, en el cual se encuentran las descripciones, características, comparaciones, usos y toda la documentación para aprender a desarrollar en este nuevo tipo de proyectos informáticos.

2.2.2. Extreme Programming XP ¹²

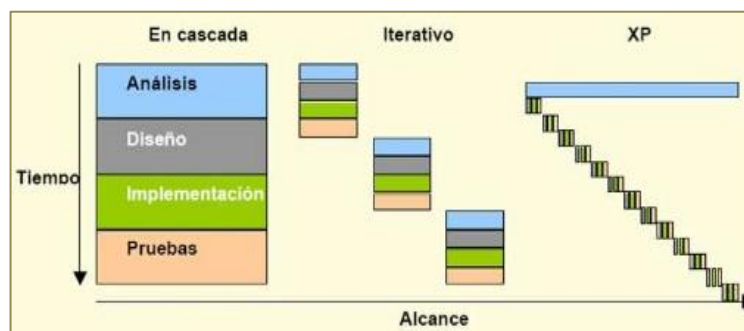
La programación extrema es una metodología de desarrollo en la ingeniería de software. Los usuarios del Extreme Programming consideran que los cambios de los requisitos solicitados sobre la marcha es una particularidad natural que todo programador desea realizar en pleno desarrollo de software, ya que el programador está apto para adaptarse a los cambios en cualquier fase del ciclo de vida del proyecto en lugar de definirlos al comienzo.

Fases del XP

Como toda metodología de desarrollo de software, la programación extrema cuenta con un ciclo de vida que consiste en seis fases, estas son: Exploración, Planificación de la entrega, Iteraciones, Producción, Mantenimiento y Muerte del proyecto. La metodología XP se basa en buscar siempre una retroalimentación, con el propósito de esclarecer las dudas con el cliente, sobre las características que deben tener prioridad. (Joskowics, 2008)

Comenzando desde la evolución larga del método de cascada, y atravesando por los ciclos más cortos del método iterativo, el XP es una mezcla repetitiva de los métodos tradicionales.

ILUSTRACIÓN 3.- ALCANCE DEL XP



La programación extrema se diferencia de las metodologías tradicionales por su proceso ágil para el desarrollo en la ingeniería de software.

¹² http://es.wikipedia.org/wiki/Programaci%C3%B3n_extrema

Valores que inspiran el XP ¹³

La programación extrema enfatiza en que cada integrante del equipo de desarrollo del software influya con los siguientes valores.

TABLA 2.- VALORES DEL XP

VALORES	DESCRIPCIÓN
SIMPLICIDAD	La base de la programación extrema es la simplicidad. Se tiene que simplificar el diseño para que el desarrollo sea ágil y el mantenimiento sea fácil. Para mantener la simplicidad del código es necesaria una refactorización en el código, así siempre se mantendrá un código simple a medida que crece.
COMUNICACIÓN	La comunicación entre programadores crea el establecimiento de una autodocumentación del código, de esa manera poder comprender el código fuente del software. Por otra parte, la comunicación con el cliente debe ser fluida ya que en el XP el cliente forma parte del equipo de desarrollo.
RETROALIMENTACIÓN	Al estar el cliente integrado en el proyecto de desarrollo, su opinión sobre la situación del proyecto se conoce en el tiempo real. Al presentar el cliente los cambios que se requiere realizar en el proyecto, se disminuye el tiempo perdido en rehacer partes que no cumplen con los requerimientos del cliente, y ayuda sobre todo a los programadores a centrarse en lo que realmente es importante.
CORAJE Y VALENTÍA	Lo práctico requiere de tener mucha valentía, ya que en pleno desarrollo de software, siempre se busca diseñar y programar para el presente y no para el mañana. La valentía se centra en el rediseño del código las veces que sea menester, no importando el esfuerzo que tenga que hacer, siempre y cuando el programador se sienta cómodo con el nuevo restablecimiento de código.
RESPECTO	El respeto se refleja en el compañerismo que debe haber entre los programadores, al entender que no se deben realizar cambios que hagan que retrase el trabajo de sus compañeros o que las pruebas fallen. Entre los que integran en el desarrollo del proyecto se respetan porque perennemente luchan por la calidad del producto.

¹³ es.wikipedia.org/wiki/Programación_extrema

Actividades del XP

La programación extrema describe cuatro actividades básicas que son efectuadas dentro del proceso de desarrollo de software. La siguiente tabla describe estas actividades.

TABLA 3.- ACTIVIDADES DEL XP

ACTIVIDAD	DESCRIPCIÓN
ESCUCHAR	XP se cimenta en la comunicación y en prácticas que demandan escuchar activamente, ya que no depende de una sumisión a una documentación formal como en las otras metodologías. Cuando el equipo que desarrolla el proyecto escucha al cliente, existe una retroalimentación en la lógica del negocio y del sistema detrás de dicho negocio.
PRUEBAS (TESTING)	Las pruebas no deben ser una actividad suplementaria que se realiza cuando se finaliza un sistema, más bien debe ser una actividad integral que se lleve a cabo durante todo el proceso de desarrollo del proyecto. En XP, las pruebas se escriben incluso antes de comenzar a codificar, de modo que se consigue calidad integrada en el sistema desde su inicio. Si se hace al final, los costos de los cambios son generalmente altos.
PROGRAMAR	En XP se considera a la programación como actividad principal, refinada por prácticas como: la refactorización, programación en pareja y revisión constante del código. Con un diseño pequeño y de incremento, los desarrolladores escriben solamente el código necesario para formular sus ideas, de tal manera que se considera el código como intermediario para explicar la lógica, los algoritmos y el flujo de sistema.
DISEÑAR	Una de las ideas radicales en XP es que el diseño siempre debe estar en evolución, y en constante crecimiento durante el proyecto. El diseño no debe permanecer estancado, ni asignado a un solo rol, sino que debe ser dinámico y orientado al equipo. En XP no se limitan las actividades del diseño, más bien se acepta su evolución.

2.2.3. Documentos XML ¹⁴

XML conocido así por sus siglas en inglés Extensible Markup Language, que en español significa Lenguaje de marcado extensible. El XML es un lenguaje de marcado, fabricado por el World Wide Web Consortium - W3C, este componente es utilizado para almacenar datos en una forma legible.

Al igual que en HTML, en XML también se permite definir una gramática de lenguajes específicos para poder estructurar grandes documentos.

XML aparte de poseer una tecnología muy sencilla, implica a su alrededor de otras tecnologías que la complementan y la innovan, haciéndola mucho más grande con posibilidades mucho mayores. Actualmente, XML ocupa un papel muy relevante para desarrollar software, ya que es compatible entre diferentes sistemas y logra distribuir la información de modo seguro, confiable y fácil.

Ventajas

- ❖ Es extensible, después de su diseño y de haber sido puesto en producción, se da la posibilidad de extender el archivo XML con el añadido de nuevas etiquetas, de esa manera, se pueda continuar utilizando sin ninguna complicación.
- ❖ El analizador es un componente estándar. No daría la necesidad de crear un analizador específico para cada versión del lenguaje XML. Esto facilita el empleo de cualquiera de los analizadores que están disponibles.
- ❖ Si un tercero espera usar un documento que ya ha sido creado en XML, le será sencillo entender su estructura y procesarla. Esto va a permitir mejorar la compatibilidad entre diferentes aplicaciones. Se podrán conectar aplicaciones de plataformas distintas.

¹⁴ http://es.wikipedia.org/wiki/Extensible_Markup_Language

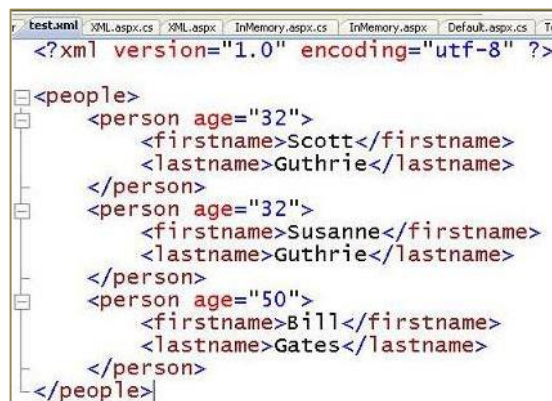
- ❖ Se transforman los datos en una información, pues a los datos se les añade un significado concreto y se los asocia a un contexto, con lo cual se obtiene flexibilidad para estructurar documentos.

Estructura de un documento XML

Un documento XML conserva su información de forma estructurada, este se compone de partes bien definidas, y esas partes son construidas a su vez por otras partes. Estas partes toman el nombre de <elementos> y se las señala mediante etiquetas.

Una etiqueta consiste en una marca hecha en el documento, que apunta un fragmento de éste como un elemento. Esta es una partícula de información con un sentido claro y bien definido. La forma que tienen las etiquetas son así: <nombre>, donde “nombre” es el nombre del elemento al que se apunta.

ILUSTRACIÓN 4.- ESTRUCTURA DE UN DOCUMENTO XML



```
<?xml version="1.0" encoding="utf-8" ?>
<people>
  <person age="32">
    <firstname>Scott</firstname>
    <lastname>Guthrie</lastname>
  </person>
  <person age="32">
    <firstname>Susanne</firstname>
    <lastname>Guthrie</lastname>
  </person>
  <person age="50">
    <firstname>Bill</firstname>
    <lastname>Gates</lastname>
  </person>
</people>
```

Partes de un documento XML

Un documento XML se forma de las siguientes partes: prólogo y cuerpo, el cuerpo del documento contiene elementos. En el cuerpo se hallan textos de etiquetas que poseen una gran diversidad de efectos positivos o negativos en el relato opcional a la que se refiere el documento.

Prólogo.- Aunque no es obligatorio incluir esta parte, todo documento XML empieza con unas líneas que detallan la versión del documento XML, el tipo del

documento y otras cosas que son opcionales. El prólogo de un documento XML esta compuesto por un enunciado XML, una declaración de tipo de documento, algunos comentarios e instrucciones de procesamiento.

Por ejemplo: `<?xml version="2.3" encoding="UTF-8"?>`

Cuerpo.- En contraste con el prólogo, la parte del cuerpo es totalmente obligatoria. El cuerpo debe de incluir tan solo un elemento raíz, esta característica es indispensable para que el documento esté bien formado.

Ejemplo: `<edit_mensaje> (...) </edit_mensaje>`

Elementos.- Los elementos en un documento XML pueden tener contenido (estos pueden ser más elementos, pueden ser caracteres o ambos), o bien estar vacíos.

Estos elementos pueden tener atributos, por medio de estos atributos se pueden incorporar características o propiedades a los elementos de un documento. El valor se lo debe escribir entre comillas. Por ejemplo un elemento "empleado" puede tener un atributo "nombre", otro "edad" y otro "teléfono", con valores "Pedro Salazar", "35" y "0982726543", respectivamente.

`<empleado nombre="Pedro Salazar" edad="35" teléfono="0982726543"> (...) </empleado>`

2.3. MARCO REFERENCIAL

Como ejemplos de software referenciales para la consecución del presente proyecto se escogieron investigar a profundidad los siguientes apartados.

2.3.1. DataSet de .NET ¹⁵

El componente DataSet de ADO.NET es un reservorio de datos que trabaja en la memoria del cliente cuando se aplica en aplicaciones Windows, y trabaja en la memoria del servidor cuando se aplica para las aplicaciones Web, estas

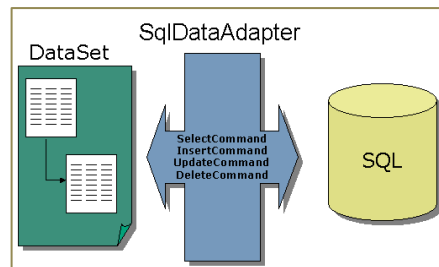
¹⁵ [http://www.elguille.info/colabora/NET2005/FernandoLuque DataSetsI.htm](http://www.elguille.info/colabora/NET2005/FernandoLuque%20DataSetsI.htm)

trabajan en ambientes desconectados, la estructura del componente DataSet es equivalente a la de una base de datos en el cual se exhiben de manera jerárquica las tablas, filas y columnas que este contiene.

¿Cómo trabaja?

La manera de trabajar con datos a partir de una base de datos en aplicaciones con Visual Studio .NET depende de un adaptador, el cual relaciona el objeto DataSet con la tabla que corresponde en la base de datos, y este se encarga de la inserción, modificación, eliminación y selección de queries.

ILUSTRACIÓN 5.- DATAADAPTER Y DATASET DE .NET



Existen muchas maneras de trabajar con un DataSet, y este se puede aplicar de manera conjunta o independiente. Por medio de la programación en un DataSet se puede crear una DataTable, DataRelation y una Constraint en un DataSet y rellenar las tablas con datos, también se puede llenar el DataSet con tablas de datos desde un origen de datos relacional existente a través del DataAdapter, y también un DataSet puede cargar y persistir su contenido de datos mediante archivos XML.

Ventajas

- ✓ El Dataset puede trabajar con varias tablas, estas pueden ser manejadas de forma individual o relacionadas.
- ✓ Se conecta a diferentes fuentes de datos como SQL Server, Oracle, etc.
- ✓ Se puede enlazar los datos almacenados en el DataSet, directamente a los controles de los formularios Windows.

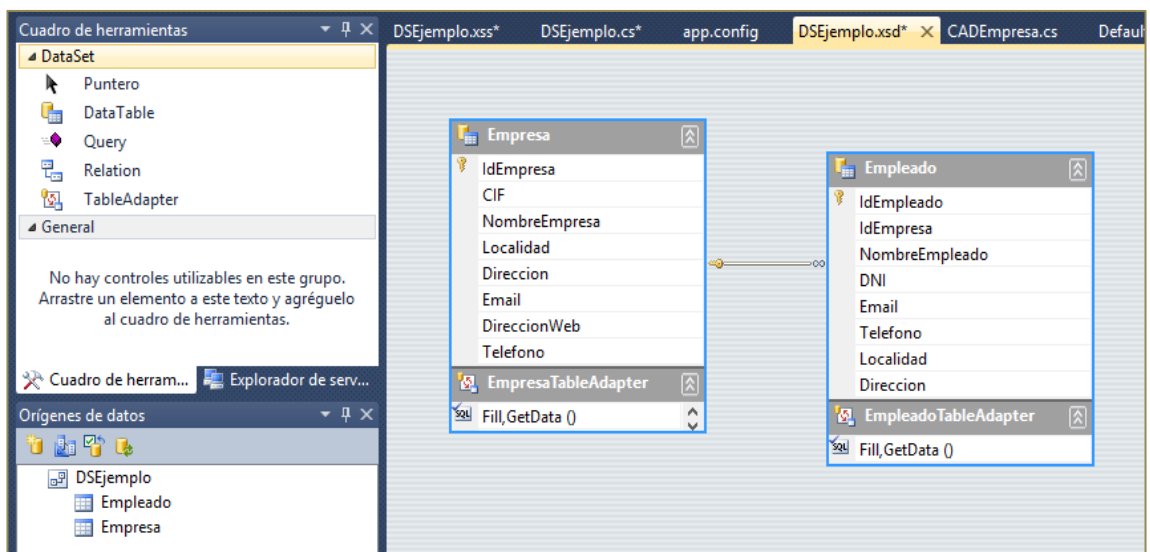
Partes principales

- ✓ Un DataSet incluye la colección de Tablas contenidos o creados en este.
- ✓ Un DataSet incluye una colección de Relaciones creados entre las tablas.
- ✓ Las Tablas creadas en el DataSet contienen colecciones de Columnas y Filas que están contenidas en las tablas.

Diseñador del DataSet ¹⁶

El Diseñador del DataSet es un grupo de herramientas visuales que el programador puede utilizar para crear y modificar los conjuntos de datos, y cada uno de los elementos individuales contenidos. El tablero diseñador del DataSet da la facilidad de visualizar cada uno de los objetos contenidos en un conjunto de datos con tipo. Con el Diseñador del DataSet, el programador puede crear y modificar los TableAdapters, realizar consultas de TableAdapters, DataTables, DataColumnns, y DataRelations.

ILUSTRACIÓN 6.- DISEÑADOR DEL DATASET DE .NET



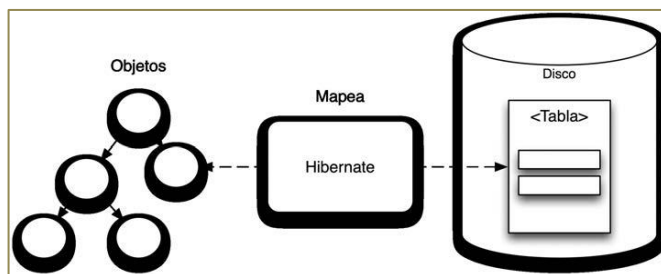
Después de abrir los objetos pertenecientes a un conjunto de datos dentro del Diseñador del componente DataSet, el programador puede realizar acciones tales como: la edición de un conjunto de datos, la creación de TableAdapters, edición de TableAdapters, creación de consultas de TableAdapters, en fin, el programador puede extender la funcionalidad de TableAdapters.

¹⁶ <http://msdn.microsoft.com/es-es/library/314t4see.aspx>

2.3.2. Hibernate ¹⁷

La herramienta de mapeo de datos Hibernate es un framework ORM (Object Relational Mapping) cuya tarea principal es permitir que un desarrollador pueda mapear objetos contra los registros de una base de datos. Al igual que todas las herramientas ya conocidas de su tipo, Hibernate busca solucionar la problemática en la diferencia que existe entre los dos modelos de datos coexistentes en una aplicación: el usado en la memoria del computador (modelo orientado a objetos) y el usado en la base de datos (modelo relacional).

ILUSTRACIÓN 7.- MAPEO DE HIBERNATE



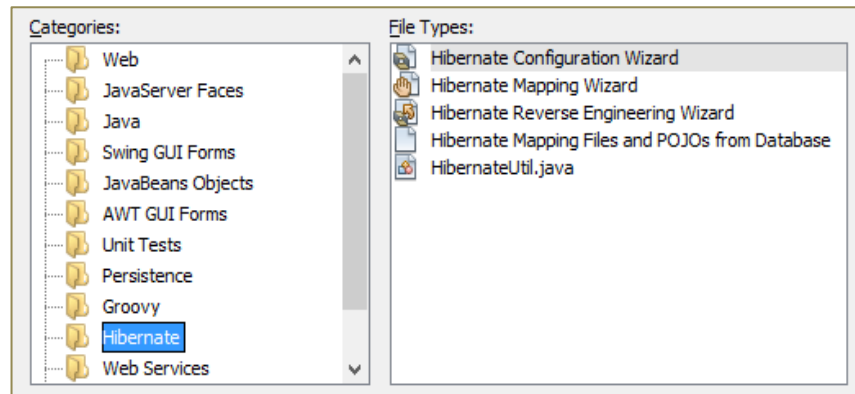
Para conseguir esto, la herramienta Hibernate permite al desarrollador de aplicaciones detallar cómo está construido su modelo de datos, de que relaciones constan y qué forma tienen. Con esta información, Hibernate permitirá a la aplicación manipular los datos que están en la base de datos operando sobre objetos, con las características de la POO. Hibernate convertirá los datos que compatibilizan Java y SQL. Hibernate crea las sentencias SQL y libera al desarrollador del manejo manual de los datos que resultan de la ejecución de dichas sentencias, manteniendo la portabilidad entre todos los motores de base de datos.

Tareas de Hibernate

En Netbeans IDE se encuentran la categoría de los tipos de archivos de Hibernate, donde se puede adicionar a cualquier proyecto de Java los ficheros que realizan las tareas de funcionamiento de Hibernate.

¹⁷ Caules, C.Á. (2012). Arquitectura Java Sólida. Pag. 155. Sun Certified Enterprise Architecture

ILUSTRACIÓN 8.- CATEGORÍA HIBERNATE EN NETBEANS IDE



Hibernate Configuration Wizard

Este es el fichero principal para la configuración del framework. Aquí es en donde se debe configurar el driver JDBC de acceso a datos, se constata el servidor de base de datos, el usuario y la password, así como los archivos de mapeo que se van a utilizar en las distintas clases.

Hibernate Mapping Wizard

Este tipo de archivo Hibernate permite crear un archivo de mapeo de Hibernate con la extensión .hbm.xml, el cual almacena la información selecta que se refiere un objeto que corresponde a una clase determinada.

Hibernate Reverse Engineering Wizard

Por medio de un asistente, este crea los archivos de ingeniería inversa de Hibernate con extensión reveng.xml. En esta opción Hibernate realiza una ingeniería inversa, la cual consiste en crear un archivo reveng.xml de almacenamiento de la información de las tablas que conforman la base de datos, que luego pueden ser manipuladas para la generación de POJOs.

Hibernate Mapping Files and POJOs from Database

Crea archivos de mapeo de Hibernate y POJOs basados en una base de datos relacional existente. Por medio de este asistente se crean los archivos de mapeo y clases Java de las tablas que corresponden a la base de datos.

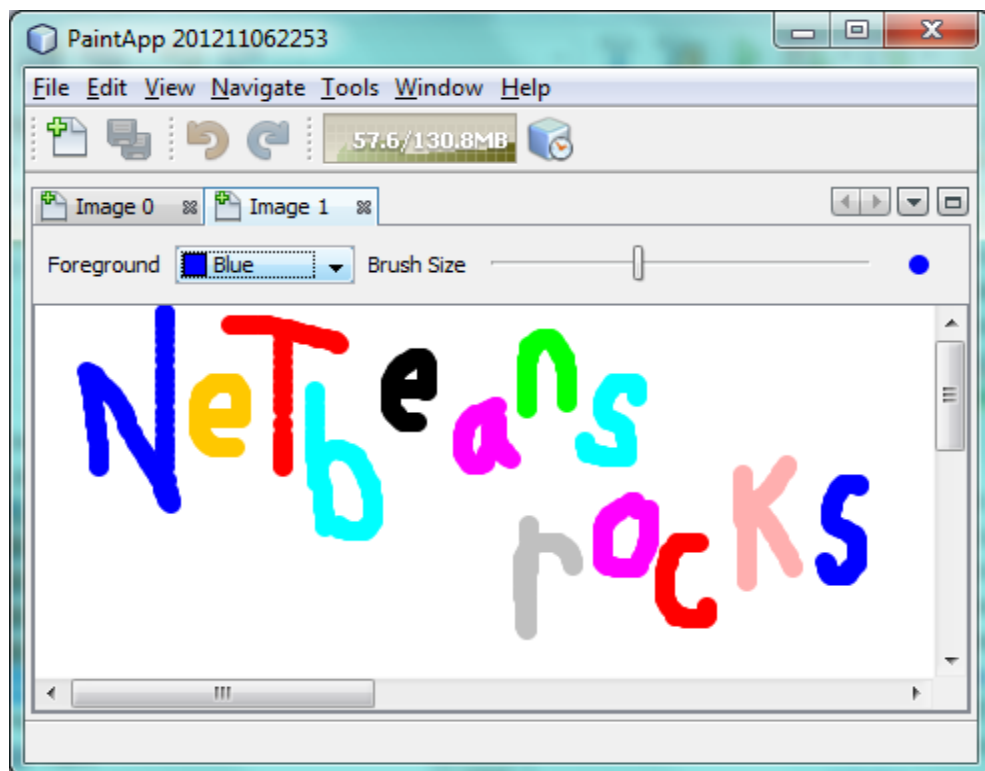
HibernateUtil.java

Crea una clase de utilidad para Hibernate, este permite una fácil recuperación de los objetos de sesión de Hibernate.

2.3.3. Proyecto AppPaint ¹⁸

AppPaint es un ejemplo de Netbeans Platform Application, utilizado por los principiantes de Netbeans Platform mientras aprenden acerca del flujo de trabajo de desarrollo y uso de los API de Netbeans.

ILUSTRACIÓN 9.- APLICACIÓN APPPAINT



El tutorial de AppPaint nos lleva a través de los fundamentos del uso de Netbeans IDE para desarrollar aplicaciones en Netbeans Platform. Al desarrollar aplicaciones sobre la plataforma de Netbeans se estará trabajando sobre el núcleo de Netbeans. Todos los módulos que pertenecen al IDE que son relevantes para su aplicación están excluidos, pero los que son útiles se

¹⁸ <https://platform.netbeans.org/tutorials/nbm-paintapp.html>

mantienen. Mediante la reutilización de características fácilmente disponibles en el núcleo del IDE se ahorra gran cantidad de tiempo.

2.4. FUNDAMENTACIÓN LEGAL

La fundamentación legal para el desarrollo del presente trabajo de tesis lo da la constitución de la república, el plan Nacional del Buen Vivir, las reglamentaciones del SENESCYT y la Ley Orgánica de Educación Superior.¹⁹

El Decreto N° 865 que fue emitido por el Presidente de la república Rafael Correa Delgado entró en vigencia desde su publicación en el Registro Oficial.

El Decreto N° 865 en la Disposición General Cuarta, establece lo siguiente: Las instituciones de educación superior tendrán la obligación de incorporar el uso de programas informáticos realizados con software libre, en caso de que la funcionalidad de estos programas sean equivalentes o superiores al software propietario. Las universidades y escuelas politécnicas deben ser responsables por la aplicación del presente artículo.

ILUSTRACIÓN 10.- DIFERENCIAS ENTRE WINDOWS Y LINUX



¹⁹ <http://casm37.wordpress.com/2011/10/05/software-libre-en-el-reglamento-de-la-ley-de-educacion-superior/>

CAPITULO III

METODOLOGÍA DE LA INVESTIGACIÓN

3.1. MATERIALES Y METODOS

3.1.1. Materiales y presupuesto

Para el desarrollo del presente trabajo de tesis se suministraron los siguientes presupuestos de Recursos Humanos, Hardware, Software y Materiales de oficina, resumidos en la siguiente tabla.

TABLA 4.- MATERIALES Y PRESUPUESTO

Cantidad	Descripción	Valor Unitario	Total
RECURSOS HUMANOS			
1	Desarrollador de software	0.00	0.00
1	Asesoría de Tutor	0.00	0.00
HARDWARE			
1	Computadora Intel Core i3 2.6 MHz o superior 4 Gb en Memoria RAM 750 Gb de Disco duro	780.00	780.00
1	Impresora	250.00	250.00
1	Decodificador de Internet	50.00	50.00
5	Meses de Servicio de Internet	250.00	250.00
1	Cableados y Herramientas de red	25.00	25.00
SOFTWARE			
1	Windows 8	170.00	170.00
1	Netbeans IDE 7.3.1	0.00	0.00
1	JDK 7u40	0.00	0.00
1	Gestores de Bases de datos MySQL, otros.	0.00	0.00
1	LibreOffice Writer	0.00	0.00
1	Apache Tomcat 7	0.00	0.00
MATERIALES DE OFICINA			
10	Paquetes de hojas A4	4.50	45.00
4	Cartuchos de tinta negra	20.00	80.00
2	Cartuchos de tinta de color	25.00	50.00
4	Carpetas plásticas	0.40	1.60
10	Lápices	0.25	2.50
INVERSIÓN TOTAL:			1,704.10

3.1.2. Factibilidad

Económica

Según las horas de análisis, consultoría, desarrollo de software, materiales y equipos, se obtuvo un valor aproximado de 1,704.10 dólares por la realización

total del trabajo según el presupuesto de la sección previa; sin embargo, ya que el trabajo fue desarrollado por mí y por tratarse de mi tesis de grado no tendrá ningún costo, a la vez se podrá contar con los equipos que la universidad permita el acceso disponible, por lo tanto, el trabajo es económicamente factible.

Técnica o tecnológica

En la cuestión de conocimientos y habilidades en el manejo de métodos, procedimientos y funciones requeridas para el desarrollo y planificación del presente trabajo, contando con el apoyo y la guía del tutor, no existen inconvenientes en la disposición de cada una de estas características intelectuales. De igual manera se cuenta con los equipos, herramientas y materiales requeridos, por lo que se puede decir que el trabajo es tecnológicamente factible.

3.1.3. Métodos de investigación

Por tratarse de un trabajo que involucra el ámbito tecnológico basado en el razonamiento para resolver una problemática en un ambiente de desarrollo de aplicaciones informáticas, se optó por utilizar los métodos de Investigación Deductivo e Inductivo.

Deductivo

El método deductivo permite representar conceptos o principios generales para la consecución de los aspectos individuales. Para el desarrollo de la presente investigación se incorporaron proposiciones basadas en la deducción, por ejemplo.

✓ El desarrollo de aplicaciones a través de código fuente escrito línea por línea causa lentitud, y los componentes gráficos ayudan a aligerar y abreviar los procedimientos manuales. En conclusión, emplear componentes gráficos ayudaría a realizar un desarrollo rápido de aplicaciones.

Inductivo

El método inductivo permite obtener conclusiones generales a partir de casos, hechos o fenómenos particulares. En este caso, para el desarrollo de la presente investigación se realizaron declaraciones generadas a base del razonamiento inductivo para ponerlos en práctica. La inducción también sirvió para poder plantear la hipótesis, por ejemplo.

✓ Un sistema con interfaz gráfica simplifica los procesos empresariales, una aplicación informática con editores gráficos agiliza los trabajos requeridos. En conclusión, un asistente gráfico especializado en el acceso a datos facilita el desarrollo de aplicaciones con conexión a una base de datos.

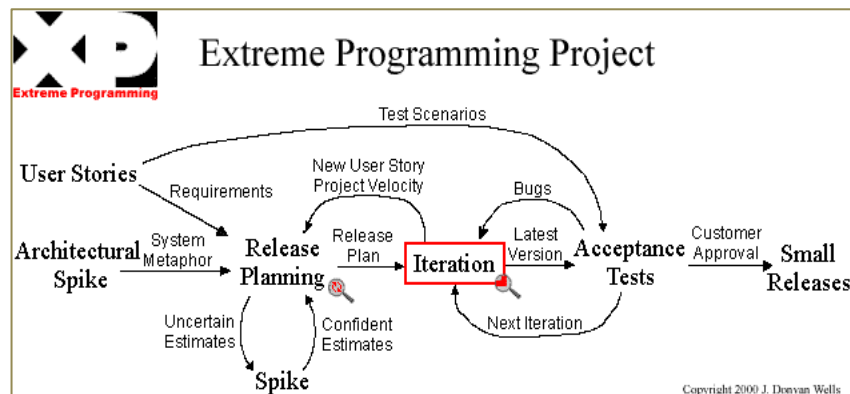
3.1.4. Metodología de desarrollo

La hipótesis planteada sustenta que realizar el manejo de datos a través de un asistente gráfico de acceso a datos facilita el desarrollo de las aplicaciones con conexión a base de datos. Por lo tanto, es indispensable crear un componente de asistencia gráfica, el cual contenga acciones que realicen las operaciones hacia la base de datos, brindando un soporte eficiente al programador de software. Este componente tiene que estar integrado en el IDE para su uso práctico en tiempo de desarrollo de software.

De esa manera se determinó crear un nuevo módulo para Netbeans IDE, o lo que más se conoce como plugins. Una vez que el plugins esté instalado, se integran nuevas acciones en el entorno de trabajo de Netbeans IDE.

Para el desarrollo de este plugins se optó por utilizar la metodología de desarrollo de Programación Extrema (XP Extreme Programming). El preámbulo de las características, fases, valores, y actividades de esta metodología se encuentran en la sección de FUNDAMENTACIÓN TEÓRICA del presente documento de tesis. La siguiente ilustración muestra con más detalles las actividades que se realizan en un proyecto de programación extrema.

ILUSTRACIÓN 11.- DIAGRAMA DINÁMICO DEL XP



Como se visualiza en el diagrama dinámico del XP, el análisis del proyecto informático comienza con la implementación de fichas de Historias de Usuarios donde se especifican los requerimientos del usuario. Luego se realiza una planificación metáfora para lanzar una primera versión del software con el fin de realizar retroalimentaciones que mejoren los requisitos más importantes, este proceso se realiza iterativamente tras la realización de pruebas aprobadas por el usuario final. La metodología XP se basa en la retroalimentación de requisitos que son optimizadas tras el rediseño en las fases de desarrollo.

Fases de desarrollo

XP comprende las siguientes fases: Exploración, Planificación de la entrega (Release), Iteraciones, Producción, Mantenimiento y Muerte del proyecto.

En la fase de exploración se constituyó las historias de usuarios que especifican los requerimientos de los clientes (en este caso, los programadores de software especifican los requisitos que desean adquirir para el desarrollo de aplicaciones con conexión a base de datos).

En la fase de Planificación de la entrega (Release) se estableció la prioridad para cada historia de usuario, en donde se estimó una cantidad de tiempo en días necesarios para realizar cada historia de usuario.

En la fase de Iteraciones se realizó un plan de iteraciones sostenido en presentar avances del producto según la prioridad de las historias de usuarios.

La fase de producción consta de revisiones a base de pruebas y evaluaciones del software, los resultados de la pruebas son detalladas en la sección de RESULTADOS del presente trabajo de tesis.

En la fase de mantenimiento se realizó un soporte para que el usuario final (programador) aporte con resultados que permitan el surgimiento de nuevas iteraciones, un tutorial sobre el uso del software que se encuentra detallado en la sección de ANEXOS.

A continuación, se presentan los detalles de los procesos realizados en las fases de la metodología de programación extrema, acotando primeramente la asignación del nombre del software desarrollado.

3.1.4.1. Asignación del nombre del proyecto

Para asignar un nombre al proyecto informático, se sustentó en la observación de los nombres de los diversos componentes de Java. Algunos componentes de Java tienen como prefijo la letra J, de manera que existen componentes con el nombre de JPanel, JForm, etc.

El proyecto desarrollado tuvo referencia de la plataforma de .NET, donde ya existe un componente con las funcionalidades de acceso a datos con el nombre de DataSet, que en español significa Conjunto de Datos.

Para comenzar con la creación de un nuevo módulo en Netbeans IDE, se requirió introducir un nombre para el proyecto con el cual se da a conocer al mundo, y un código de nombre base para designar el nombre del plugins o archivo NBM. Finalmente se establecieron los nombres de la siguiente manera.

TABLA 5.- NOMBRE DEL PROYECTO Y ARCHIVO NBM

Nombre del plugins:	JDataSet
Nombre del archivo NBM:	org-netbeans-dataset1.nbm

3.1.4.2. Historias de Usuarios

La fase de exploración se origina según la observación directa de los aspectos requeridos por los usuarios (en este caso, los programadores de software). Los requerimientos del usuario se basan en el procedimiento de acceso a datos al momento de desarrollar aplicaciones utilizando Java como lenguaje de programación y Netbeans como IDE. También se toma como referencia base el objeto DataSet de ADO.NET para idealizar las acciones indispensables.

Conectar con la Base de Datos

TABLA 6.- HISTORIA DE USUARIO CONECTAR CON LA BD

HISTORIA DE USUARIO	
Historia N°: 01	Usuario: Programador
Nombre de la historia: Conectar con la base de datos	Iteración asignada: 01
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Programador responsable: Daniel De los Santos	Módulo: Estructura
Descripción: Cuando el programador se encuentra desarrollando una aplicación, requiere conectarse a una base de datos que ya esté totalmente lista para soportar las operaciones con datos.	
Observaciones: El programador utiliza una URL como cadena de conexión.	

El usuario (programador) requiere de la conexión a una base de datos para desarrollar las operaciones internas de una aplicación utilizando la escritura del código fuente del lenguaje de programación Java.

Visualizar el modelo estructural

TABLA 7.- HISTORIA DE USUARIO VISUALIZAR EL MODELO ESTRUCTURAL

HISTORIA DE USUARIO	
Historia N°: 02	Usuario: Programador
Nombre de la historia: Visualizar el modelo estructural de la base de datos	Iteración asignada: 01
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta

Programador responsable: Daniel De los Santos	Módulo: Estructura
Descripción: El programador tiene que acudir al gestor de base de datos para visualizar los objetos del modelo estructural de la base de datos. El modelo estructural se refiere a las tablas con sus campos, tipos de dato, relaciones, etc.	
Observaciones: en el IDE de Netbeans no existe una opción para poder visualizar los objetos de la base de datos.	

En la construcción de un software, el programador tiene que tener en mente la imagen del diagrama del modelo relacional o estructural de la base de datos para desarrollar los métodos y funciones que realizan las transacciones de los datos de la aplicación. Para esto, el programador acude constantemente al sistema de gestión de la base de datos (DBMS) para visualizar el diagrama en caso de que exista esta opción.

Crear métodos con sentencias SQL

TABLA 8.- HISTORIA DE USUARIO CREAR MÉTODOS CON SENTENCIAS SQL

HISTORIA DE USUARIO	
Historia N°: 03	Usuario: Programador
Nombre de la historia: Crear métodos con sentencias SQL	Iteración asignada: 02
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Programador responsable: Daniel De los Santos	Módulo: Estructura
Descripción: El programador requiere crear clases que contengan los métodos de consulta, inserción, actualización o eliminación de datos, los cuales retornen los valores requeridos según la sentencia SQL efectuada.	
Observaciones: En Java solo se puede realizar este procedimiento por medio de código fuente.	

Así como el programador requiere establecer métodos para la conexión de una base de datos, también requiere codificar los métodos del sistema que realicen las operaciones transaccionales hacia la base de datos a través del envío de sentencias SQL. El programador desarrolla estos métodos por medio de la escritura del código fuente puro del lenguaje de Java, según sea el caso, el programador utiliza los datos adquiridos desde la base de datos como mejor sea conveniente para el sistema informático.

Ejecutar los métodos dentro del proyecto

TABLA 9.- HISTORIA DE USUARIO EJECUTAR MÉTODOS DENTRO DEL PROYECTO

HISTORIA DE USUARIO	
Historia N°: 04	Usuario: Programador
Nombre de la historia: Ejecutar los métodos dentro del proyecto Java	Iteración asignada: 03
Prioridad en negocio: Media	Riesgo en desarrollo: Media
Programador responsable: Daniel De los Santos	Módulo: Estructura
Descripción: El programador requiere llamar al método previamente elaborado con la sentencia efectuada, para ponerla en funcionamiento dentro de la aplicación en desarrollo.	
Observaciones: De nada sirve crear clases con métodos que contengan las sentencias SQL, sin poder integrarlos al proyecto en construcción.	

El programador crea objetos de la clase contenedora de los métodos de selección, inserción, actualización y eliminación de datos para ponerlos en funcionamiento dentro de la aplicación.

3.1.4.3. Análisis de prioridades y estimación

En la fase de planificación de entregas se realizó el análisis de prioridades y estimación del tiempo en que se va a desarrollar cada historia de usuario, esto se lo puede verificar en la siguiente tabla.

TABLA 10.- ANALISIS DE PRIORIDADES Y ESTIMACIÓN

N°	HISTORIA DE USUARIO	PRIORIDAD	ITERACIÓN	ESTIMACIÓN
1	Conectar con la base de datos	ALTA	01	30 días
2	Visualizar el modelo estructural	ALTA	01	50 días
3	Crear métodos con sentencias SQL	ALTA	02	40 días
4	Ejecutar los métodos dentro del proyecto	MEDIA	03	20 días
TOTAL EN DÍAS				140 días
TOTAL EN HORAS (8 HORAS DIARIAS)				1120 horas

Al mismo tiempo, juntamente con el director de tesis (tutor), se estimaron fechas para presentar avances del desarrollo del plugins. Esta fase se complementa con un plan de iteraciones que determina la prioridad de cada historia de usuario.

3.1.4.4. Plan de iteraciones

Una vez realizado el análisis de prioridades y la estimación del tiempo en días laborables de las cuatro Historias de Usuarios, se procedió a la planificación de entrega de avances; es decir, un plan de iteraciones basado en el acuerdo de demostraciones sucesivas del avance del producto, en donde cada iteración involucra el desarrollo de historias de usuarios categorizados por el nivel de prioridad.

Iteración 01

En la primera iteración se procedió a definir las tareas de ingeniería para desarrollar las historias de usuarios 01: Conectar con la base de datos, y 02: Visualizar el modelo estructural. Como lo detallan sus nombres, primeramente se realizó una interfaz gráfica para ingresar y establecer los datos de conexión de la base de datos a emplear en un proyecto de Java; por ende, también se desarrolló un editor que permita visualizar el diagrama del modelo relacional de la base de datos. Se presentó un primer avance cuando estos requerimientos estuvieron listos.

Iteración 02

En la segunda iteración se presentaron los requisitos corregidos y mejorados de la primera iteración, juntamente con los requerimientos establecidos de la historia de usuario 03: Crear métodos con sentencias SQL. Para esto, se desarrolló una interfaz gráfica que abrevie los pasos para realizar el procedimiento de selección, inserción, actualización y eliminación de datos a través de sentencias SQL, esta interfaz gráfica deja almacenada la sentencia sql definida por el programador en un archivo XML.

Iteración 03

Finalmente en la tercera iteración se presentó el requerimiento de la historia de usuario 04: Ejecutar los métodos dentro del proyecto de Java. Donde a través de la incorporación de un archivo JAR elaborado, se podrá implementar un reducido código fuente para realizar los métodos de acceso a datos.

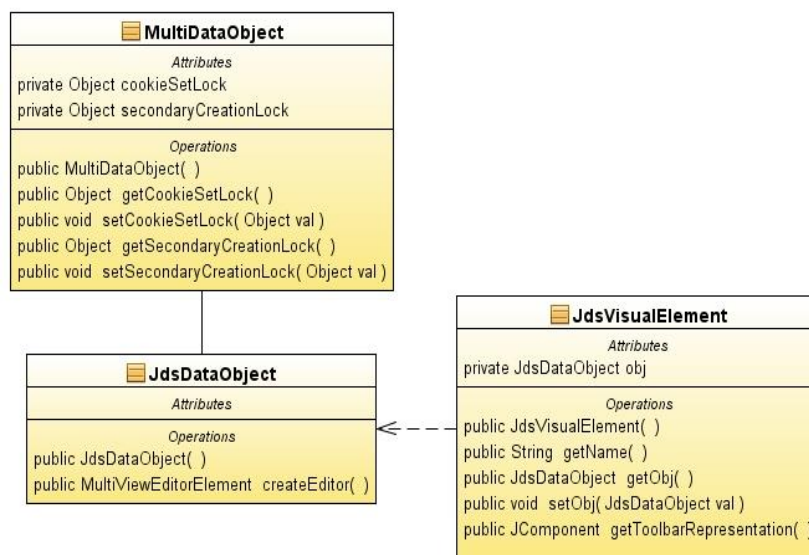
Todo este plan de iteraciones fue expuesto al tutor de tesis, presentando en cada iteración las correcciones observadas y mejoras realizadas.

3.1.4.5. Diagramas de Clases

Entrando en el desarrollo de la codificación del plugins, se realizó un nuevo proyecto tipo Module en la categoría Netbeans Modules del Netbeans IDE, se utilizaron las APIs de Netbeans para reducir esfuerzos. El uso de cada API se dividió por paquetes para mantener un orden y mejor comprensión de la vista del proyecto.

Paquete org.netbeans.dataset1.typefile

ILUSTRACIÓN 12.- DIAGRAMA DE CLASES DEL PAQUETE TYPEFILE

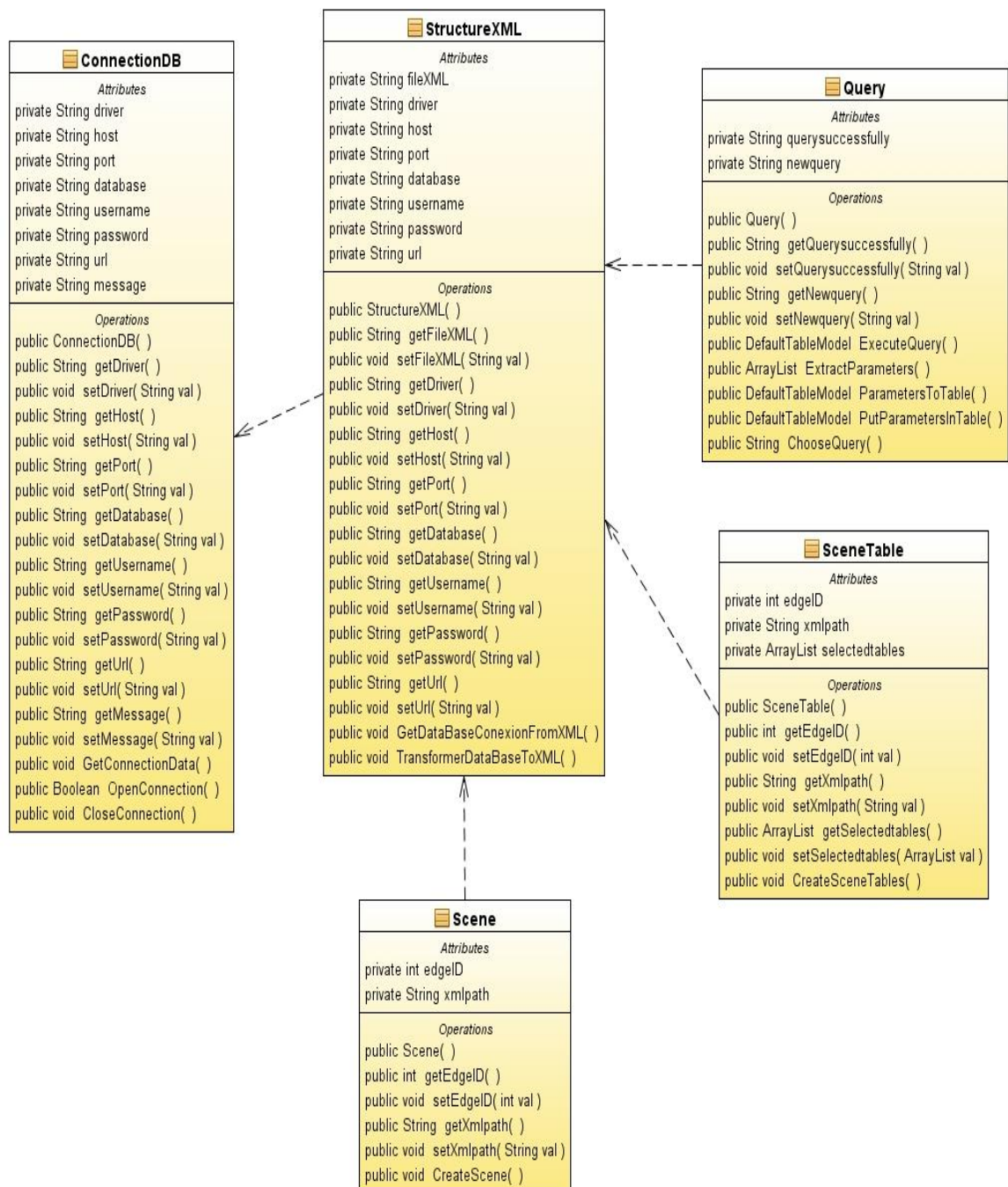


En este paquete se utiliza un API de desarrollo de módulo denominado File Type, esta API proporciona la creación de un nuevo tipo de archivo, se le puso como extensión JDS. Por defecto se crea una clase JdsDataObject que hereda

a la clase MultiDataObject del API del sistema de archivos. También se crea por defecto una clase JdsVisualElement que hereda las propiedades de un JPanel e implementa la clase MultiViewElement.

Paquete org.netbeans.dataset1.dbstructure

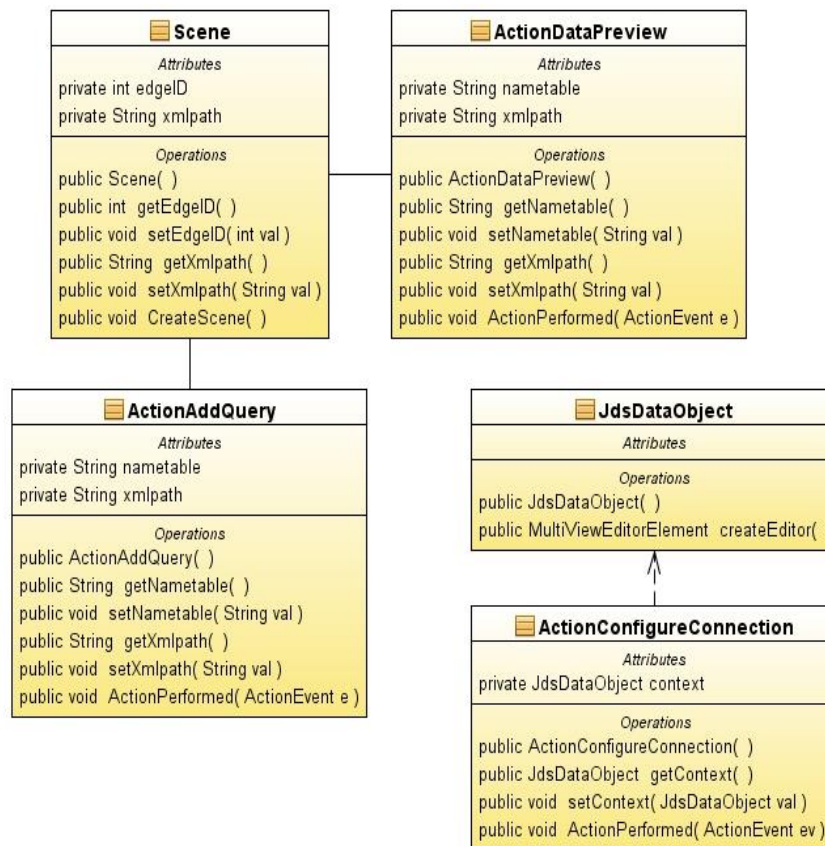
ILUSTRACIÓN 13.- DIAGRAMA DE CLASES DEL PAQUETE DBSTRUCTURE



En este paquete se realizaron las clases que dan funcionalidades específicas al JDataSet. La clase ConnectionDB realiza la conexión a la base de datos, la clase StructureXML realiza las operaciones de almacenamiento de datos en un archivo XML, la clase Query realiza las validaciones de sentencias SQL, la clase Scene es la encargada de cargar el diagrama del modelo relacional de la base de datos en la ventana Visual del archivo JDS proporcionando acciones de acceso a datos. Y la clase SceneTable solo carga las tablas que se requieren añadir al panel del constructor de sentencias.

Paquete org.netbeans.dataset1.actions

ILUSTRACIÓN 14.- DIAGRAMA DE CLASES DEL PAQUETE ACTIONS

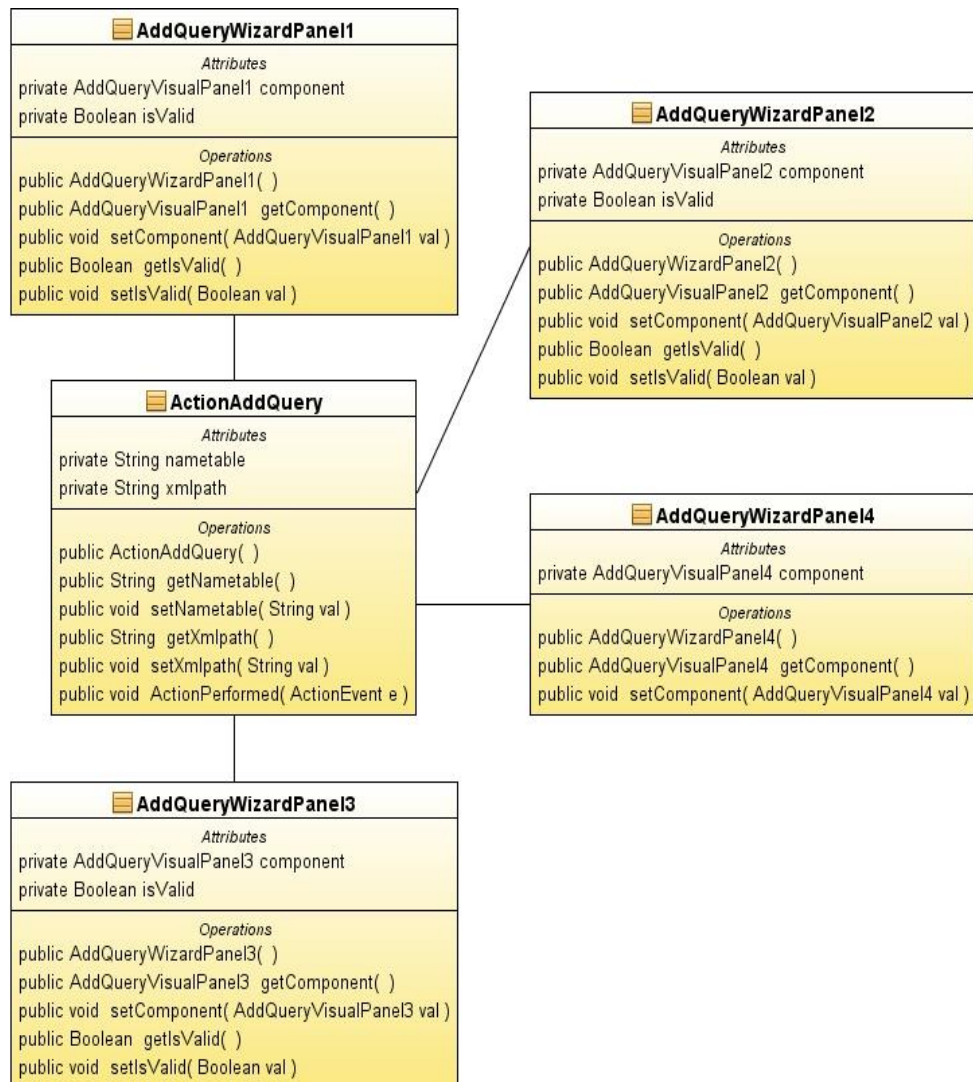


En este paquete se crearon las clases que proporcionan las acciones que suministran el llamado para la presentación de formularios. Las clases ActionAddQuery y ActionDataPreview están contenidas dentro de la clase Scene, donde el usuario puede realizar las acciones para llamar al formulario que mejor requiera.

La clase ActionConfigureConnection es un tipo de acción diferente, esta clase realiza la adición de una nueva acción, a la lista de acciones que vienen por defecto en el tipo de archivo JDS recién creado; esta acción realiza el llamado de un formulario haciendo clic derecho sobre el nodo del archivo JDS.

Paquete org.netbeans.dataset1.wizards

ILUSTRACIÓN 15.- DIAGRAMA DE CLASES DEL PAQUETE WIZARDS



En este paquete se añade el API para el desarrollo de módulo denominado Wizard, este proporciona la creación de un wizard. El Wizard es un conjunto de varios paneles, en cada panel se diseña una parte del requerimiento total, para que al final se logre un objetivo en común. Este conjunto de paneles permite abreviar un proceso largo por medio de una secuencia de pasos a seguir. A

este wizard se le dio el nombre de AddQuery con una cantidad de cuatro paneles, de ahí es donde se genera la distinción en los números. Este conjunto de paneles es llamado desde la clase acción ActionAddQuery.

Librería de clases jdataset-lib.jar

Aparte de la elaboración del plugins, también se elaboró una pequeña librería de clases (Java Class Library), este fue desarrollado en la tercera o última iteración del Plan de Iteraciones del proyecto, basado en la Historia de Usuario 04 la cual tiene como requerimiento: Ejecutar los métodos dentro del proyecto de Java. La incorporación de este archivo JAR a un proyecto de Java en desarrollo proporcionaría la utilización de un código fuente reducido para realizar el llamado de la funciones generadas en el editor del JDataSet que quedan almacenadas en el archivo XML del componente. El diagrama de clases del archivo JAR (jdataset-lib) consta de la siguiente manera.

ILUSTRACIÓN 16.- DIAGRAMA DE CLASES DEL JDATASET-LIB.JAR



3.1.4.6. Diagramas de Secuencia

Los diagramas de secuencia muestran las interacciones entre el usuario (programador) y el software, permitiendo visualizar las operaciones internas que realizan los objetos del software. Cada diagrama de secuencia se desarrolla a partir de una historia de usuario.

ILUSTRACIÓN 17.- DIAGRAMA DE SECUENCIA - CONECTAR CON LA BD

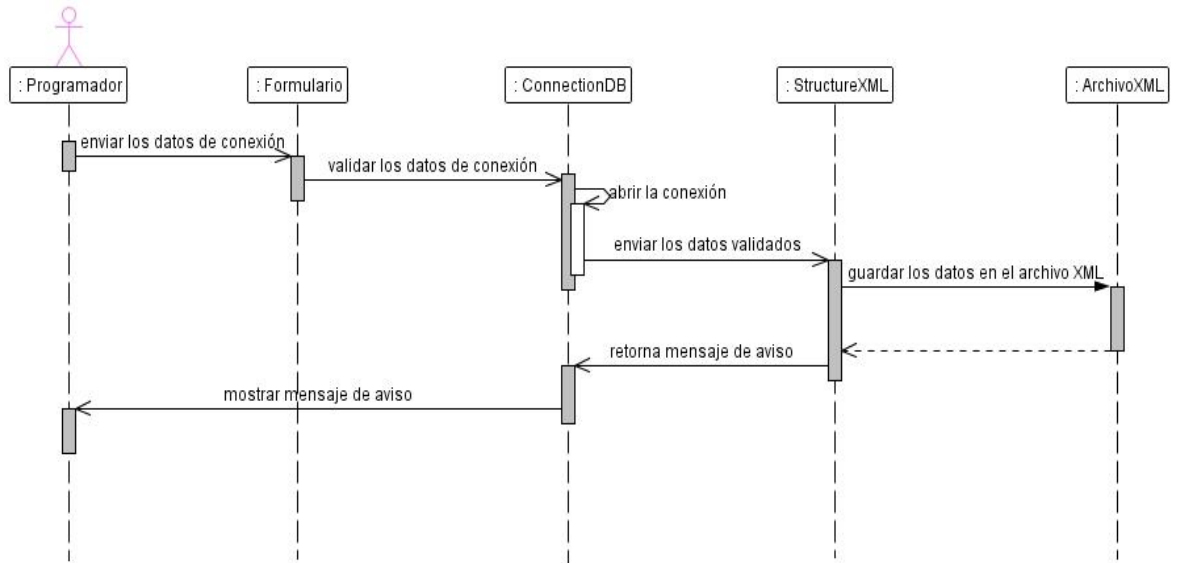


ILUSTRACIÓN 18.- DIAGRAMA DE SECUENCIA - VISUALIZAR EL DIAGRAMA DE LA BD

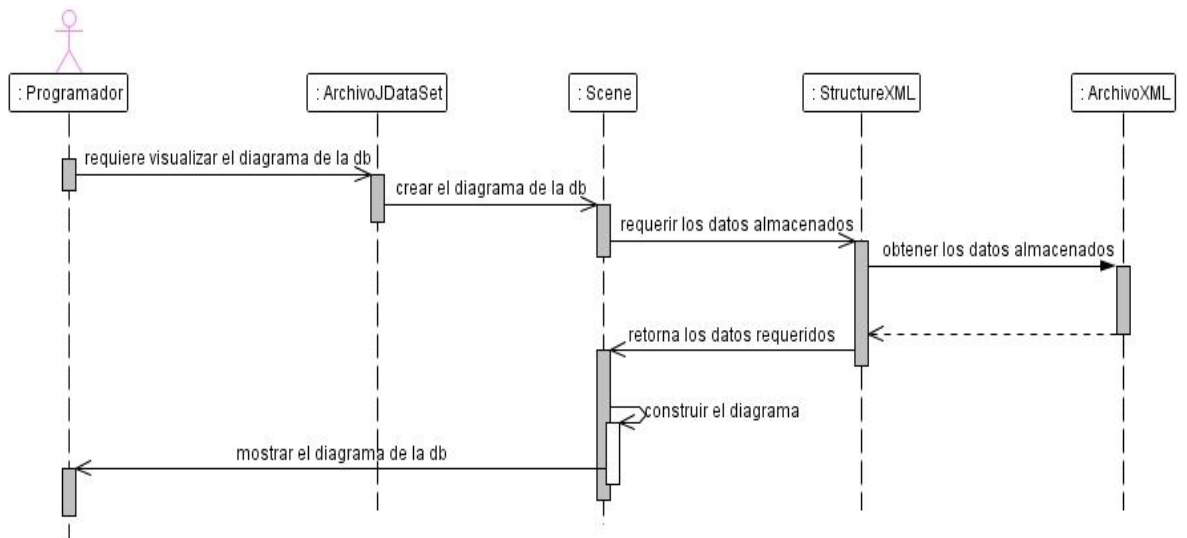


ILUSTRACIÓN 19.- DIAGRAMA DE SECUENCIA - CREAR MÉTODOS CON SENTENCIAS SQL

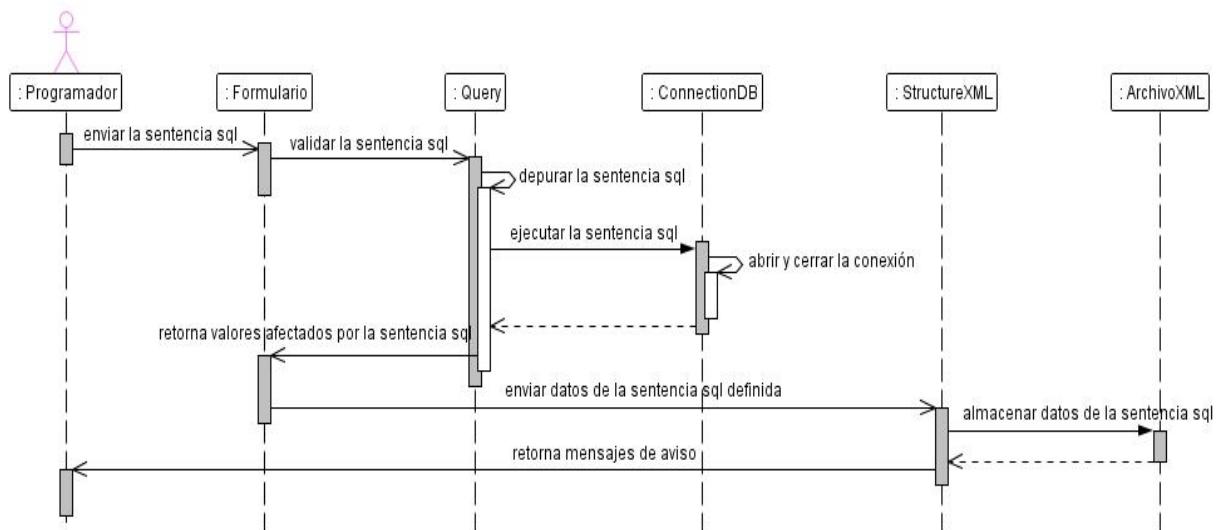
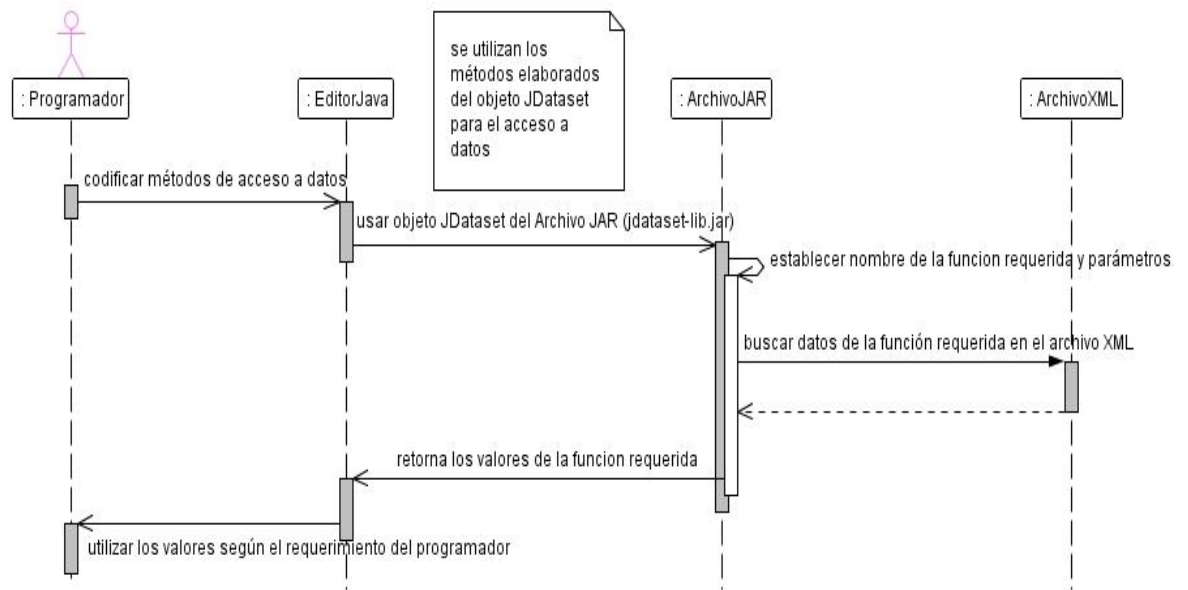


ILUSTRACIÓN 20.- DIAGRAMA DE SECUENCIA - EJECUTAR MÉTODOS DENTRO DEL PROYECTO JAVA



3.1.4.7. Persistencia de datos

Como ya se ha mencionado anteriormente, la creación de un módulo de Netbeans o plugins, consiste en desarrollar un componente de asistencia gráfica que permita el manejo de datos dentro de Netbeans IDE, en donde el programador pueda manipular los objetos de la base de datos por un medio de una interfaz gráfica.

Para esto es indispensable que internamente queden almacenados los datos que otorguen el funcionamiento del componente JDataSet, estos son: los datos de conexión, los nombres de tablas con sus campos, las relaciones y otros objetos importantes de la base de datos. De esa manera, se determinó el uso de archivos XML para el almacenamiento de estos datos.

Implementación de documentos XML

Para el almacenamiento de la información se optó por la implementación de archivos XML, el cual es de reconocimiento a nivel mundial en la comunidad informática. Los elementos que conforman la estructura de este archivo XML están establecidos de la siguiente manera.

ILUSTRACIÓN 21.- ELEMENTO RAÍZ DEL ARCHIVO XML

```
<database drivername="com.mysql.jdbc.Driver" host="localhost"
        namedb="farmacia" password="1234" port="3306"
        url="jdbc:mysql://localhost/farmacia" username="root">
    <table>
    <table>
    <table>
    <table>
</database>
```

La etiqueta *<database>* describe los principales datos de conexión con la base de datos. Este elemento se complementa con los atributos *drivername*, *host*, *namedb*, *password*, *port*, *url* y *username*.

En estos atributos se define el driver de conexión del motor de base de datos, el nombre del servidor, el puerto del servidor, nombre de usuario y password del motor de base datos, nombre de la base de datos, y la url o cadena de conexión a la base de datos.

El elemento raíz *<database>* se compone por nodos hijos, estableciendo elementos *<table>* donde se especifican los elementos que apropián a una tabla de la base de datos.

ILUSTRACIÓN 22.- OTROS ELEMENTOS DEL ARCHIVO XML

```
<table name="cliente">
  <column type="INT">idcliente</column>
  <column type="VARCHAR">cedula</column>
  <column type="VARCHAR">nombre</column>
  <column type="VARCHAR">apellido</column>
  <relation>
    <foreingkey>factura:idcliente</foreingkey>
    <primarykey>cliente:idcliente</primarykey>
  </relation>
  <function name="GetData()">select *from cliente</function>
  <function name="InsertarCliente()">INSERT INTO
    cliente (cedula, nombre, apellido) VALUES
    (@cedula, @nombre, @apellido)
  </function>
</table>
```

El elemento `<table>` contiene un atributo *name* el cual especifica el nombre de la tabla. Este nodo `<table>` contiene nodos hijos `<column>`, `<relation>`, y `<function>`.

En el nodo `<column>` se usa el atributo *type* para especificar el tipo de dato de la columna, y en su contenido se pone el nombre de la columna o campo de la tabla.

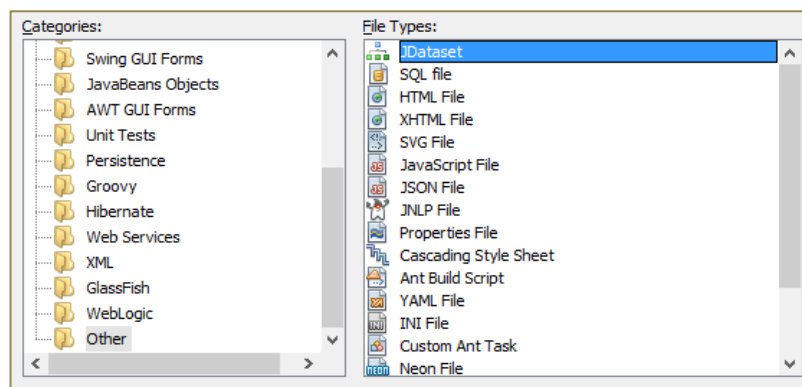
El nodo `<relation>` se usa para especificar los campos que estén vinculados a relaciones con otras tablas, este nodo está compuesto por dos nodos hijos. El primer nodo *foreingkey* detalla la clave foránea, es decir, especifica la tabla foránea con su respectivo campo. El segundo nodo *primarykey* especifica la tabla anfitrión con el campo al cual se establece la relación respectiva.

Y el último nodo `<function>` especifica las funciones agregadas que el usuario ha realizado en el editor del JDataSet. Este elemento está compuesto por el atributo *name* para especificar el nombre de la función, y en su contenido se especifica la sentencia SQL establecida.

3.1.4.8. Interfaces de usuario

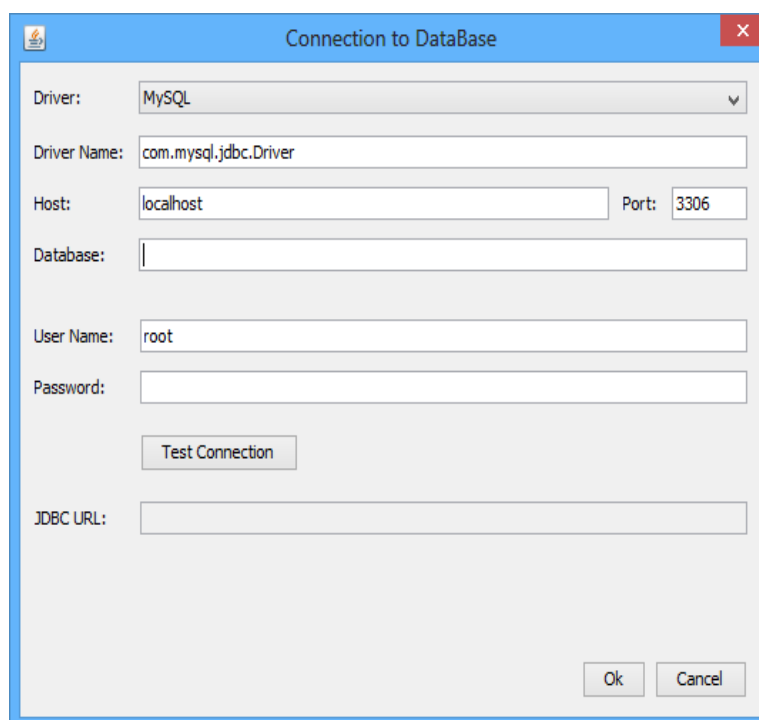
Una vez que el plugin JDataSet sea instalado en Netbeans IDE, se presentan las siguientes interfaces gráficas.

ILUSTRACIÓN 23.- CUADRO DE DIÁLOGO NEW FILE



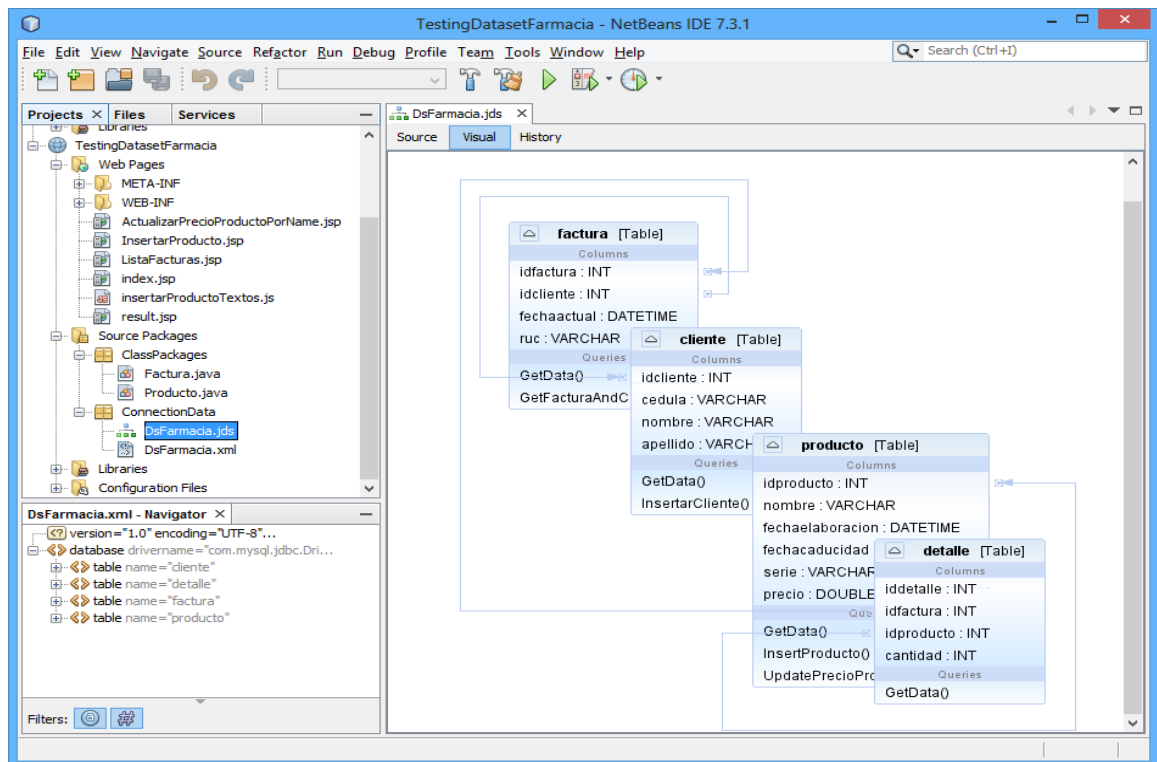
Gracias a la asistencia gráfica que existe en el entorno de desarrollo de Netbeans Modules, es posible construir automáticamente un nuevo tipo de archivo, y agregarlo al cuadro de diálogo New File de Netbeans. De esa manera, se realizó la construcción de un nuevo tipo archivo con extensión JDS, denominado JDataSet, tal y como presenta la ilustración previa.

ILUSTRACIÓN 24.- FORMULARIO CONFIGURAR CONEXIÓN



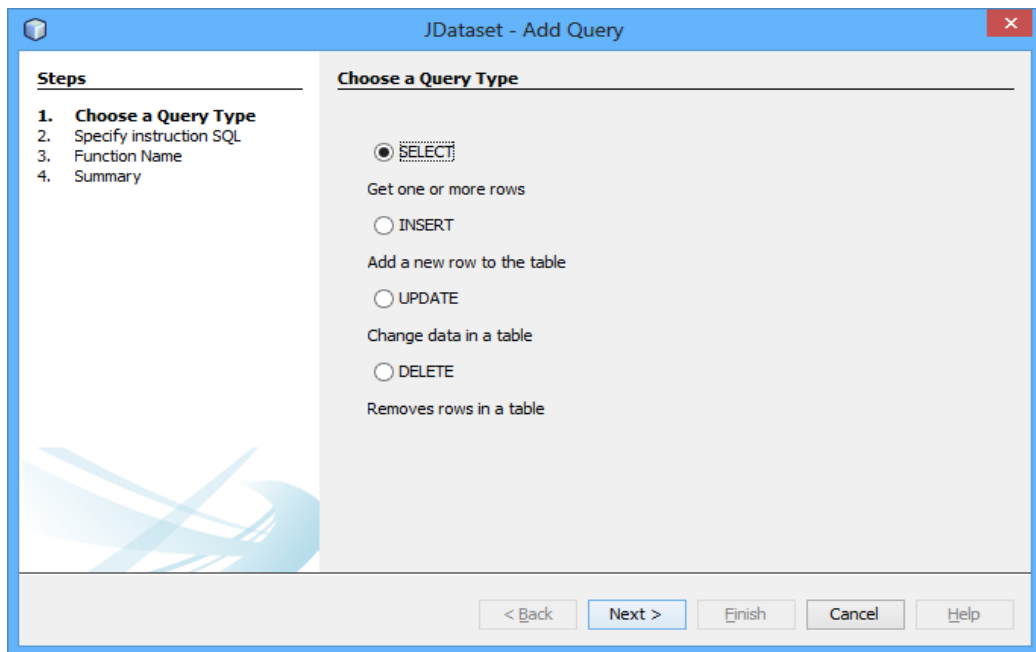
Como primer requerimiento por parte del usuario (programador), se necesita una interfaz gráfica que permita realizar fácilmente la configuración de una nueva conexión de base de datos.

ILUSTRACIÓN 25.- VENTANA EDITOR DEL JDATASET



Una vez que se añade el componente JDataSet, un nuevo tipo de archivo con extensión JDS es agregado a un proyecto en Java en desarrollo. Después de configurar una nueva conexión a una base de datos, nos muestra una ventana editor la cual consta de tres pestañas: Source, Visual e History. Pero solo la pestaña Visual permite trabajar en las tareas del componente JDataSet.

ILUSTRACIÓN 26.- WIZARD PARA AGREGAR SENTENCIAS SQL

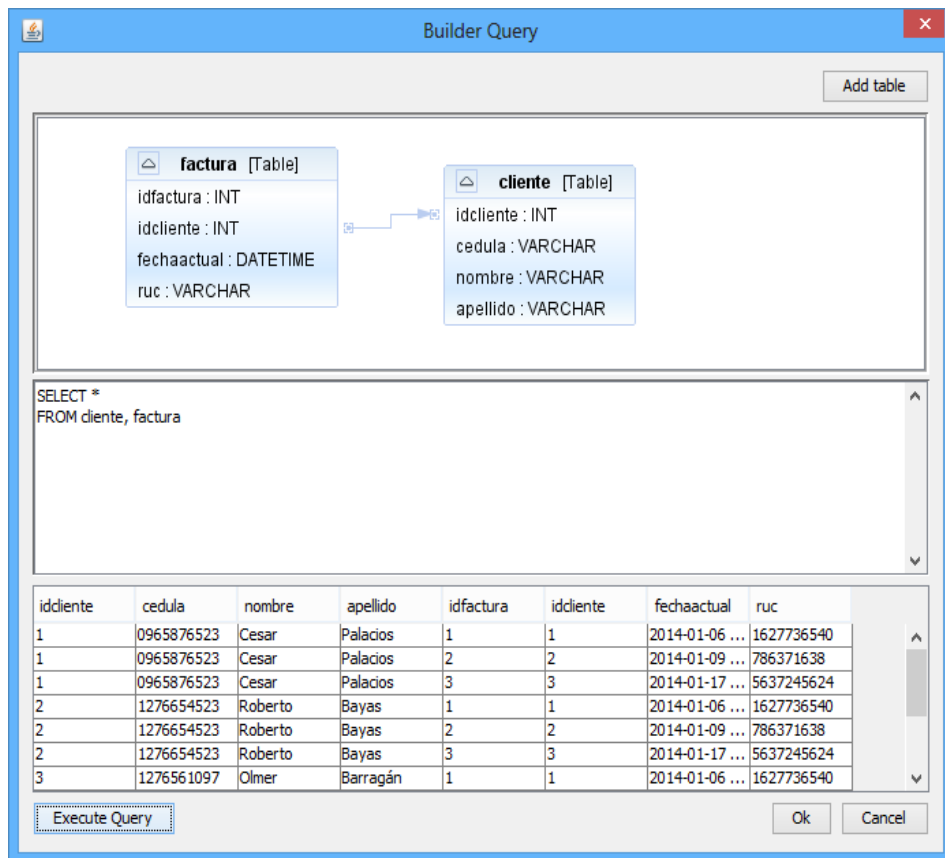


La asistencia gráfica de Netbeans también permite la construcción de Wizards. Este wizard permite generar cualquier procedimiento a través de la secuencia de pasos a seguir en diferentes páginas.

En la primera página del wizard solicita que se seleccione el tipo de sentencia que se desea efectuar. En la segunda página pide que se arme definitivamente la sentencia SQL con todos sus detalles. En la tercera página pide que se ingrese un nombre para reconocimiento de la función que se ha realizado previamente. Y en la última página muestra un resumen de todos los pasos establecidos anteriormente.

La sentencia SQL queda almacenada en un archivo XML visualizado por el usuario (programador) en el IDE. La sentencia se registra en el XML juntamente con el nombre de la función ingresado en este wizard. Esta función se podrá llamar por medio de un código fuente reducido desde la aplicación Java en desarrollo, utilizando el archivo JAR `jdataset-lib.jar` que contiene métodos ya elaborados para el llamado de estas funciones.

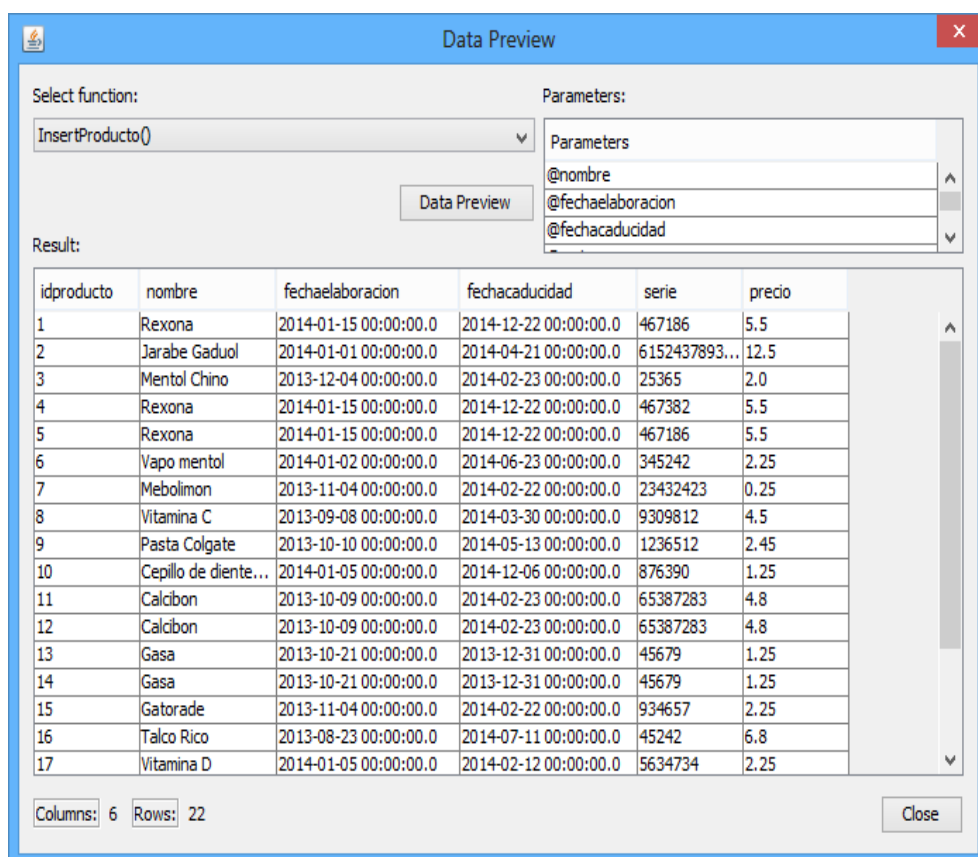
ILUSTRACIÓN 27.- FORMULARIO CONSTRUCTOR DE SENTENCIAS



Para armar, probar, y definir una sentencia SQL, se desarrolló una interfaz que permita ejecutar estas acciones. En la segunda página del wizard pide definir una sentencia SQL, para lo cual se da la opción de que el programador acceda a este formulario constructor de sentencias SQL.

También se permite el uso de parámetros antecidos del prefijo @ con el fin de establecer sentencias SQL de uso dinámico, permitiendo que los valores de los parámetros sean diferentes al momento de ejecutarlos, similar a la definición de procedimientos almacenados con parámetros.

ILUSTRACIÓN 28.- FORMULARIO DE REVISIÓN DE DATOS



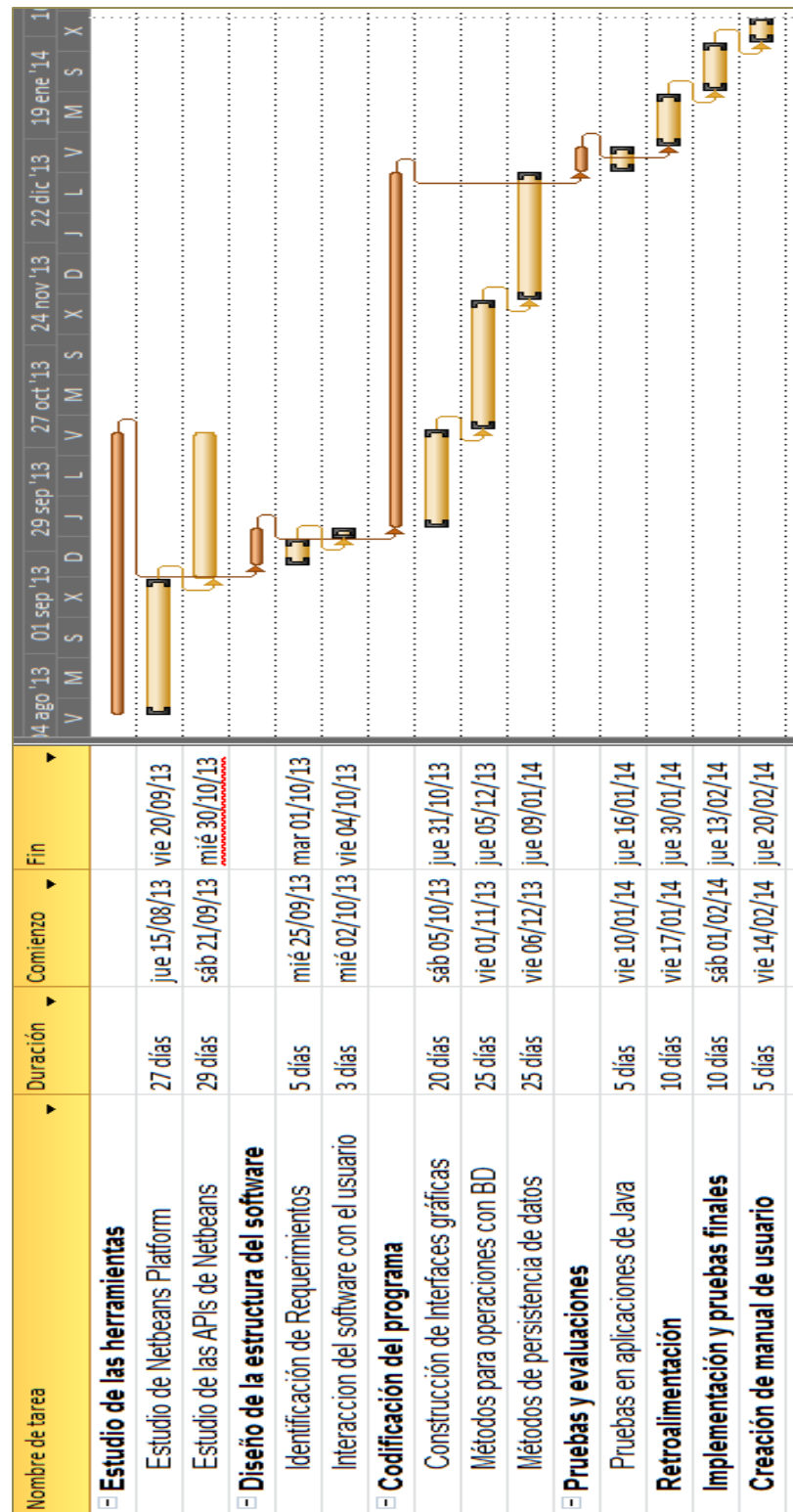
The screenshot shows a 'Data Preview' window with a blue title bar. At the top left, there is a 'Select function:' dropdown menu with 'InsertProducto()' selected. To its right is a 'Parameters:' section with a list box containing '@nombre', '@fechaelaboracion', and '@fechacaducidad'. Below these is a 'Data Preview' button. The main area is labeled 'Result:' and contains a table with 6 columns: 'idproducto', 'nombre', 'fechaelaboracion', 'fechacaducidad', 'serie', and 'precio'. The table lists 17 products. At the bottom left, it shows 'Columns: 6' and 'Rows: 22'. A 'Close' button is at the bottom right.

idproducto	nombre	fechaelaboracion	fechacaducidad	serie	precio
1	Rexona	2014-01-15 00:00:00.0	2014-12-22 00:00:00.0	467186	5.5
2	Jarabe Gaduol	2014-01-01 00:00:00.0	2014-04-21 00:00:00.0	6152437893...	12.5
3	Mentol Chino	2013-12-04 00:00:00.0	2014-02-23 00:00:00.0	25365	2.0
4	Rexona	2014-01-15 00:00:00.0	2014-12-22 00:00:00.0	467382	5.5
5	Rexona	2014-01-15 00:00:00.0	2014-12-22 00:00:00.0	467186	5.5
6	Vapo mentol	2014-01-02 00:00:00.0	2014-06-23 00:00:00.0	345242	2.25
7	Mebolimon	2013-11-04 00:00:00.0	2014-02-22 00:00:00.0	23432423	0.25
8	Vitamina C	2013-09-08 00:00:00.0	2014-03-30 00:00:00.0	9309812	4.5
9	Pasta Colgate	2013-10-10 00:00:00.0	2014-05-13 00:00:00.0	1236512	2.45
10	Cepillo de diente...	2014-01-05 00:00:00.0	2014-12-06 00:00:00.0	876390	1.25
11	Calcibon	2013-10-09 00:00:00.0	2014-02-23 00:00:00.0	65387283	4.8
12	Calcibon	2013-10-09 00:00:00.0	2014-02-23 00:00:00.0	65387283	4.8
13	Gasa	2013-10-21 00:00:00.0	2013-12-31 00:00:00.0	45679	1.25
14	Gasa	2013-10-21 00:00:00.0	2013-12-31 00:00:00.0	45679	1.25
15	Gatorade	2013-11-04 00:00:00.0	2014-02-22 00:00:00.0	934657	2.25
16	Talco Rico	2013-08-23 00:00:00.0	2014-07-11 00:00:00.0	45242	6.8
17	Vitamina D	2014-01-05 00:00:00.0	2014-02-12 00:00:00.0	5634734	2.25

Finalmente el usuario (programador) podrá visualizar todas las sentencias SQL que fueron realizadas en el wizard para agregar sentencias del JDataSet, y que están almacenadas en el archivo XML. Aquí se muestra una lista desplegable con todas las funciones, sus parámetros en caso de que los tenga; y también, se puede ejecutar la función para la revisión y visualización de las consultas de datos.

3.1.4.9. Cronograma

ILUSTRACIÓN 29.- DIAGRAMA DE GANTT



3.2. TIPO DE INVESTIGACIÓN

Para establecer fundamentos que confirmen la teoría planteada, se tendría que realizar un contacto directo con el objeto de estudio. En este caso, para la búsqueda de información se tuvo que llegar hasta los programadores de software, es así como se optó por realizar una investigación de campo.

3.2.1. Investigación de Campo

La investigación de campo permite la observación directa con las personas que sirven de estudio para confirmar la problemática a nivel de la comunidad de programadores de software, y a la vez para establecer propuestas directas para lograr una solución concreta.

Esta investigación de campo centra las acciones de recoger datos en el área temática en que se está tratando, para su posterior análisis.

El área de estudio fue constituida por los estudiantes de la carrera de Sistemas de la facultad de Ciencias de la Ingeniería de la UTEQ, ya que es una área en el cual existe libre disponibilidad para acceder a las personas, y por distancia más cercana.

3.2.2. Recolección de datos

Para la recolección de datos se optó por la elaboración de encuestas. En esta encuesta se formulan siete enunciados basados en la problemática de la presente investigación con el fin de confirmarla, y basado en la hipótesis y objetivos planteados para proponer soluciones.

Elaboración de la Encuesta

La encuesta fue realizada para evaluar la satisfacción de los programadores de software, en cuanto al procedimiento de acceso a datos cuando desarrollan aplicaciones. Dirigida a los estudiantes del segundo, tercero y cuarto año de la carrera de Sistemas de la FCI de la UTEQ.

Se realizaron siete proposiciones, cada uno con un objetivo claro y conciso. Cada proposición tenía que ser contestada seleccionando un nivel de acuerdo o desacuerdo, según el enunciado leído. Se pidió total honestidad en base a la experiencia obtenida a lo largo de sus estudios universitarios. A continuación las proposiciones contenidas en la encuesta, con sus respectivos objetivos y opciones de respuesta.

Primera proposición

Objetivo de la 1ra proposición.- Verificar la confianza y seguridad que tienen los programadores al emplear solo código fuente para realizar las operaciones transaccionales hacia la bases de datos.

1ra proposición.- El programador obtiene completa confianza y seguridad al emplear solamente código fuente para realizar las operaciones transaccionales hacia la base de datos.

- Totalmente de acuerdo
- De acuerdo
- Ni en acuerdo, ni en desacuerdo
- En desacuerdo
- Totalmente en desacuerdo

Segunda proposición

Objetivo de la 2da proposición.- Saber si el proceso de codificar las operaciones transaccionales hacia la base de datos provoca monotonía y complejidad en el desarrollo de aplicaciones.

2da proposición.- Codificar los métodos o funciones para la selección, inserción, actualización y eliminación de datos, provocan que el desarrollo de la aplicación sea monótona, repetitiva y a la vez compleja.

- Totalmente de acuerdo
- De acuerdo
- Ni en acuerdo, ni en desacuerdo
- En desacuerdo
- Totalmente en desacuerdo

Tercera proposición

Objetivo de la 3ra proposición.- Verificar si los componentes gráficos de asistencia al programador ayudan en la construcción rápida de aplicaciones.

3ra proposición.- Los componentes gráficos que ayudan al programador a desarrollar aplicaciones, permiten que la construcción del software sea más sencilla y rápida.

- Totalmente de acuerdo
- De acuerdo
- Ni en acuerdo, ni en desacuerdo
- En desacuerdo
- Totalmente en desacuerdo

Cuarta proposición

Objetivo de la 4ta proposición.- Conocer si a los programadores les resulta más fácil y rápido realizar las operaciones de acceso a datos por medio de componentes gráficos especializados en el acceso a datos.

4ta proposición.- El desarrollo de los métodos de selección, inserción, actualización y eliminación de datos sería más fácil y rápido si se realizara con la ayuda de componentes de asistencia gráfica.

- Totalmente de acuerdo
- De acuerdo
- Ni en acuerdo, ni en desacuerdo
- En desacuerdo
- Totalmente en desacuerdo

Quinta proposición

Objetivo de la 5ta proposición.- Conocer si a los programadores les resulta más fácil realizar las operaciones hacia la base de datos si visualizan desde el IDE el modelo relacional de la base de datos.

5ta proposición.- El procedimiento para realizar las operaciones con la base de datos sería más fácil si se pudiese visualizar desde el IDE (entorno de desarrollo integrado) el diagrama del modelo relacional de la base de datos.

- Totalmente de acuerdo
- De acuerdo
- Ni en acuerdo, ni en desacuerdo
- En desacuerdo
- Totalmente en desacuerdo

Sexta proposición

Objetivo de la 6ta proposición.- Verificar la comodidad y facilidad obtenida por los programadores al emplear componentes gráficos de ayuda para realizar las operaciones con bases de datos.

6ta proposición.- Emplear componentes gráficos especializados en las operaciones transaccionales con base de datos infunden total comodidad y facilidad al desarrollar aplicaciones.

- Totalmente de acuerdo
- De acuerdo
- Ni en acuerdo, ni en desacuerdo
- En desacuerdo
- Totalmente en desacuerdo

Séptima proposición

Objetivo de la 7ma proposición.- Saber si el programador desea mantener un equilibrio entre la implementación de código fuente y componentes de asistencia gráfica.

7ma proposición.- La implementación de código fuente y componentes de asistencia gráfica debe ser equilibrada al momento de desarrollar aplicaciones.

- Totalmente de acuerdo
- De acuerdo
- Ni en acuerdo, ni en desacuerdo
- En desacuerdo
- Totalmente en desacuerdo

3.2.3. Presentación de datos

Una vez recopilada la información se procedió a organizar los datos en tablas y figuras para una mejor comprensión de los mismos. Primeramente se muestran los resultados globales según los estudiantes encuestados por cada semestre, para luego presentar los resultados por cada proposición con sus respectivas cantidades, figuras y su respectivo análisis.

TABLA 11.- RESPUESTAS DE ENCUESTADOS DEL 4TO SEMESTRE (2DO AÑO)

Según las proposiciones	Totalmente de acuerdo	De acuerdo	Ni en acuerdo, ni en desacuerdo	En desacuerdo	Totalmente en desacuerdo	Cantidad de encuestados
Primera proposición	10	7	2	0	0	19
Segunda proposición	4	8	3	4	0	19
Tercera proposición	7	9	2	1	0	19
Cuarta proposición	3	10	2	3	1	19
Quinta proposición	4	11	3	1	0	19
Sexta proposición	2	9	7	1	0	19
Séptima proposición	6	9	2	1	1	19

En el 4to semestre (2do año) de la carrera de sistemas se encuestó a un total de 19 estudiantes, proporcionando resultados favorables a la consecución del presente proyecto de tesis.

TABLA 12.- RESPUESTAS DE ENCUESTADOS DEL 6TO SEMESTRE (3ER AÑO)

Según las proposiciones	Totalmente de acuerdo	De acuerdo	Ni en acuerdo, ni en desacuerdo	En desacuerdo	Totalmente en desacuerdo	Cantidad de encuestados
Primera proposición	3	2	0	0	0	5
Segunda proposición	2	2	0	1	0	5
Tercera proposición	3	1	1	0	0	5
Cuarta proposición	4	1	0	0	0	5
Quinta	4	1	0	0	0	5

proposición						
Sexta proposición	2	3	0	0	0	5
Séptima proposición	2	3	0	0	0	5

En el 6to semestre (3er año) de la carrera de sistemas se encuestó a un total de 5 estudiantes, en este caso no se podrían asegurar resultados puntuales, debido a la reducida cantidad de estudiantes que se encuestó.

TABLA 13.- RESPUESTAS DE ENCUESTADOS DEL 8VO SEMESTRE (4TO AÑO)

Según las proposiciones	Totalmente de acuerdo	De acuerdo	Ni en acuerdo, ni en desacuerdo	En desacuerdo	Totalmente en desacuerdo	Cantidad de encuestados
Primera proposición	1	8	7	1	0	17
Segunda proposición	4	9	2	1	1	17
Tercera proposición	10	5	2	0	0	17
Cuarta proposición	10	6	1	0	0	17
Quinta proposición	7	7	3	0	0	17
Sexta proposición	4	12	1	0	0	17
Séptima proposición	6	7	4	0	0	17

En el 8vo semestre (4to año) de la carrera de sistemas se encuestó a un total de 17 estudiantes, dando resultados más favorables y permitiendo percibir coherencia a las propuestas planteadas en el presente proyecto.

En resumen, en el 4to semestre (2do año) se encuestaron a 19 estudiantes, en el 6to semestre (3er año) se encuestaron a 5 estudiantes, y en el 8vo semestre (4to año) se encuestaron a 17 estudiantes. Dando un total de **41** estudiantes de la carrera de Ingeniería en Sistemas que fueron encuestados.

Ahora se presentan los datos globales de los resultados obtenidos por cada proposición de las encuestas realizadas, con sus respectivos porcentajes, figuras y análisis.

Primera proposición.- El programador obtiene completa confianza y seguridad al emplear solamente código fuente para realizar las operaciones transaccionales hacia la base de datos.

ILUSTRACIÓN 30.- PORCENTAJES DE LA PRIMERA PROPOSICIÓN DE LA ENCUESTA

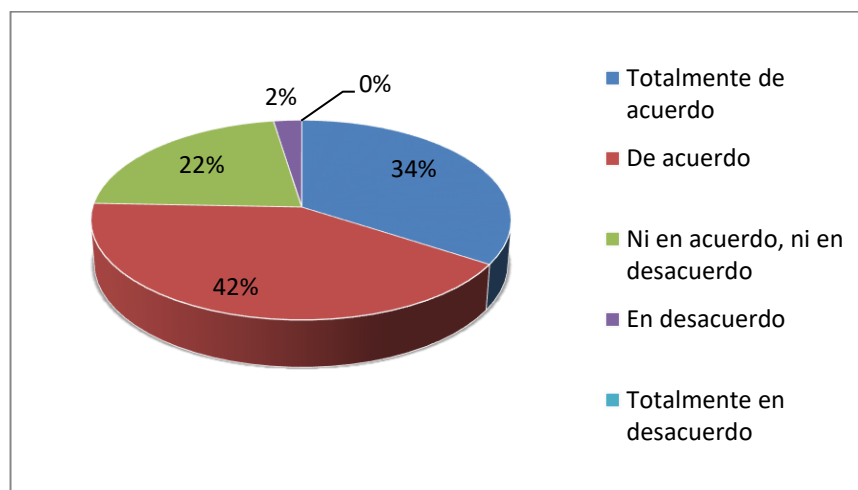


TABLA 14.- RESULTADOS DE LA PRIMERA PROPOSICIÓN DE LA ENCUESTA

RESPUESTAS	RESULTADOS	PORCENTAJE (%)
Totalmente de acuerdo	14	34.14%
De acuerdo	17	41.46%
Ni en acuerdo, ni en desacuerdo	9	21.95%
En desacuerdo	1	2.43%
Totalmente en desacuerdo	0	0%
TOTAL	41	100%

Análisis de la 1ra proposición.- El objetivo de este enunciado es verificar la confianza y seguridad que tienen los programadores de aplicaciones, al emplear código fuente escrito línea por línea para realizar el acceso a datos.

Observando los resultados se nota que es indispensable la escritura de código fuente en el desarrollo de aplicaciones, un total de 31 encuestados están de acuerdo con esta proposición, 9 encuestados tienen una respuesta neutra dando a entender que no obtiene la suficiente confianza cuando emplean código fuente, y 1 solo estudiante está en desacuerdo con esta proposición.

Segunda proposición.- Codificar los métodos o funciones para la selección, inserción, actualización, y eliminación de datos, provocan que el desarrollo de la aplicación sea monótona, repetitiva y la vez compleja.

ILUSTRACIÓN 31.- PORCENTAJES DE LA SEGUNDA PROPOSICIÓN DE LA ENCUESTA

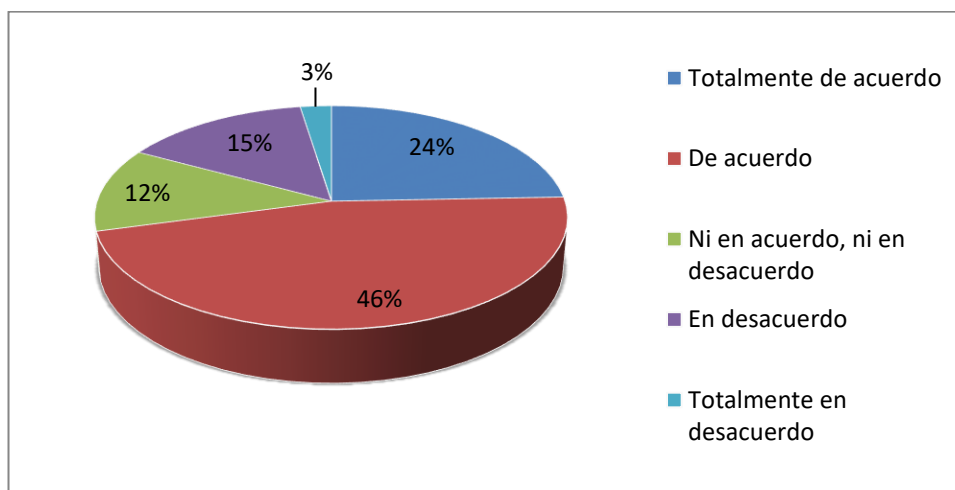


TABLA 15.- RESULTADOS DE LA SEGUNDA PROPOSICIÓN DE LA ENCUESTA

RESPUESTAS	RESULTADOS	PORCENTAJE (%)
Totalmente de acuerdo	10	24.39%
De acuerdo	19	46.34%
Ni en acuerdo, ni en desacuerdo	5	12.19%
En desacuerdo	6	14.63%
Totalmente en desacuerdo	1	2.43%
TOTAL	41	100%

Análisis de la 2da proposición.- El objetivo de este enunciado es saber si el proceso de codificar las operaciones transaccionales hacia la base de datos provoca monotonía, repetitividad y complejidad en el desarrollo. **Observando** los resultados la mayoría de encuestados, 29 en total, están de acuerdo que la codificación de los métodos para el llamado de datos provoca malestar al realizar las tareas por medio de escritura de código. Solo 5 encuestados son neutrales dando a entender que no tienen ni malestar ni facilidad al codificar los métodos de acceso a datos. Y un total de 7 encuestados están en desacuerdo con esta proposición, para estos 7 estudiantes no produce ningún malestar

escribir código fuente para acceder a los datos. En resumen, prevalece la complejidad y monotonía al codificar métodos de acceso a datos en la mayoría de estudiantes.

Tercera proposición.- Los componentes gráficos que ayudan al programador a desarrollar aplicaciones, permiten que la construcción del software sea más sencilla y rápida.

ILUSTRACIÓN 32.- PORCENTAJES DE LA TERCERA PROPOSICIÓN DE LA ENCUESTA

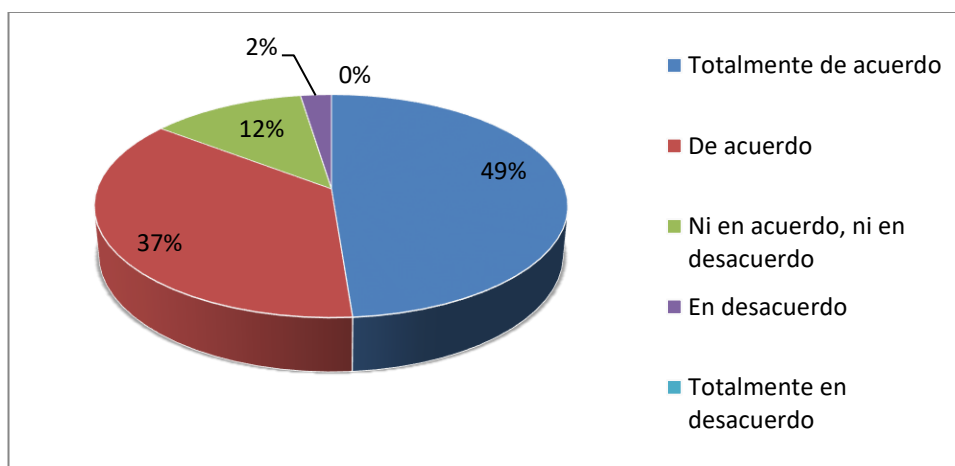


TABLA 16.- RESULTADOS DE LA TERCERA PROPOSICIÓN DE LA ENCUESTA

RESPUESTAS	RESULTADOS	PORCENTAJE (%)
Totalmente de acuerdo	20	48.78%
De acuerdo	15	36.58%
Ni en acuerdo, ni en desacuerdo	5	12.19%
En desacuerdo	1	2.43%
Totalmente en desacuerdo	0	0%
TOTAL	41	100%

Análisis de la 3ra proposición.- El objetivo de esta proposición es verificar si los componentes gráficos de asistencia al programador ayudan en la construcción rápida de aplicaciones. **Observando** los resultados, se puede notar que la mayoría de los encuestados, 35 en total, están de acuerdo en que los componentes gráficos ayudan al programador a realizar un desarrollo más sencillo y rápido del software, dando a notar que esta proporción es

plenamente aceptada. Una cantidad de 5 encuestados son neutrales dando a entender que los componentes gráficos no ayudan con el desarrollo rápido, ni causan lentitud. Y solo 1 encuestado está en desacuerdo con esta proposición. La aceptación de esta proposición en la comunidad informática es mayoritaria, es decir que los componentes gráficos facilitan el desarrollo rápido de aplicaciones.

Cuarta proposición.- El desarrollo de los métodos de selección, inserción, actualización, y eliminación de datos sería más fácil y rápida si se realizara con la ayuda de componentes de asistencia gráfica.

ILUSTRACIÓN 33.- PORCENTAJES DE LA CUARTA PROPOSICIÓN DE LA ENCUESTA

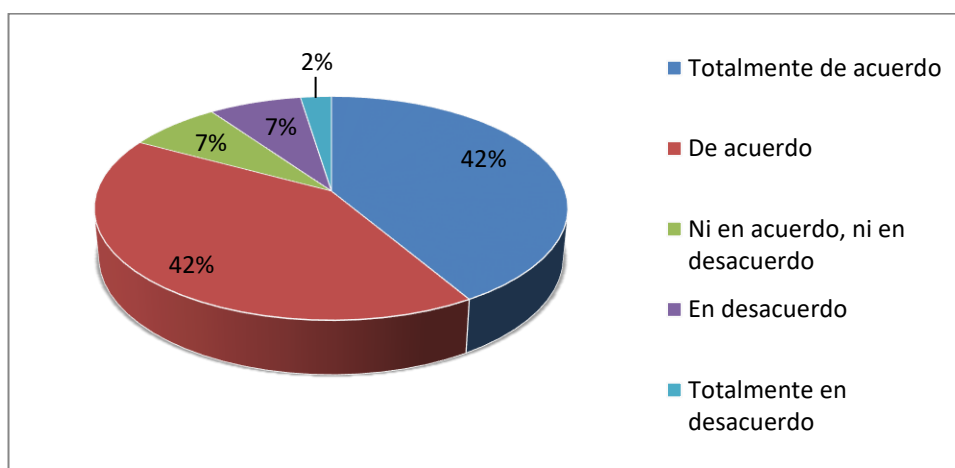


TABLA 17.- RESULTADOS DE LA CUARTA PROPOSICIÓN DE LA ENCUESTA

RESPUESTAS	RESULTADOS	PORCENTAJE (%)
Totalmente de acuerdo	17	41.46%
De acuerdo	17	41.46%
Ni en acuerdo, ni en desacuerdo	3	7.31%
En desacuerdo	3	7.31%
Totalmente en desacuerdo	1	2.43%
TOTAL	41	100%

Análisis de la 4ta proposición.- El objetivo de esta proposición es conocer si a los programadores les resulta más fácil realizar las operaciones de acceso a datos por medio de componentes gráficos de acceso a datos. **Observando** los

resultados suman un total de 34 encuestados que están de acuerdo con esta proposición, confirmando la propuesta de facilitar el procedimiento de acceso a datos por medio de componentes de asistencia gráfica. Una cantidad de 3 estudiantes son neutrales, no están de acuerdo ni en desacuerdo con esta proposición. Y un total de 4 encuestados dice que están en desacuerdo, dando a entender que realizar los métodos de acceso a datos con la asistencia de componentes gráficos no facilita ni acelera este proceso. Como conclusión, la mayoría de encuestados afirman que los componentes gráficos facilitan y aceleran la realización de los métodos para el acceso a datos.

Quinta proposición.- El procedimiento para realizar las operaciones con la base de datos sería más fácil si se pudiera visualizar desde el IDE (entorno de desarrollo integrado) el diagrama del modelo relacional de la base de datos.

ILUSTRACIÓN 34.- PORCENTAJES DE LA QUINTA PROPOSICIÓN DE LA ENCUESTA

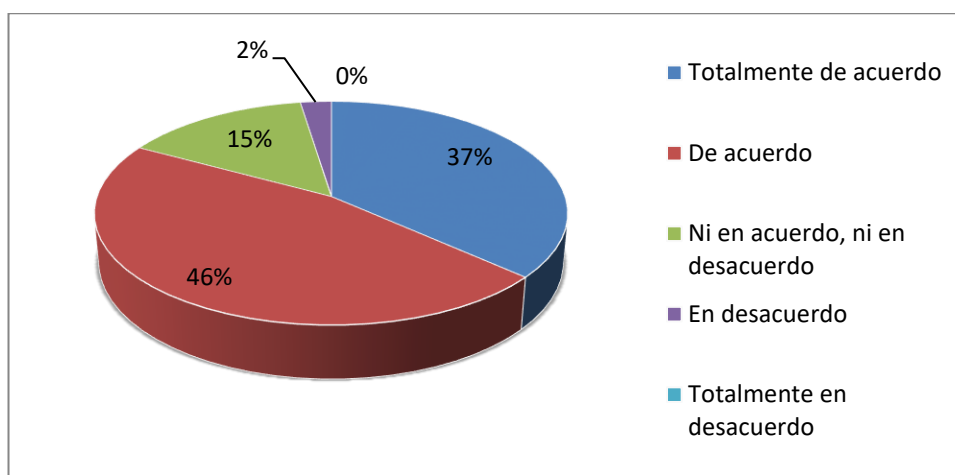


TABLA 18.- RESULTADOS DE LA QUINTA PROPOSICIÓN DE LA ENCUESTA

RESPUESTAS	RESULTADOS	PORCENTAJE (%)
Totalmente de acuerdo	15	36.58%
De acuerdo	19	46.34%
Ni en acuerdo, ni en desacuerdo	6	14.63%
En desacuerdo	1	2.43%
Totalmente en desacuerdo	0	0%
TOTAL	41	100%

Análisis de la 5ta proposición.- El objetivo de esta proposición es conocer si a los programadores les resulta más fácil realizar las operaciones hacia la base de datos si se visualizara desde el IDE el modelo relacional de la base de datos. **Observando** los resultados se tiene que 34 encuestados están de acuerdo en visualizar en el IDE el diagrama del modelo relacional de la base de datos para tener un desarrollo con más facilidad. Un total de 6 encuestados no están de acuerdo ni en desacuerdo con esta proposición. Y solamente 1 encuestado está en desacuerdo, dando a notar que la visualización del diagrama de base de datos desde el IDE no aumenta la facilidad en el desarrollo. En resumen, la mayoría de estudiantes confirma que la propuesta de visualizar desde el IDE el modelo relacional de la base de datos, facilita y acelera el tiempo de desarrollo.

Sexta proposición.- Emplear componentes gráficos especializados en las operaciones transaccionales con base de datos infunden comodidad y facilidad al desarrollar aplicaciones.

ILUSTRACIÓN 35.- PORCENTAJES DE LA SEXTA PROPOSICIÓN DE LA ENCUESTA

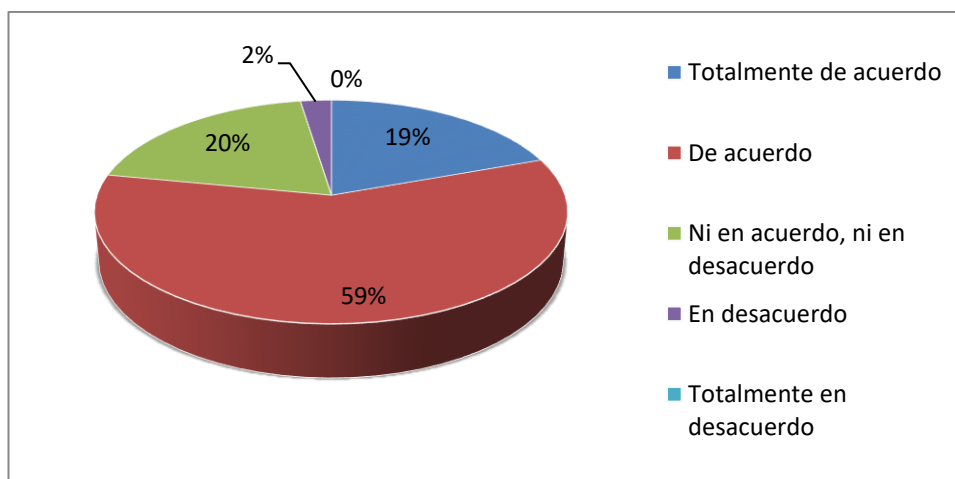


TABLA 19.- RESULTADOS DE LA SEXTA PROPOSICIÓN DE LA ENCUESTA

RESPUESTAS	RESULTADOS	PORCENTAJE (%)
Totalmente de acuerdo	8	19.51%
De acuerdo	24	58.53%
Ni en acuerdo, ni en desacuerdo	8	19.51%

En desacuerdo	1	2.43%
Totalmente en desacuerdo	0	0%
TOTAL	41	100%

Análisis de la 6ta proposición.- El objetivo de esta proposición es verificar la comodidad y facilidad obtenida por los programadores al emplear componentes gráficos de ayuda para realizar las operaciones con base de datos. **Observando** los resultados, se verifica que un total de 32 encuestados encuentran completa comodidad al utilizar componentes gráficos especializados en el acceso a datos. Una cantidad de 8 encuestados son neutrales a esta proposición dando a notar que las experiencias obtenidas con componentes para el acceso a datos no han sido tan buenas. Y finalmente 1 encuestado está en desacuerdo con esta proposición, dando a entender que no ha encontrado comodidad al usar componentes gráficos. En esta proposición la resolución es que la mayoría de estudiantes encuentra completa comodidad cuando ha implementado componentes de asistencia gráfica.

Séptima proposición.- La implementación de código fuente y componentes de asistencia gráfica debe ser equilibrada al momento de desarrollar aplicaciones.

ILUSTRACIÓN 36.- PORCENTAJES DE LA SÉPTIMA PROPOSICIÓN DE LA ENCUESTA

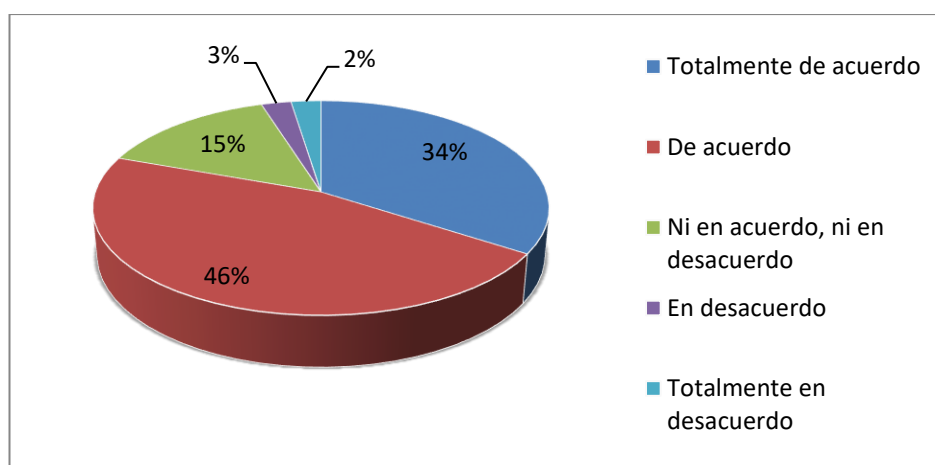


TABLA 20.- RESULTADOS DE LA SÉPTIMA PROPOSICIÓN DE LA ENCUESTA

RESPUESTAS	RESULTADOS	PORCENTAJE (%)
Totalmente de acuerdo	14	34.14%

De acuerdo	19	46.34%
Ni en acuerdo, ni en desacuerdo	6	14.63%
En desacuerdo	1	2.43%
Totalmente de desacuerdo	1	2.43%
TOTAL	41	100%

Análisis de la 7ma proposición.- El objetivo de esta proposición es saber si el programador desea mantener un equilibrio entre la implementación de código fuente y componentes de asistencia gráfica. **Observando** los resultados globales de esta proposición, se tiene que un total de 33 encuestados desea establecer un equilibrio entre la implementación de código fuente escrito e implementación de componentes gráficos, dando a entender que las dos formas de trabajo son importantes para realizar las actividades del desarrollo. Una cantidad de 6 encuestados son neutrales a esta proposición, y junto con otros 2 encuestados que están en desacuerdo con esta proposición, dan a notar que no desean tener un equilibrio entre estas dos formas de trabajo, sino que muestran más inclinación a una de estas dos formas de trabajo. En resumen, la mayoría de los estudiantes desean mantener un equilibrio entre escribir código fuente escrito y la ayuda de componentes gráficos.

Después del análisis realizado por cada proposición, se realiza un análisis general de toda la encuesta, esta se presenta en la sección de RESULTADOS del presente documento de tesis. Las encuestas realizadas a los estudiantes se encuentran en la sección de ANEXOS.

3.3. DISEÑO DE LA INVESTIGACIÓN

El diseño de la investigación radica en poner en evaluación el proyecto informático realizado dentro de la dimensión de desarrollo de aplicaciones; en este caso, se tendría que poner a prueba el componente JDataSet bajo la custodia de programadores expertos e inexpertos, que por medio de reportes den resultados propios para la comprobación de la hipótesis. Analizando estas acciones y debido al presente tipo de investigación se optó por realizar un diseño cuasi-experimental.

3.3.1. Diseño cuasi-experimental

Se considera apropiado el método cuasi-experimental, de un solo grupo que genere medidas pretest y posttest con una sola variable independiente; es decir, una medición antes y después de implementar la variable independiente. La variable independiente es el componente JDataSet.

En la aplicación del método cuasi-experimental primero se identifican las medidas pretest y luego las posttest. Para la medida pretest se denominó el procedimiento de acceso a datos implementando solo el código fuente del lenguaje de Java escrito línea a línea, y para la medida posttest se denominó el procedimiento de acceso a datos implementando las funcionalidades del componente JDataSet.

La notación del diseño cuasi-experimental es $O_1 X O_2$. Donde X es la variable independiente, en este caso sería el componente JDataSet.

O_1 es el grupo de individuos que se utilizó para realizar el estudio, que permitió proporcionar medidas del desarrollo de software en minutos, específicamente en el procedimiento de acceso a datos implementando la escritura de código fuente de Java línea a línea.

O_2 es el mismo grupo de individuos que se utilizó para realizar el estudio, y que proporcionó medidas del desarrollo de software en minutos, específicamente en el procedimiento de acceso a datos, pero esta vez implementando el componente JDataSet.

Este diseño permitió generar resultados cuantitativos que en lo posterior se compararon para definir la aceptación de la hipótesis planteada.

3.3.2. Pasos del diseño cuasi-experimental ²⁰

Los principales pasos que se realizaron para el desarrollo de un cuasi-experimento fueron los siguientes.

²⁰ <http://www.monografias.com/trabajos10/cuasi/cuasi.shtml>

Paso 1.- El primer paso consta de definir cuantas variables independientes y dependientes deberán ser incluidas en el cuasi-experimento. En este caso, la implementación del componente **JDataSet** será la variable independiente, y el **desarrollo de aplicaciones con conexión a base de datos** será la variable dependiente.

Paso 2.- El segundo paso consta de elegir los niveles de manipulación de la variable independiente y traducirlos en tratamientos experimentales. Como se observa en la tabla de matriz de operacionalización de variables de la hipótesis de la presente investigación, existe un indicador que mide tanto a la variable independiente como a la dependiente, este es el **tiempo de desarrollo**. Se comprobó el tiempo en minutos que se tarda en desarrollar ciertas actividades el programador cuando realiza la construcción de un software.

Paso 3.- El tercer paso consta de desarrollar el instrumento para medir la variable dependiente. Como se especifica en el paso anterior, el tratamiento para medir se basa en comprobar el tiempo en que el programador se tarda en realizar ciertas actividades de programación, es así como se optó por realizar **fichas** en las cuales se registraron de una manera ordenada los datos del cuasi-experimento. Las fichas se detallan de la siguiente manera.

TABLA 21.- FICHA DE REGISTRO DE DATOS DEL DISEÑO CUASI-EXPERIMENTAL

Actividades del programador		O ₁ = Implementando la escritura de código fuente línea por línea	O ₂ = Implementando el componente JDataSet
		Tiempo en minutos	Tiempo en minutos
1	Desarrollar un método para la conexión con la base de datos		
2	Desarrollar un método que realice una consulta de datos entre varias tablas relacionadas		
3	Desarrollar un método para la inserción de datos		
4	Desarrollar un método para la actualización de datos		
5	Desarrollar un método para la eliminación de datos		
6	Visualizar los registros afectados después de ejecutar una operación hacia la base de datos		

Aquí, el programador escribió el tiempo en minutos que tardó en realizar las actividades expuestas en este reporte. El ingreso de los valores consta en registrar la medida de cada actividad, primeramente implementando la escritura de código fuente línea por línea O₁ este vendría a ser la medida inicial o pretest, y después registrar la medida implementado el componente JDataSet O₂, este vendría a ser el posttest.

Paso 4.- El cuarto paso consta de seleccionar una muestra de personas para el experimento idealmente representativa de la población. En nuestro caso, se tiene una población finita compuesta de 91 estudiantes de la carrera de Sistemas de la FCI de la UTEQ, que genera una muestra de **41** estudiantes, este cálculo está detallado en la sección de Población y Muestra.

Paso 5.- El quinto paso consta en reclutar al grupo de personas del cuasi-experimento, esto implica tener contacto con ellos, darles las indicaciones necesarias y el máximo de facilidades para cumplir con el experimento. Siendo este paso el más importante a realizar en nuestro cuasi-experimento, se acudió a compañeros estudiantes, egresados y graduados de Sistemas que contribuyan con sus conocimientos, cabe destacar que para comprobar el tiempo de desarrollo de aplicaciones se necesitan programadores con experiencia en el campo laboral. Para esto también se realizó un tutorial basado en el uso del JDataSet, este contenido se subió en uno de los servidores de la UTEQ con el link <http://svrb.uteq.edu.ec:81/jdataset/>, para que los estudiantes a través de la web puedan descargar el plugin JDataSet, el archivo JAR jdataset-lib, y realizar el ejemplo señalado.

Paso 6.- El sexto paso consta en seleccionar el diseño cuasi-experimental apropiado para las muestras, hipótesis, objetivos y preguntas de la investigación. El **diseño de prepruebas (pretest) – postprueba (posttest) y grupo intacto** fue el seleccionado, ya que al principio se tuvo que realizar una preprueba para verificar la equivalencia inicial del grupo y las diferencias significativas de conocimiento, y posterior al tratamiento experimental realizar una postprueba para sacar las diferencias que fueron atribuidas al uso de la variable independiente.

Paso 7.- El séptimo paso consta de planear cómo se va a manejar al grupo de personas que participen en el experimento, desde el inicio hasta el fin del tratamiento. Como se explicó anteriormente el grupo de programadores fueron guiados por medio de un tutorial subido en la web sobre el uso del JDataSet, este fue el tratamiento experimental. Luego ellos registraron los valores en las fichas de registro de datos, estas fichas fueron almacenadas para su posterior análisis de resultados.

Los resultados de este diseño cuasi-experimental se detallan en la sección de RESULTADOS, y las fichas de registro de datos de cada programador están añadidas en la sección de ANEXOS.

3.4. POBLACIÓN Y MUESTRA

Al realizar una investigación siempre es necesaria la contribución de un grupo de personas con las cuales se pueda experimentar, con el fin de generar resultados de la puesta en práctica del proyecto informático.

3.4.1. Población finita

Este trabajo comprende como población a los programadores de software, en específico a los programadores que utilizan Java como lenguaje de programación y a Netbeans como IDE (entorno de desarrollo integrado).

Para esto, se consideraron a los estudiantes de la carrera de Sistemas de la Facultad de Ciencias de la Ingeniería de la UTEQ. La obtención de las cantidades de estudiantes por cada semestre resultó ser un poco complejo, ya que actualmente los estudiantes se pueden matricular por materias, pudiendo tomar clases anticipadas de materias que estén más arriba del semestre que cursan. Esto causa que en cada clase se encuentren cantidades diferentes de alumnos. Finalmente se optó por tomar las cantidades mayores de estudiantes de las materias que pertenecen a cada semestre, quedando los resultados presentados en la siguiente tabla.

TABLA 22.- POBLACIÓN FINITA CARRERA DE SISTEMAS

CARRERA DE SISTEMAS		CANTIDAD DE ESTUDIANTES	POBLACIÓN FINITA
Año	Semestre		
Primer	Primer	39	91 estudiantes
	Segundo	24	
Segundo	Tercer	0	
	Cuarto	35	
Tercer	Quinto	0	
	Sexto	25	
Cuarto	Séptimo	0	
	Octavo	31	
Quinto	Noveno	0	
	Décimo	17	

En la población finita solo se incluye a los estudiantes de segundo, tercero y cuarto año, dando un total de 91 estudiantes. Se excluye a los estudiantes del primer año por tener poca experiencia en el desarrollo de aplicaciones, y también se excluye a los estudiantes del quinto año porque la asistencia de sus clases no es a diario, siendo de baja disponibilidad.

3.4.2. Cálculo de la muestra

Teniendo una población finita de 91 estudiantes de la carrera de Sistemas de la FCI de la UTEQ, se tuvo que realizar una muestra representativa para realizar la investigación experimental. Dado que la población es finita, y se deseaba saber cuántos del total se tendría que estudiar, para eso se utilizó la siguiente formula.

$$n = \frac{Z^2 \cdot pq \cdot N}{Z^2 pq + e^2(N - 1)}$$

Dónde **N** es el total de la población, en este caso 91, **Z** es igual a 1.96 al cuadrado si la seguridad es del 95%, **p** es la proporción esperada en este caso 0.05, sino tuviésemos idea de dicha proposición utilizaríamos el valor 0.05 que

maximizará el tamaño muestral, q es igual a $1-p$ en este caso $1 - 0.05$ es igual a 0.95 , y e es el límite aceptable del error muestral en este caso 0.05 .

Según diferentes seguridades el coeficiente de Z varía así: si la seguridad Z fuese del 90% el coeficiente sería 1.645 , si la seguridad Z fuese del 95% el coeficiente sería 1.96 , si la seguridad Z fuese del 97.5% el coeficiente sería 2.24 , si la seguridad Z fuese del 99% el coeficiente sería 2.576 .

Una vez explicado las variables y los valores a incluir en la fórmula, haremos el cálculo preciso de la muestra:

$$n = \frac{1.96^2 * 0.05 * 0.95 * 91}{1.96^2 * 0.05 * 0.95 + 0.05^2(91 - 1)}$$

$$n = \frac{16.60}{0.18 + 0.22}$$

$$n = \frac{16.61}{0.41}$$

$$n = 40,79 \approx 41$$

El resultado muestral indica que debemos tomar a 41 estudiantes para realizar el estudio respectivo.

CAPITULO IV

RESULTADOS Y DISCUSIÓN

4.1 RESULTADOS

Los resultados constan de la presentación de todos los hechos importantes positivos y negativos en el desarrollo de la investigación.

4.1.1. Pruebas de software

Las pruebas y evaluaciones sobre el funcionamiento del componente JDataSet se realizaron según los factores que se ameritan considerar en la implantación de un software. Estas pruebas se realizaron en diferentes computadores, cada uno con características de procesamiento diferentes, en las cuales estuviere el programa instalado de Netbeans IDE, como requerimiento indispensable.

Se realizaron los siguientes tipos de prueba de desempeño, de recuperación y tolerancia a fallas, de volumen, de compatibilidad y conversiones, del GUI (Graphical User Interface), de aceptación y de instalación. También se verificaron los modos de uso en librerías de clases, en aplicaciones web y aplicaciones de escritorio.

Pruebas de desempeño

Este tipo de prueba se basa en medir el tiempo de respuesta de las transacciones, se midieron los tiempos de respuesta específicamente en las tareas de ejecución al probar una sentencia SQL, esto es en la parte del constructor de sentencias (Query Builder). El tiempo de respuesta de las operaciones transaccionales demuestra un resultado normal, ya que la carga de los datos extraídos desde la base de datos se realiza en forma directa sin mecanismos intermediarios.

Pruebas de recuperación y tolerancia a fallas

Estas pruebas aseguran que el software se recupere de una variedad de anomalías de hardware, software o fallas de integridad. La plataforma de Java tiene la virtud de no colgarse en la ejecución de sus aplicaciones, pero al mismo tiempo genera grandes dudas al no saber en qué parte del código está el error. Afortunadamente, los errores encontrados en el funcionamiento del

componente JDataSet son comprensibles, y estos pueden ser por no encontrar la base de datos registrada en el archivo XML, o la incorrecta modificación del archivo XML, ya que de este consta el funcionamiento del JDataSet. Al editar el contenido escrito del archivo XML se comprobó que su modificación puede resultar tanto para cometimiento de errores, como también para solucionar problemas.

Pruebas de volumen

En esta prueba se verifica el tamaño de almacenamiento que el software podría soportar o alcanzar en diferentes gestiones de uso. El método utilizado para la gestión de almacenamiento se trata de un documento XML que se genera al configurar una conexión con la base de datos. En este archivo XML se registran los datos de conexión, los nombres de los objetos que pertenecen a la base de datos y sus respectivos atributos. Al usar un archivo XML permite que sea comprensible a cualquier sujeto de intelecto informático, ya que este tipo de archivos es reconocido a nivel mundial.

Pruebas de compatibilidad y conversiones

El propósito de este tipo de prueba es buscar si existen problemas de compatibilidad y conversión del software. Como primer paso se probó el componente JDataSet en una versión más avanzada de Netbeans IDE 7.4, la última versión hasta la presente fecha, en la cual no se obtuvo ningún problema de compatibilidad, de lo contrario, sirvió para realizar las correcciones pertinentes de su funcionamiento.

Pruebas del GUI

Esta es la prueba de las interfaces de usuarios que verifica la interacción del usuario con el software. Las interfaces gráficas del JDataSet son completamente amigables y entendibles para el uso de programadores con experiencia y sin experiencia en el desarrollo de aplicaciones en Java. El componente JDataSet consta de acciones cortas y explícitas, fáciles de

comprender. Es una versión inicial y solo presenta las actividades más conocidas en lo que concibe al proceso de acceso a datos.

Pruebas de Aceptación

La prueba de aceptación es ejecutada antes de que el software sea instalado dentro del ambiente de trabajo; es decir, radica en la determinación por parte del usuario (programador) la aceptación o rechazo del software. En este caso, el componente JDataSet fue construido usando la metodología de desarrollo de programación extrema, en la cual se itera la evaluación por parte del usuario final. Las conclusiones de los programadores fueron positivas dando un nivel de aceptación en sus funcionamientos y características. Se alcanzó críticas de conformidad en las operaciones de acceso a datos. El componente JDataSet lanza buenas perspectivas a futuro para ser un producto que se encuentra en una versión inicial.

Pruebas de instalación

Esta prueba consiste en asegurar que el software puede ser instalado en todas las configuraciones posibles, y también verificar que, una vez instalado, el software opera correctamente. El componente JDataSet integra sus funciones y acciones al Netbeans IDE a través de la instalación manual del archivo `org.netbeans.dataset1.nbm`, ejecutando las operaciones correctamente.

Uso en librerías de clases Java (Java Class Library)

Por lo general, los programadores suelen realizar su capa de acceso a datos en una librería aparte del proyecto en cuestión, por esa razón se evaluó la integración del JDataSet en una Librería de Clases sin tener inconvenientes, añadiendo una clase con método Main para verificar en la ventana de salida (Output) los resultados de la sentencia efectuada en el editor del JDataSet. Así se determina que el JDataSet puede estar vinculado a una biblioteca de clases de Java y establecer las funciones requeridas por medio de la distribución de módulos o librerías.

Uso en aplicaciones de escritorio (Java Application)

Al igual que la prueba en el modo de uso dentro de una librería de clases se dieron las correcciones pertinentes. No se obtuvo ningún inconveniente en el uso del componente JDataSet para aplicaciones de escritorio en Java.

Uso en aplicaciones web (Web Application)

Para la prueba del componente JDataSet en aplicaciones web, se realizó por medio de la compilación de un archivo WAR. Este archivo WAR es un encapsulador de todos los archivos que contiene una aplicación web en Java. Este archivo WAR fue subido a la carpeta de aplicaciones del servidor Apache Tomcat; en donde, al iniciar el servidor descomprime el contenido de sus archivos en una carpeta. Se hicieron las correcciones de los errores encontrados dando la solución adecuada. El principal inconveniente fue encontrado en el llamado del path o directorio del archivo XML que genera el componente JDataSet, después de hacer la configuración de una conexión con la base de datos; en fin, luego de seguir realizando las pruebas y evaluaciones se alcanzó a solucionar el problema.

4.1.2. Análisis general de las encuestas

Las proposiciones planteadas en la encuesta fueron trazadas con el fin de llevar a cabo la consecución del desarrollo del software. A través de las encuestas se buscó consolidar la propuesta de la creación del componente JDataSet para Netbeans IDE. A continuación se presenta una tabla con los resultados generales de las encuestas, luego un análisis general de los resultados obtenidos.

TABLA 23.- RESULTADOS GENERALES DE LAS ENCUESTAS

Según las proposiciones	Totalmente de acuerdo	De acuerdo	Ni en acuerdo, ni en desacuerdo	En desacuerdo	Totalmente en desacuerdo	Cantidad de encuestados
1.- El programador obtiene completa confianza y seguridad al emplear solamente código fuente para realizar las operaciones transaccionales hacia la base de datos	14	17	9	1	0	41
2.- Codificar los métodos o funciones para la selección, inserción, actualización, y eliminación de datos, provocan que el desarrollo de la aplicación sea monótona, repetitiva y la vez compleja	10	19	5	6	1	41
3.- Los componentes gráficos que ayudan al programador a desarrollar aplicaciones, permiten que la construcción del software sea más sencilla y rápida	20	15	5	1	0	41
4.- El desarrollo de los métodos de selección, inserción, actualización, y eliminación de datos sería más fácil y rápida si se realizara con la ayuda de componentes de asistencia gráfica	17	17	3	3	1	41
5.- El procedimiento para realizar las operaciones con la base de datos sería más fácil si pudiéramos visualizar desde el IDE (entorno de desarrollo integrado) el diagrama del modelo relacional de la base de datos	15	19	6	1	0	41
6.- Emplear componentes gráficos especializados en las operaciones transaccionales con base de datos infunden comodidad y facilidad al desarrollar aplicaciones	8	24	8	1	0	41
7.- La implementación de código fuente y componentes de asistencia gráfica debe ser equilibrada al momento de desarrollar aplicaciones	14	19	6	1	1	41

Confirmación de la problemática

Las dos primeras proposiciones se basan en un objetivo en común, están trazadas en las tareas de codificación escrito línea por línea. Por ende, estas proposiciones juegan un papel elemental en la presente investigación de tesis.

Según las respuestas obtenidas se asegura que en la mayoría de los encuestados existen molestias al emplear solo código fuente para desarrollar aplicaciones, tales molestias pueden ser por recurrencia de código, lentitud en el desarrollo, monotonía de procedimientos al depender de herramientas externas al IDE, falta de visualización del modelo relacional de la base de datos, entre otras molestias.

Es verdad que un programador obtiene total confianza y seguridad en el código fuente escrito línea por línea, pero también causa molestias en las tareas de codificación del acceso a datos, aumentando el trabajo de los programadores.

Consecución del software

Las proposiciones siguientes, desde la tercera hasta la séptima proposición, fueron planteadas con el propósito de proponer con más detalle las características funcionales que debe adquirir el componente gráfico a desarrollar.

Primero se necesitaba conocer el concepto que tienen los estudiantes de sistemas al emplear componentes gráficos especializados en diferentes índoles de desarrollo. También se necesitaba conocer si los estudiantes obtenían un trabajo más sencillo y rápido al utilizar herramientas gráficas, la mayoría de los encuestados dieron repuestas positivas.

Después se proponen enunciados en las que se deduce que las operaciones transaccionales hacia la base de datos podrían realizarse con mayor facilidad y rapidez si existiera un método gráfico para realizar estas tareas. También, se da la propuesta de que el programador tenga la posibilidad de visualizar en el IDE el diagrama modelo relacional de la base de datos. Estas proposiciones

tuvieron firme aceptación en la mayoría de los estudiantes, lo que establece la consecución de los objetivos específicos planteados en la presente investigación de tesis.

Luego se realiza una sexta proposición con el fin de saber si los estudiantes han tenido satisfacción al utilizar herramientas o componentes gráficos especializados en las operaciones transaccionales hacia la base de datos, la mayoría dieron a entender que si han usado herramientas con opciones para el acceso a datos, y que al utilizarlos les brinda la suficiente comodidad y facilidad.

En la última proposición, solo se aspira saber si los programadores desean tener un equilibrio entre la implementación de código fuente escrito línea por línea y la implementación de componentes de asistencia gráfica, la mayoría dieron resultados afirmativos a esta proposición.

En conclusión, según el mutuo acuerdo a las proposiciones planteadas en la encuesta, certifican la consecución de la creación del componente de asistencia gráfica de acceso a datos JDataSet, especializado en facilitar el desarrollo de aplicaciones con conexión a base de datos.

4.1.3. Análisis del cuasi-experimento

Una vez ejecutado el diseño cuasi-experimental detallado en la sección DISEÑO DE LA INVESTIGACION del presente documento de tesis, se obtienen los resultados de manera general, quedando como resolución que el empleo del componente JDataSet facilita y reduce los tiempos de desarrollo de aplicaciones.

Una vez que los programadores cumplieron con las tareas exigidas en el cuasi-experimento, se finalizó con la entrega de las fichas de registro de datos. Cada programador registró su tiempo estimado al cumplir con las actividades especificadas en la fichas, por lo cual las cantidades tuvieron que ser sumadas para sacar un promedio general de tiempos. El siguiente cuadro muestra los tiempos en promedios generales.

TABLA 24.- FICHA DE LOS RESULTADOS GENERALES DEL DISEÑO CUASI-EXPERIMENTAL

Actividades del programador		O ₁ = Implementando la escritura de código fuente línea por línea	O ₂ = Implementando el componente JDataSet
		Tiempo en minutos	Tiempo en minutos
1	Desarrollar un método para la conexión con la base de datos	15	0.3
2	Desarrollar un método que realice una consulta de datos entre varias tablas relacionadas	25	3
3	Desarrollar un método para la inserción de datos	16	0.5
4	Desarrollar un método para la actualización de datos	20	0.5
5	Desarrollar un método para la eliminación de datos	13	0.5
6	Visualizar los registros afectados después de ejecutar una operación hacia la base de datos	2	0.2
TOTAL		91	5
MEDIA		15.16	0.83
DESVIACIÓN ESTÁNDAR		7.73	0.98

Como se puede observar, el resultado de la Media del grupo con O₁ es de 15.16 y su desviación estándar es de 7.73, mientras que la Media del grupo con O₂ es de 0.83 y su desviación estándar es 0.98, rotundamente existe una diferencia clara entre las medidas del pretest y postest. Según estos resultados, se podría aprobar la hipótesis planteada, pero es necesario realizar un cálculo estadístico para confirmar si la diferencia entre las medias y las desviaciones estándar del grupo de estudio corresponden a la implementación de la variable independiente (en este caso, el uso del componente JDataSet).

Para realizar el cálculo estadístico, se determina la hipótesis de nulidad H₀ invirtiendo la hipótesis planteada. De esa manera se tendría que,

H₀ = Realizar el manejo de datos a través de un asistente gráfico de acceso a datos, **no** facilita el desarrollo de las aplicaciones con conexión a base de datos.

De igual forma, se plantea una hipótesis alterna. Esta es la hipótesis planteada.

H_a = Realizar el manejo de datos a través de un asistente gráfico de acceso a datos, facilita el desarrollo de las aplicaciones con conexión a base de datos.

Prueba t-Student

La prueba t-Student sirve para comparar las medias y las desviaciones estándar de dos grupos de datos. Con esta prueba se determina si entre esos parámetros las diferencias son estadísticamente significativas o por mera casualidad. Primero se calcula el error estándar de la diferencia entre las dos medias, este también es denominado como margen de error.

$$S_{x_1-x_2} = \sqrt{\frac{\sum x_{1^2} + \sum x_{2^2}}{n_1 + n_2 - 2} \left(\frac{1}{n_1} + \frac{1}{n_2} \right)}$$

$$S_{x_1-x_2} = \sqrt{\frac{15.16 + 0.83}{6 + 6 - 2} \left(\frac{1}{6} + \frac{1}{6} \right)}$$

$$S_{x_1-x_2} = 0.53$$

El error estándar o margen de error es de 0.53, ahora se puede obtener el resultado de la prueba t-Student, con la siguiente fórmula.

$$T = \frac{\bar{X}_1 - \bar{X}_2}{S_{x_1-x_2}}$$

$$T = \frac{15.16 - 0.83}{0.53}$$

$$T = 27.04$$

El resultado de la prueba T es de **27.04**. Esta cantidad se la debe comparar a una cantidad proveída según la tabla t-Student basándose en el grado de libertad y a un nivel de 0.05. Para despejar dudas, se calcula los grados de libertad (gl), con la siguiente fórmula.

$$gl = n1 + n2 - 2$$

$$gl = 6 + 6 - 2$$

$$gl = 10$$

Con 10 grados de libertad, teniendo un nivel de 0.05 asentándose en la tabla t-Student se obtiene una razón de **1.8125**.

Comparando, el resultado de la prueba T de **27.04** es mayor que **1.8125**, lo que demuestra que la intervención de la variable independiente es estadísticamente significativa; por lo tanto, los resultados son suficientemente concretos para llegar a la conclusión que la diferencia no deriva de la casualidad. Consecuentemente se acepta la H_a .

En resumen, **27.04 > 1.8125**, de esa manera se da por aceptada la hipótesis alterna y se puede aseverar que H_a realizar el manejo de datos a través de un asistente gráfico de acceso a datos, facilita el desarrollo de las aplicaciones con conexión a base de datos.

4.1.4. Comprobación de Hipótesis

El planteamiento de la hipótesis sustenta que *“Realizar el manejo de datos a través de un **asistente gráfico de acceso a datos**, facilita el **desarrollo de las aplicaciones con conexión a base de datos**”*.

Como se puede analizar la hipótesis consta de una variable independiente (asistente gráfico de acceso a datos), y una variable dependiente (desarrollo de las aplicaciones con conexión a base de datos). El fundamento de la presente investigación de tesis radica en encontrar una manera de facilitar el desarrollo de aplicaciones con conexión a base de datos, ya que en la dimensión de emplear solo el código fuente del lenguaje de Java escrito línea a línea provoca un código repetido y extenso, cediendo complejidad y monotonía al programador de software.

La propuesta de realizar un asistente gráfico especializado en el acceso a datos, surgió de la observación directa en otras plataformas de desarrollo de software, que tienen métodos y componentes que ayudan a una programación rápida y sencilla. Siendo Netbeans IDE una herramienta muy empleada en la comunidad informática de software libre, no cuenta con un componente con características de asistencia gráfica para el manejo de datos. Es así como a base de encuestas a los programadores de software, se logró la conciliación de que un asistente gráfico solucionaría esta problemática; y por medio de un cuasi-experimento realizado se obtuvieron resultados estadísticos significativos.

La variable independiente “**asistente gráfico de acceso a datos**” establecido en esta investigación como el componente JDataSet, presenta al usuario (programador de software) una interfaz amigable y de fácil manejo, permitiendo familiarizarse rápidamente con este. Las operaciones transaccionales que este componente permite realizar en su área de trabajo son de fácil comprensión, gracias a la secuencia de pasos que han sido abreviados para conseguir un proceso ágil en el acceso a datos.

La variable dependiente “**desarrollo de las aplicaciones con conexión a base de datos**” con el paso del tiempo se torna más accesible a recursos que facilitan su trabajo. Ahora, con la creación del JDataSet, el desarrollo de aplicaciones con conexión a base de datos no solamente se vería dependiente de la escritura de código fuente, sino también sería complementada por el componente JDataSet que reduce el tiempo de desarrollo de software, facilita los procedimientos para el acceso a datos, es cómodo y adaptable al programador.

4.1.5. Resultados de los objetivos planteados

La solución propuesta frente a la problemática expuesta en la presente investigación de tesis, se encuentra cimentada en la definición de objetivos generales y específicos, con el fin de enmendar los efectos causales del problema.

Según los objetivos específicos

Primer objetivo específico.- Visualizar los objetos que contiene una base de datos directamente desde un editor gráfico de Netbeans IDE para su respectivo manejo.

Resultado del primer objetivo.- La visualización de las tablas, los campos con sus respectivos tipos de datos, y las relaciones entre tablas son visualizados a través de la ventana editor del componente JDataSet.

Segundo objetivo específico.- Abreviar el procedimiento de manejo de datos, por medio de una secuencia de pasos a seguir en una interfaz gráfica.

Resultados del segundo objetivo.- El proceso para armar, definir y validar una sentencia SQL dentro del componente JDataSet, consiste en completar una secuencia de pasos a seguir, comenzando desde el escogimiento del tipo de sentencia a realizar (estas pueden ser select, insert, update, delete) hasta dejar almacenada la función efectuada para su debido uso en el proyecto en Java.

Tercer objetivo específico.- Complementar el desarrollo de aplicaciones, empleando este asistente gráfico como un componente esencial en un proyecto en Java usando Netbeans IDE.

Resultado del tercer objetivo.- El componente JDataSet fue desarrollado con la característica de poder ser incorporado como un tipo de archivo JDS, dentro de un proyecto aplicativo en Java. Una vez que el plugins JDataSet sea instalado en Netbeans IDE, se podrá añadir este componente desde el cuadro de diálogo File Type, al proyecto de Java en desarrollo.

Según el objetivo general

Objetivo general.- Crear un asistente gráfico de acceso a datos que facilite el procedimiento de manejo de datos en tiempo de desarrollo de aplicaciones en Java utilizando Netbeans como IDE.

Resultado del objetivo general.- El componente JDataSet contiene las funciones de manejo de datos más conocidas, tales como la selección (select), inserción (insert), actualización (update) y eliminación (delete) de los datos. El mismo que permite la sencillez de ejecutar tales procedimientos al momento de desarrollar aplicaciones en Java usando Netbeans como IDE.

4.2 DISCUSIÓN

A continuación, se expone una discusión sobre el porqué se escogió el tema de la tesis, con el que se aspira dar la autenticidad a la presente investigación de tesis.

4.2.1. Tema de la investigación

Después de un análisis sobre las tareas de codificación en el lenguaje de programación Java usando Netbeans como IDE, se da la notoriedad que al realizar los métodos de conexión y operaciones hacia la base de datos, es posible aportar con herramientas que reduzcan los esfuerzos en la programación de aplicaciones de software libre. En este caso, la investigación corresponde a los procesos o procedimientos que se deben realizar para la manipulación de datos.

Desde un principio, la investigación no solo consistió en encontrar los requisitos esenciales para lograr un asistente gráfico especializado en el acceso a datos, sino que también la investigación consistió en encontrar la herramienta de desarrollo, o el espacio de trabajo en donde se construiría el componente gráfico para Netbeans IDE.

Fue así, que por medio de deducciones del tutor y otros mentores, dieron opciones de trabajar sobre la misma plataforma de Netbeans. De esa manera, la investigación comenzó por el estudio de este nuevo tipo de proyecto, fundamentado en un framework de Java nombrado como Netbeans Platforms en donde se posibilita el desarrollo de módulos para Netbeans, o lo que mejor se conoce como un plugins.

4.2.2. Requisitos de usuario no existentes

En resumen, existen varios requisitos que el usuario (programador) desearía implementar en el desarrollo de una aplicación en Java empleando el componente JDataSet que aún no se encuentran dentro de sus acciones y funcionalidades, estos requisitos pueden ser los siguientes.

La integración de funcionalidades para el acceso a datos, tales como las operaciones en el manejo de procedimientos almacenados, disparadores, funciones, entre otras acciones que son contenidas en el administrador de base de datos.

La implementación de varios motores de base de datos de software libre, a los cuales se logre la conexión para realizar los trabajos exigidos por sus clientes. En el actual JDataSet solo se implementa la conexión para los motores de MySQL y MS SQL Server, el uno en representación del software de código abierto y el otro en representación del software de código cerrado.

La visualización y modificación de todos los objetos en el diagrama del modelo relacional de la base de datos conectada; es decir, que en el editor del JDataSet existan las opciones para modificar las relaciones, tablas, campos, y otros objetos presentados en el IDE, con la certeza de que estas acciones tengan efecto en la base de datos original. Esto es con el fin de ahorrar esfuerzos prescindiendo de acudir al gestor de base de datos, y realizar todos los cambios en el mismo IDE.

Todos estos requisitos por parte del usuario (programador) son aceptables y verdaderamente indispensables para la asistencia de los programadores, pero también por razonamiento lógico se entiende que este proyecto informático está en su etapa inicial, sin descartar su evolución con el paso del tiempo y que las funcionalidades del componente JDataSet puedan seguir extendiéndose.

CAPITULO V

CONCLUSIONES Y RECOMENDACIONES

5.1 Conclusiones

- ❖ La posibilidad de visualizar los objetos; es decir, las tablas, campos y relaciones que comprenden la base de datos desde el editor visual del componente JDataSet, hace que el programador ocupe mayor dependencia del entorno de trabajo del IDE que en acudir al gestor de la base de datos. La visualización de los objetos de la base de datos también ayuda en la preparación de las consultas complejas.
- ❖ Todas las acciones incorporadas en el editor gráfico del componente JDataSet, reemplazan las extensas y monótonas tareas de codificar los métodos establecidos para realizar las operaciones transaccionales hacia la base de datos en el desarrollo de aplicaciones. De esa manera, se puede decir que el procedimiento de acceso a datos es abreviado.
- ❖ Los ensayos efectuados en esta investigación demuestran las capacidades funcionales que posee el JDataSet. Esto permite determinar que emplear este componente en un proyecto de Java complementa en gran manera los planes de codificación que realiza el equipo de desarrolladores de software.
- ❖ Implementar el componente JDataSet al momento de desarrollar aplicaciones en Java usando Netbeans como IDE, reduce el tiempo de desarrollo, facilita el procedimiento de acceso a datos, proporciona agilidad y mayor comodidad al programador.

5.2 Recomendaciones

- ❖ Para que el programador obtenga mayor dependencia del entorno de trabajo de Netbeans IDE, se podría realizar más funcionalidad sobre la creación y modificación de los objetos que comprende la base de datos, directamente desde el editor gráfico del componente JDataSet.
- ❖ La creación de un nuevo módulo para Netbeans (plugins) especializado en las demás operaciones de acceso a datos que no se presentan en este módulo (JDataSet), operaciones como procedimientos almacenados, disparadores, entre otros, mejoraría y complementaría el uso del componente JDataSet, y además contribuiría con el esparcimiento de nuevos proyectos de esta índole.
- ❖ Se debe seguir con la investigación de otras tareas que se realizan en los diferentes procedimientos de codificación que el equipo de programadores realiza al desarrollar aplicaciones, para crear componentes de asistencia gráfica especializados en esas tareas, con el propósito de seguir facilitando e impulsando el desarrollo rápido de aplicaciones.

CAPITULO VI

BIBLIOGRAFÍA

6.1. Literatura Citada

Böck, Heiko. (2012). *The Definitive Guide to Netbeans Platform 7*. Apress.

Böck, Heiko. (2009). *The Definitive Guide to Netbeans Platform 6.5*. Apress.

Boudreau, Tim. (2007). *Rich Client Programming: Plugging into the Netbeans Platform*. Prentice Hall.

Caules, Cecilio. Á. (2012). *Arquitectura Java Sólida*. Sun Certified Enterprise Architect.

Jurgüen, Petri. (2010). *Netbeans Platform 6.9 Developer's Guide*. Packt Publishing.

6.2. Linkografía

Bravo, A. V. (s.f.). *Avbravo Tutoriales español*. Obtenido de http://wiki.netbeans.org/Avbravo_TutorialesEspanol

Funes, L. E. (s.f.). *Conociendo Netbeans Platform*. Obtenido de <http://wiki.netbeans.org/ConociendoNetbeansPlatformIntroduccion>

NetBeans. (s.f.). *NetBeans Platform Learning Trail*. Obtenido de <https://netbeans.org/kb/trails/platform.html>

Rittner, A. (s.f.). *JNBB – Joschs NetBeans Blog*. Obtenido de <http://www.aljoscha-rittner.de/blog/>

Wielenga, G. (s.f.). *Geertjan's Blog*. Obtenido de <https://blogs.oracle.com/geertjan/>

Wielenga, G. (s.f.). *How to Create a Pluggable Photo Album in Java*. Obtenido de <http://java.dzone.com/news/how-create-pluggable-photo-alb>

CAPITULO VII

ANEXOS

UNIVERSIDAD TÉCNICA ESTATAL DE QUEVEDO



Uso del Plugins JDataSet

Daniel Andrés De los Santos Villao

Uso del Plugins JDataSet

Este tutorial demuestra el funcionamiento de plugins JDataSet, en el IDE Netbeans versión 7.4



La creación de este plugins se basa en el siguiente objetivo: Crear un asistente gráfico de acceso a datos, que facilite el procedimiento de llamado y manejo de datos en tiempo de desarrollo de una aplicación en Java.

Con la ayuda de este componente, el programador podrá realizar las operaciones de llamado de datos a través de un asistente gráfico al momento de desarrollar una aplicación en Java usando Netbeans como IDE.

Este asistente gráfico de acceso a datos permite la conexión a una base de datos, donde el programador podrá visualizar en el IDE los objetos del modelo estructural de la base datos, es decir que podrá ver las tablas con sus respectivos campos y relaciones. También el asistente gráfico permitirá que el programador arme cualquier sentencia SQL y al mismo tiempo probarla para comprobar su validez, todas estas tareas se podrán realizar en el IDE.

Al final, después de haber realizado todos estos procedimientos, permitirá que el programador ya no utilice los largos y redundantes métodos para el acceso a datos a través de código fuente, sino implementar un reducido código fuente para llamar las sentencias realizadas en el asistente gráfico del JDataSet.

Para un mejor entendimiento de las tareas que proporciona el JDataSet haremos una aplicación en Java usando el IDE Netbeans en su última versión 7.4

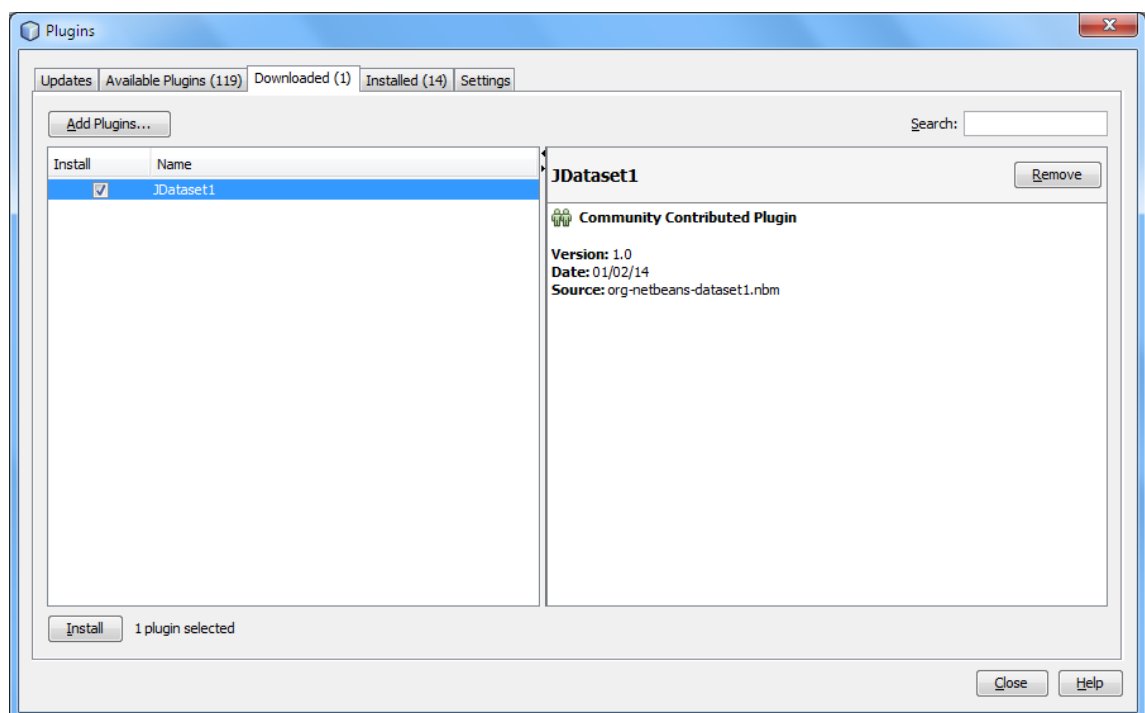
En este ejemplo se utilizarán los siguientes archivos: el plugins JDataSet, y el archivo JAR jdataset-lib.jar. Estos archivos los puedes descargar desde la siguiente dirección web en donde también se encuentra este mismo manual de usuario: <http://svrb.uteq.edu.ec:81/jdataset/>,

1.- Instalando el plugins JDataSet en Netbeans IDE

Una vez realizada la descarga del archivo org-netbeans-dataset1.nbm, lo instalamos de la siguiente manera.

En el menú principal de Netbeans vamos a la opción Tools, seleccionamos Plugins, se abrirá una ventana en la cual debemos escoger la pestaña Downloaded. En esta opción, Netbeans nos permite añadir un plugins que hayamos descargado y luego instalarlo. Damos clic en el botón Add Plugins, seguidamente mostrará un cuadro de diálogo que permite buscar un archivo NBM.

Un archivo NBM es un plugins para Netbeans. Buscamos el archivo org-netbeans-dataset1.nbm que hemos descargado y lo agregamos. En este momento debes de tener la siguiente vista:

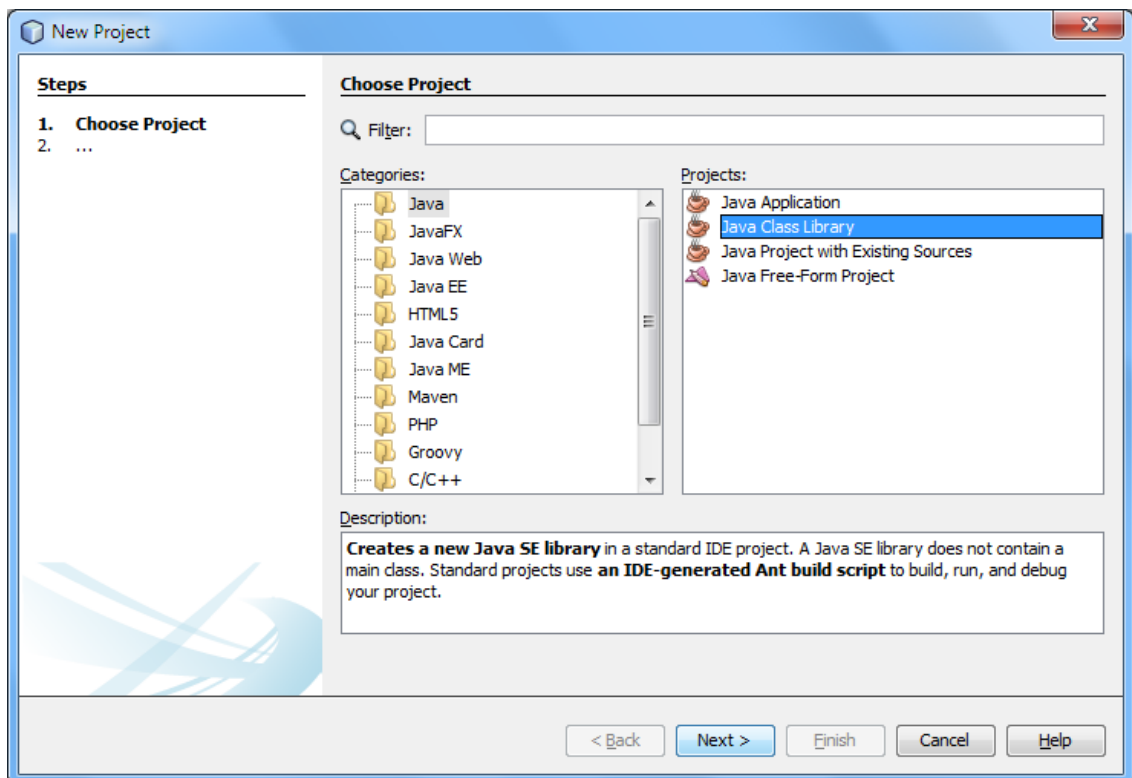


Procedemos a instalar el plugins JDataset1, damos clic en el botón Install. Como en toda instalación te pedirá que aceptes los términos de licencia, y de allí en adelante solo dale clic en el botón Next cuando lo amerite.

Una vez ya instalado el plugins explicaré las tareas que este permite realizar. Para eso crearemos una aplicación web que permita realizar la sistematización de una concesionaria de alquiler de vehículos, pero solo realizaremos la inserción y selección de vehículos.

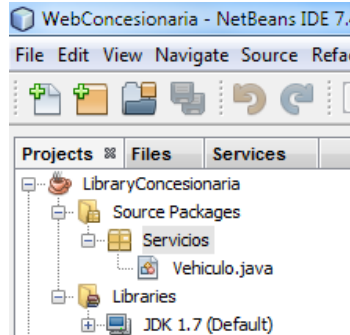
2.- Creación de una librería que contenga el componente JDataSet

En primer lugar crearemos una librería de clases. Vamos al menú principal de Netbeans seleccionamos File, en el menú desplegable seleccionamos New Project..., en el cuadro de dialogo que se muestra escogemos la categoría Java de la parte izquierda, y seleccionamos un proyecto tipo Java Class Library, tal y como se muestra en la siguiente figura.



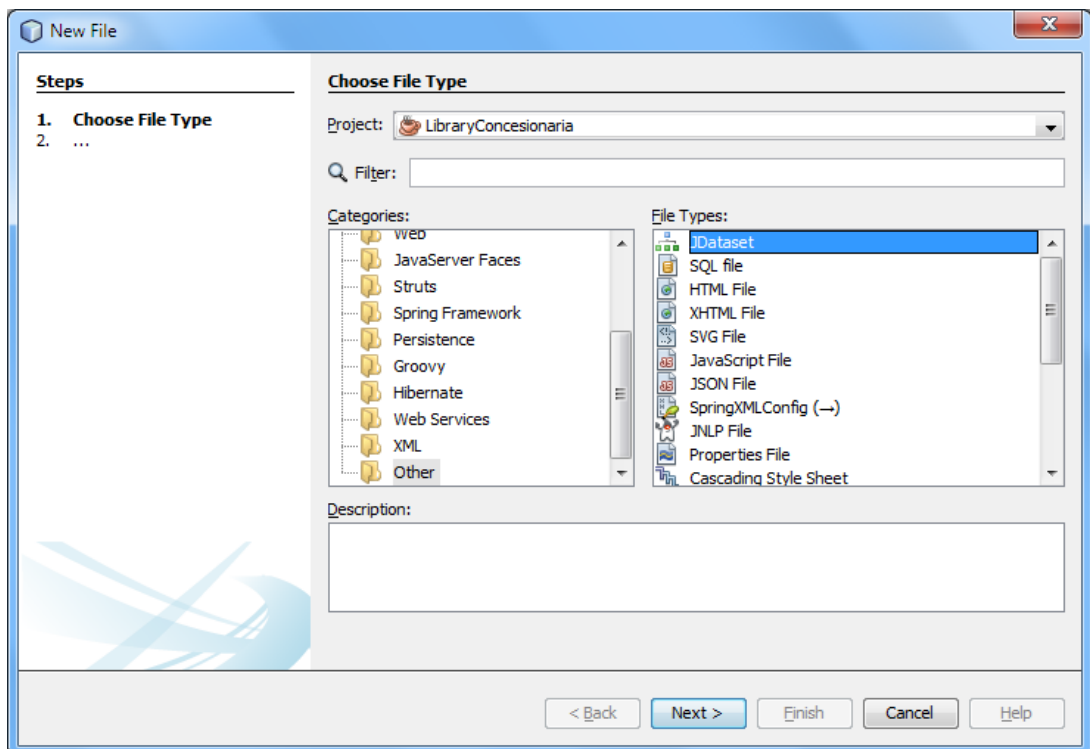
Damos clic en Next, y en la siguiente página le ponemos como nombre del proyecto LibraryConcesionaria, y finalmente damos clic en Finish.

Seguidamente el IDE creará el nuevo proyecto. En el nodo Source Packages agregamos un nuevo paquete Servicios y dentro de este agregamos una clase Vehiculo. Tendrás que tener la siguiente vista en la pestaña Projects.

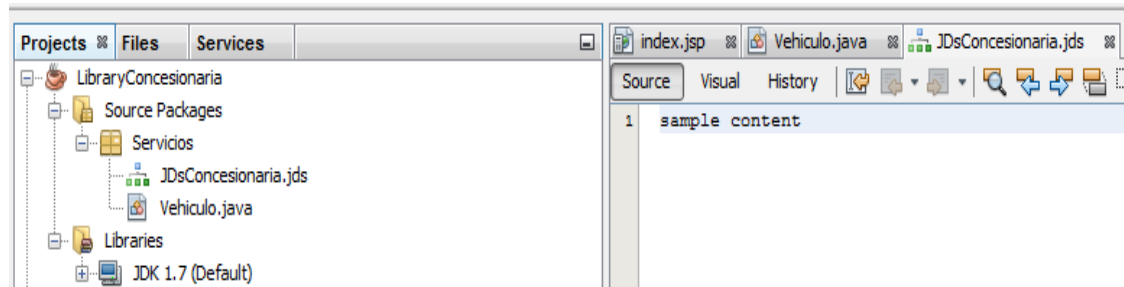


Antes de comenzar a codificar la clase Vehículo, agregaremos el componente JDataSet para realizar el procedimiento de acceso y llamado de datos.

Damos clic derecho sobre el nodo del paquete Servicios, damos clic en New, clic en Other..., seguidamente mostrará un cuadro de diálogo para seleccionar un tipo de archivo, en el lado izquierdo de categorías escogemos Other, y en el lado de Tipos de archivo seleccionamos JDataSet, como vemos en la siguiente figura:



Damos clic en Next, y en la siguiente página le ponemos como nombre JDsConcesionaria y damos clic en Finish. Vemos que se nos añade un nuevo archivo JDsConcesionaria.jds a nuestro proyecto LibraryConcesionaria, y también una nueva ventana editor en el entorno de trabajo.



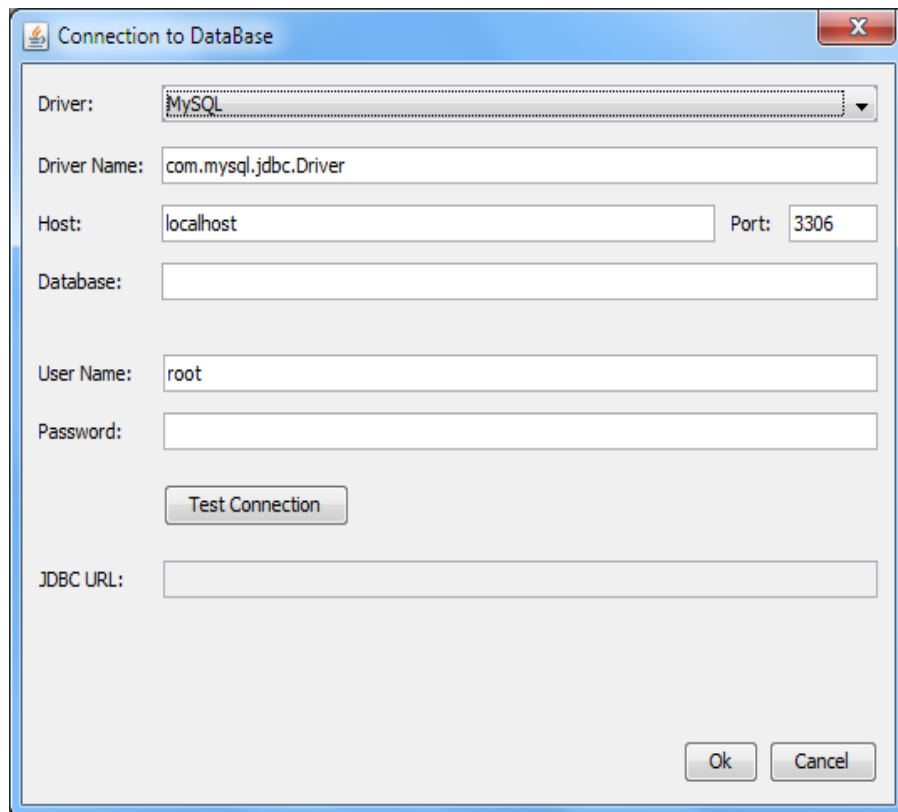
En la ventana editor del archivo JDsConcesionaria.jds se muestran tres opciones de vista: Source, Visual e History, pero solo la opción Visual es la que sirve y representa el modelo estructural de la base de datos.

Ya que todavía no hemos configurado la conexión a una base de datos, la opción Visual no mostrará nada.

Nota: La instalación del plugins o archivo org-netbeans-dataset1.nbm antes realizada, generó la adición de un nuevo tipo de archivo en el cuadro de dialogo de selección de tipos de archivos, en la última categoría Other. De esta instalación es donde se origina el surgimiento de estas nuevas opciones que han permitido extender el funcionamiento del IDE Netbeans.

3.- Configuración de conexión a la base de datos en el JDataSet.

Damos clic derecho sobre el nodo JDsConcesionaria.jds y seleccionamos Configure Connection, nos mostrará la siguiente ventana:



Aquí establecemos los datos de conexión a la base de datos. En primera instancia aparecerán los datos de conexión hacia el motor de base de datos de MySQL, también existe la opción para conectar a SQL Server, uno en representación de código abierto y el otro de código cerrado.

En este caso vamos a utilizar una base de datos concesionaria creada en MySQL, esta base de datos fue creada de la siguiente manera.

```

use conscesionario;

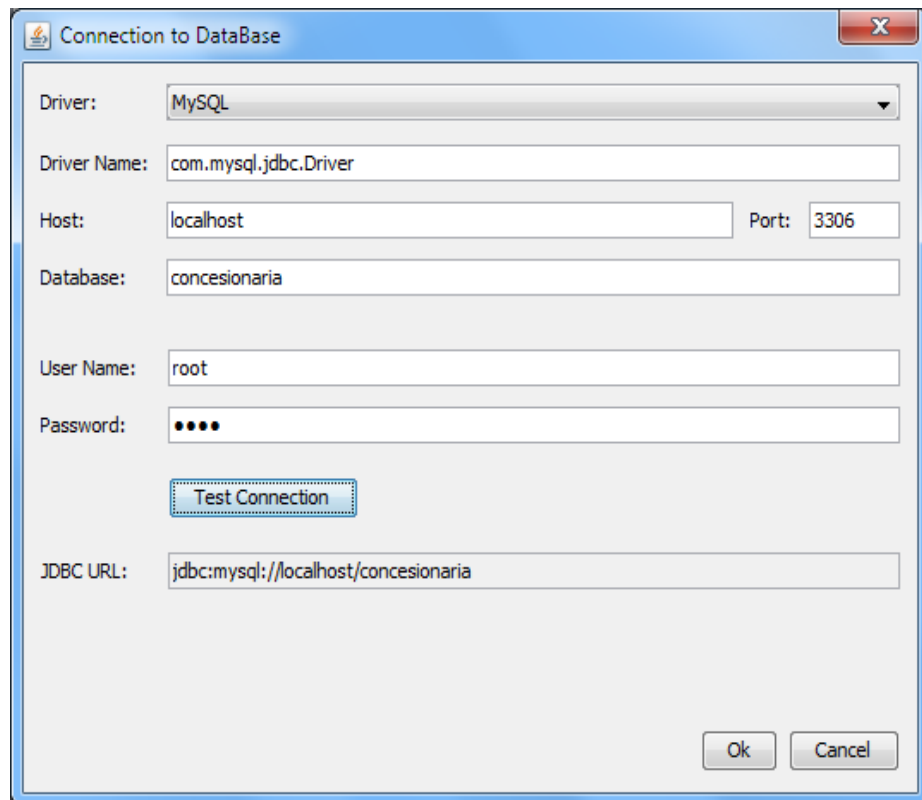
create table cliente(
cedula varchar(10) not null primary key,
nombre varchar(40) not null,
apellido varchar(40)
);

create table vehiculo(
placa varchar(7) not null primary key,
matricula varchar(10) not null,
marca varchar(40) not null,
modelo varchar(40) not null
);

create table alquiler(
idalquiler integer not null auto_increment primary key,
cedula varchar(10) not null,
placa varchar(7) not null,
fechasalida datetime not null,
fechaentrega datetime not null,
constraint fk_alquiler_1 foreign key fk_alquiler_1 (cedula)
references cliente (cedula)
on delete cascade
on update cascade,
constraint fk_alquiler_2 foreign key fk_alquiler_2 (placa)
references vehiculo (placa)
on delete cascade
on update cascade
)

```

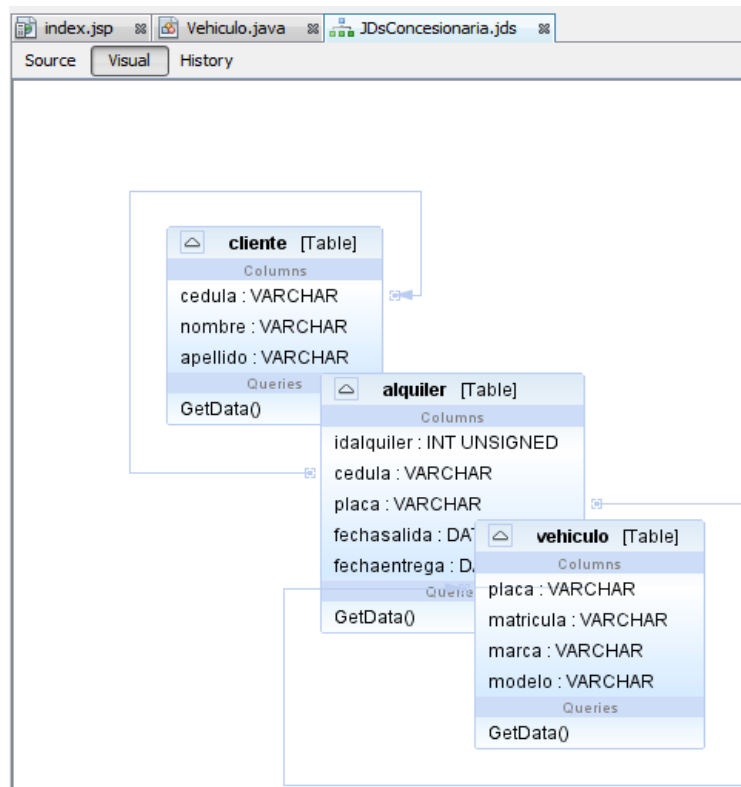
En el cuadro de conexión a la base de datos dejamos los datos que aparecen por defecto. En el cuadro de texto Database ingresamos el nombre de la base de datos “concesionaria”, también ingresamos el password según como tengas configurado tu MySQL, damos clic en el botón Test Connection para probar la conexión, si todo está bien se añade la cadena de conexión al cuadro de texto JDBC URL, como vemos en la siguiente figura.



La configuración de los datos de conexión depende de cómo tengas configurado tu MySQL, si tienes otros datos tendrás que ingresar los datos que corresponden. Finalmente damos clic en el botón Ok.

Inmediatamente se te añadirá un archivo XML con el mismo nombre que le diste al componente JDataset, de esa manera la ventana Visual del archivo JDsConcesionaria.jds te mostrará el modelo estructural de la base de datos.

Para que la ventana Visual del archivo JDsConcesionaria.jds pueda actualizarse tendrás que dar clic en cualquier otra de las ventanas editoras de los archivos que tengas abiertos actualmente y luego volver a darle clic en la ventana Visual del archivo JDsConcesionaria.jds. Verás que esta ventana se te actualizará dando un movimiento de las tablas en forma diagonal desde la parte superior izquierda hacia la parte inferior derecha, como verás en la siguiente figura.



Aquí puedes ver cómo están estructuradas las tablas con sus respectivos campos y relaciones. Si abres el archivo JDsConcesionaria.xml verás todos los datos de conexión y los objetos de la base de datos, pues de este archivo XML depende todo el funcionamiento del componente JDataSet.

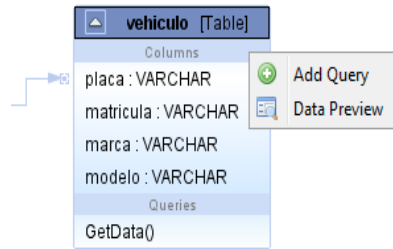
Abre el archivo JDsConcesionaria.xml y dentro de este dale clic derecho, selecciona la opción Format para que la estructura del código se pueda ordenar según las etiquetas, como muestra en la siguiente figura:

```
1 <?xml version="1.0" encoding="UTF-8" standalone="no"?>
2 <database drivername="com.mysql.jdbc.Driver" host="localhost" namedb="concesionaria" password="1234" port="3306" url="jdbc:mysql:
3
4 <table name="alquiler">
5 <column type="INT UNSIGNED">idalquiler</column>
6 <column type="VARCHAR">cedula</column>
7 <column type="VARCHAR">placa</column>
8 <column type="DATETIME">fechasalida</column>
9 <column type="DATETIME">fechaentrega</column>
10 <function name="GetData()">select *from alquiler</function>
11 </table>
12 <table name="cliente">
13 <column type="VARCHAR">cedula</column>
14 <column type="VARCHAR">nombre</column>
15 <column type="VARCHAR">apellido</column>
16 <relation>
17 <foreignKey>alquiler:cedula</foreignKey>
18 <primaryKey>cliente:cedula</primaryKey>
19 </relation>
20 <function name="GetData()">select *from cliente</function>
21 </table>
22 <table name="vehiculo">
23 <column type="VARCHAR">placa</column>
24 <column type="VARCHAR">matricula</column>
25 <column type="VARCHAR">marca</column>
26 <column type="VARCHAR">modelo</column>
27 <relation>
28 <foreignKey>alquiler:placa</foreignKey>
29 <primaryKey>vehiculo:placa</primaryKey>
30 </relation>
31 <function name="GetData()">select *from vehiculo</function>
32 <function name="InsertarVehiculo()">INSERT INTO vehiculo (placa, matricula, marca, modelo)
33 VALUES (@placa, @matricula, @marca, @modelo)</function>
34 <function name="SeleccionarPorMarca()">SELECT placa, matricula, marca, modelo
35 FROM vehiculo
WHERE marca = @marca</function></table>
```

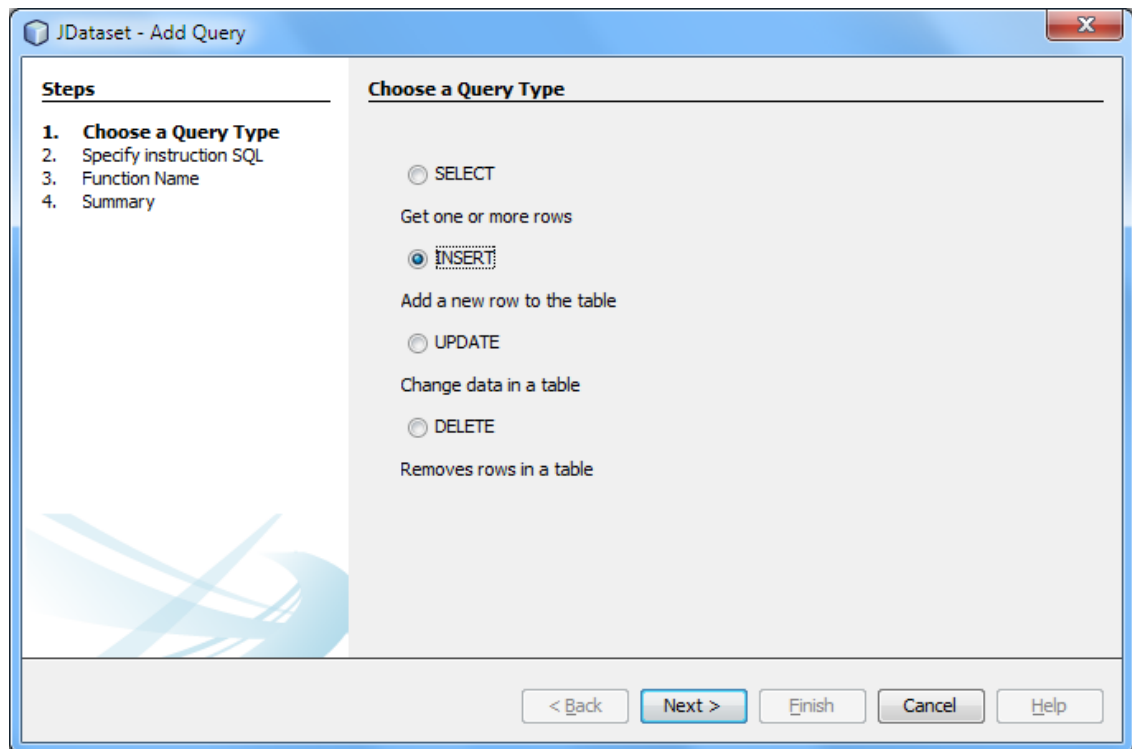
Debes de tener cuidado de que el código del archivo JDsConcesionaria.xml no sea modificado, pues aquí es donde guardamos todos los datos de conexión con la base de datos y los datos de todas las operaciones que van a ser realizadas en este componente. Y como antes mencioné el funcionamiento del componente JDataset depende de este archivo XML.

4.- Realizar una sentencia tipo INSERT en el JDataSet.

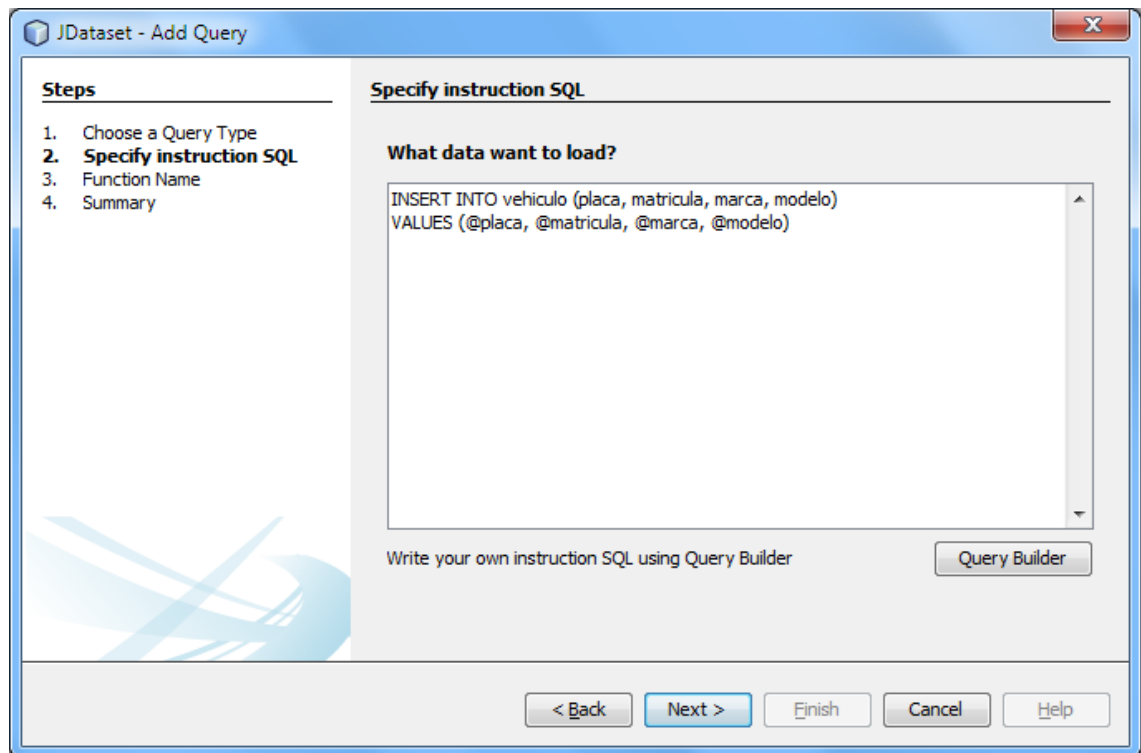
Ahora vamos a detallar cuales son las tareas que el componente JDataSet posee. En la ventana Visual del archivo JDsConcesionaria.jds vamos a la tabla vehículo damos clic derecho sobre este, tal y como se muestra en la siguiente figura.



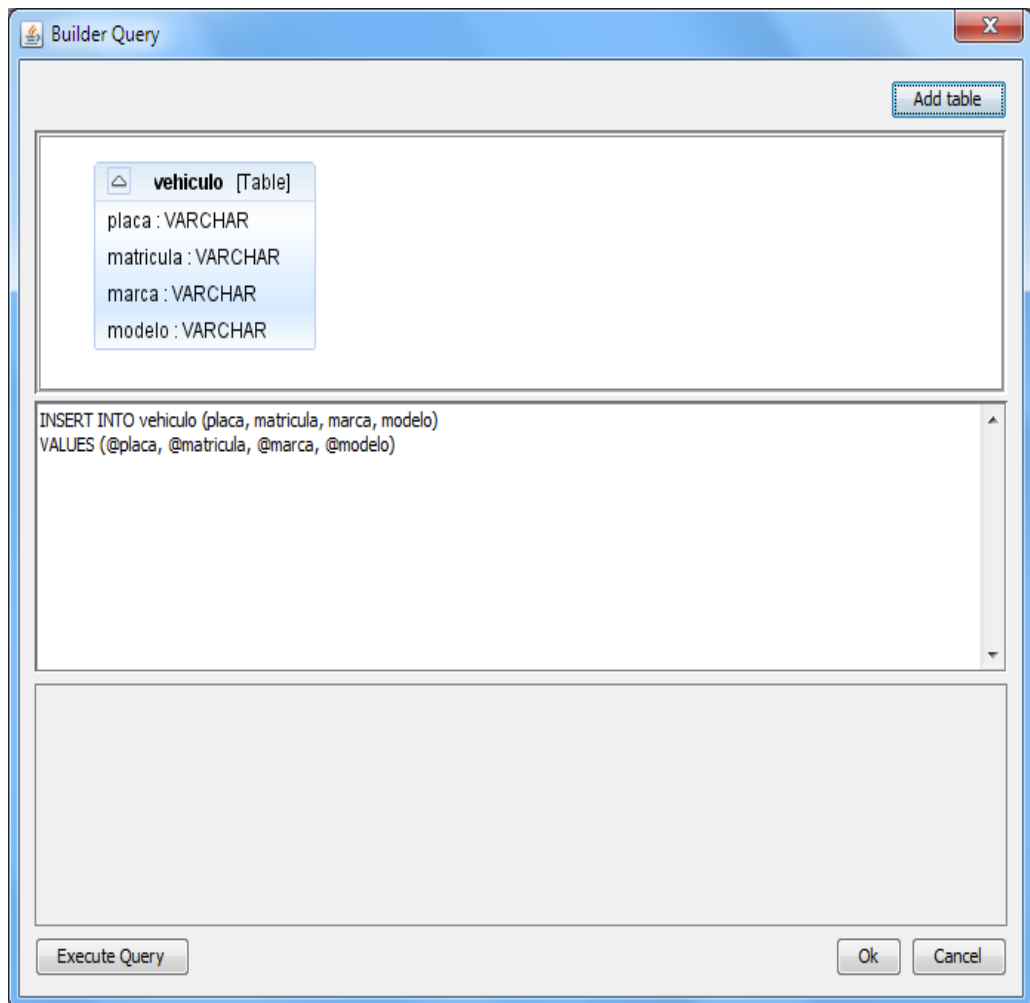
Primeramente escogemos la opción Add Query, esta opción permitirá construir una sentencia según el requerimiento del sistema a desarrollar, en este caso vamos a realizar una inserción en la tabla vehículo. Al dar clic en Add Query nos mostrará el siguiente wizard:



En la primera página escogemos el tipo de sentencia INSERT. Luego damos clic en Next.

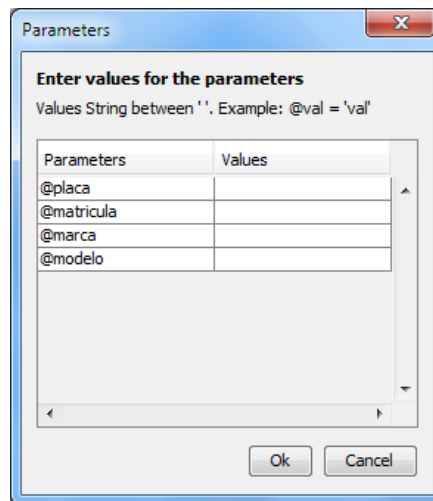


En la siguiente página nos pide que especifiquemos la sentencia a construir, aquí nos da una muestra de cómo podemos armar la sentencia INSERT utilizando parámetros. Para definir una sentencia como realmente lo exigimos y luego sea probada, damos clic en el botón Query Builder, nos mostrará la siguiente ventana:



En este momento el programador depende de sus capacidades intelectuales para armar bien una sentencia de inserción de datos según la sintaxis de MySQL. En este caso la tabla vehículo contiene los campos placa, matrícula, marca y modelo, todos de tipo VARCHAR, por lo tanto no modificaremos la sentencia INSERT.

Ahora debemos probar la sentencia dando clic en el botón Execute Query. Si la sentencia ha sido armada con parámetros nos mostrará el siguiente cuadro:



En este cuadro introduciremos los valores que deseemos dar a cada parámetro.

En esta parte existe un pequeño detalle que el programador debe admitir, ya que este proyecto JDataSet es una primera versión de la funcionalidad que realmente se desea dar en el procedimiento de acceso a datos.

El ingreso de valores depende del campo al que esté asociado el parámetro, es decir que, si el campo es de tipo VARCHAR tendremos que ingresar el valor entre comillas simples. En este caso, escribimos como valor 'QTR-546' para parámetro @placa, pues este parámetro está asociado al campo placa de la tabla Vehiculo teniendo un tipo de dato VARCHAR(7).

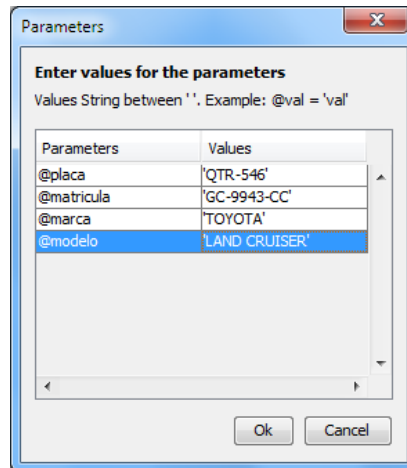
En otro caso, si el parámetro estuviere asociado a un campo de tipo INT, el valor solo sería un número entero sin comillas. Para una mejor comprensión analiza el siguiente cuadro.

ASOCIACIÓN DEL PARÁMETRO	VALOR
Si el parámetro estuviere asociado a un campo tipo VARCHAR(10)	'GC-1234-RE'
Si el parámetro estuviere asociado a un campo tipo INT	5
Si el parámetro estuviere asociado a un campo tipo DATETIME	'2014-02-06 12:05:21'
Si el parámetro estuviere asociado a un campo tipo DOUBLE	2.25

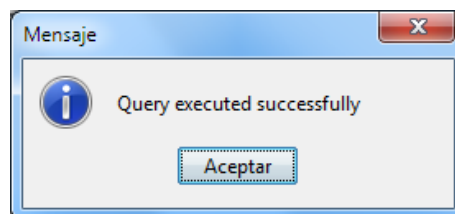
Es decir, el valor a ingresar depende del tipo de dato del campo al que este asociado el parámetro, tal y como si estuviésemos armando una sentencia SQL

pura, pues lo que internamente sucede en esta operación es reemplazar el parámetro @placa por el valor dado 'QTR-546'.

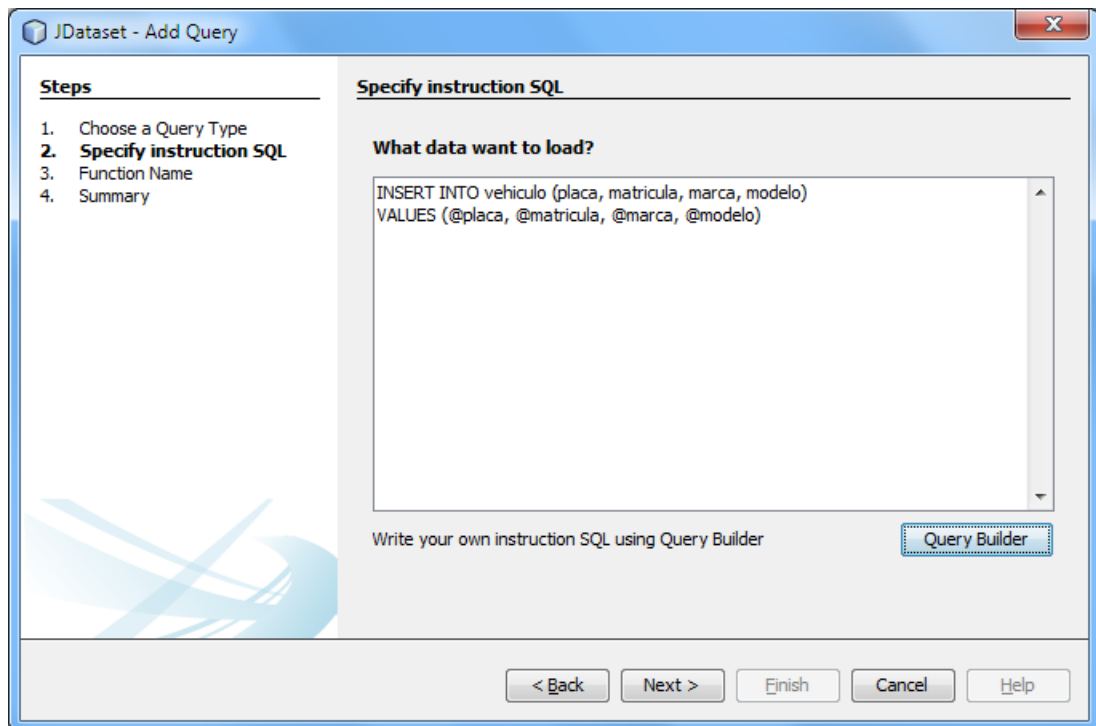
Bueno! una vez explicada esta parte realizamos el ingreso de valores para cada parámetro:



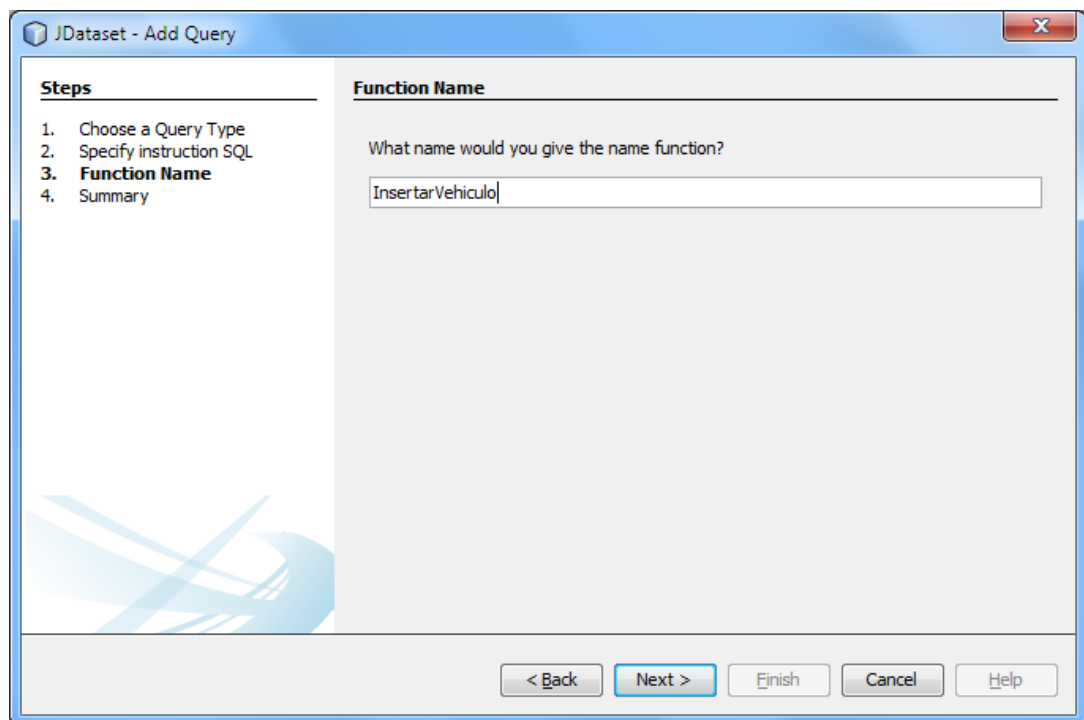
Una vez llenada las celdas con los valores correspondientes damos clic en Ok. Si no existe ningún error en el ingreso de datos mostrará el siguiente cuadro:



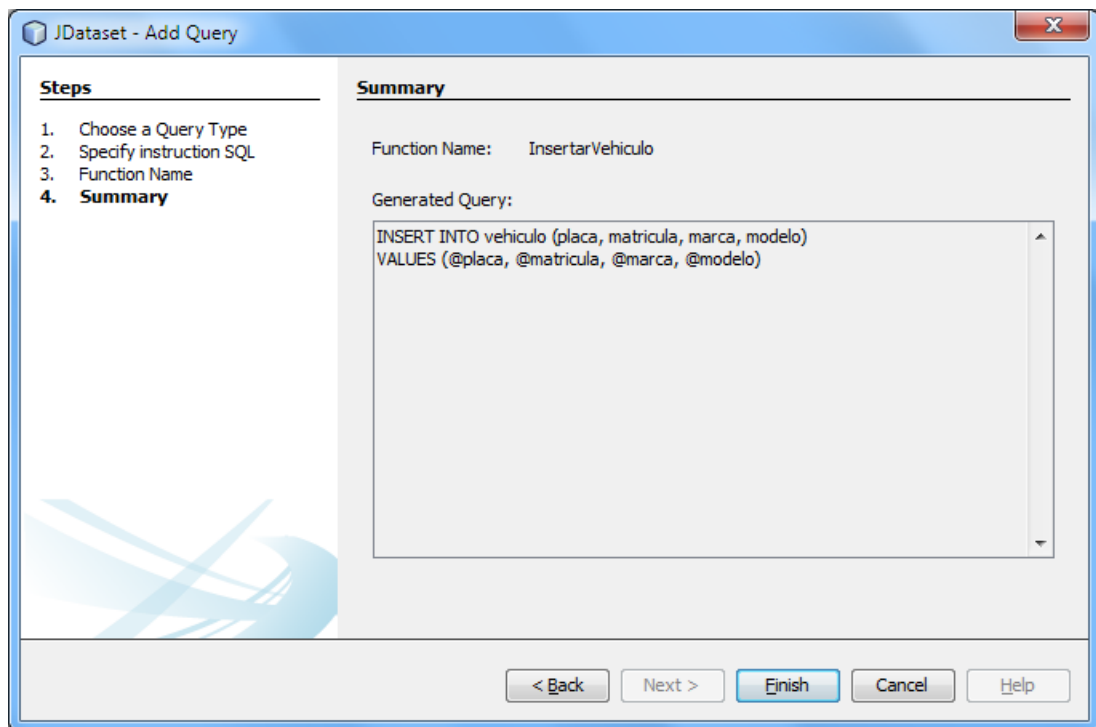
Este mensaje nos da a entender que los datos han sido insertados correctamente, o sea que la sentencia se ha ejecutado de una manera satisfactoria. Ahora en el cuadro Query Builder damos clic en Ok. En el caso que hubiésemos modificado la sentencia la página del wizard se actualizaría con la nueva sentencia construida.



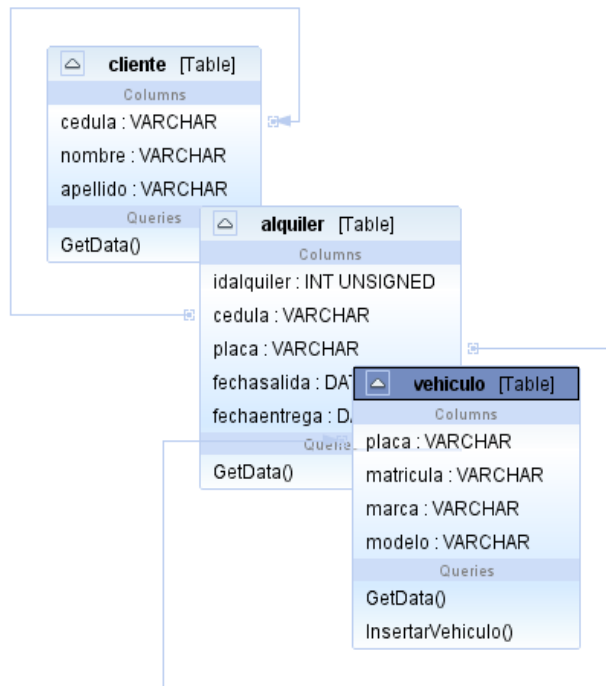
Damos clic en Next, y en la siguiente página nos pide un nombre para la función, este debe ser según las condiciones con que hemos armado la sentencia y solo debe contener letras sin espacios, en este caso le pondré InsertarVehiculo.



Damos clic en Next y en la última página muestra un resumen del procedimiento ya realizado, el nombre de la función y la sentencia generada.



Finalmente damos clic en Finish. Cerrado el wizard no mostrará ninguna modificación en el diagrama de la ventana Visual del archivo JDsConcesionaria.jds. Como lo mencionamos anteriormente tenemos que dar clic en cualquier otra ventana editor de los archivos que tengamos abiertos, y nuevamente dar clic en la ventana Visual del archivo JDsConcesionaria.jds para que el diagrama se actualice mostrando el nombre de la nueva función agregada. Tal y como se muestra en la siguiente figura.



De la misma manera el archivo JDsConcesionaria.xml ha sufrido una modificación, agregando al nodo de la tabla Vehiculo un nuevo nodo función nombrado InsertarVehiculo(), tal y como se muestra en la siguiente figura. Si el archivo JDsConcesionaria.xml estaba abierto te mostrará un mensaje para que recargues el contenido del archivo.

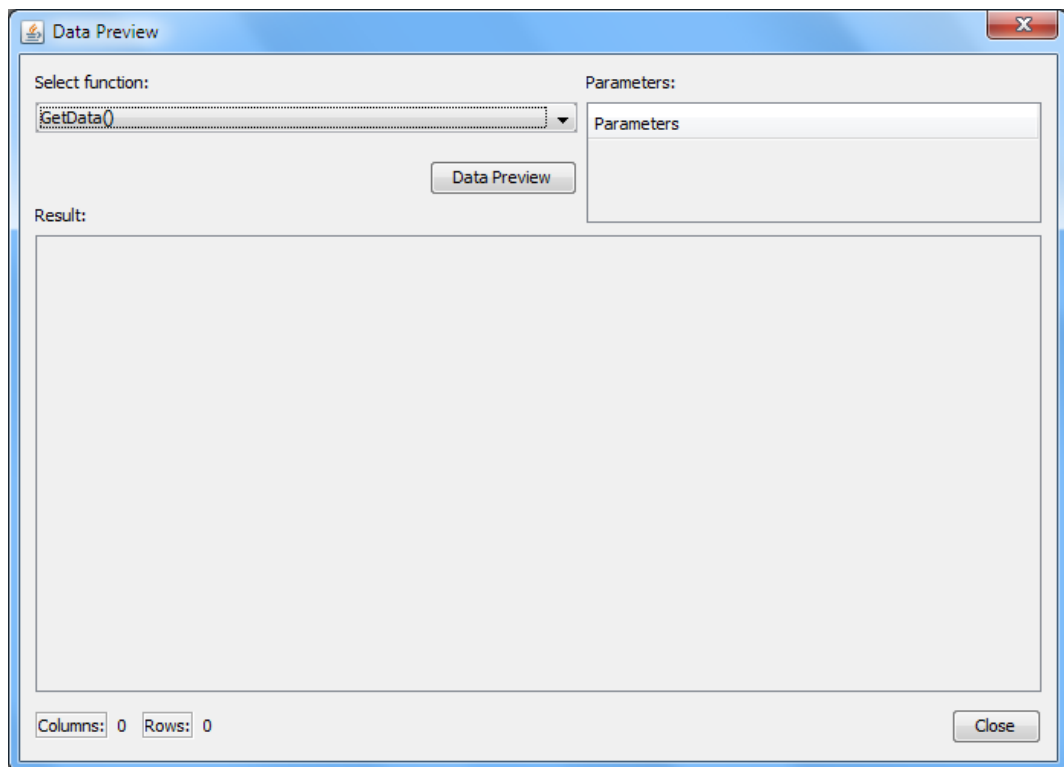
```

<table name="vehiculo">
  <column type="VARCHAR">placa</column>
  <column type="VARCHAR">matricula</column>
  <column type="VARCHAR">marca</column>
  <column type="VARCHAR">modelo</column>
  <relation>
    <foreingkey>alquiler:placa</foreingkey>
    <primarykey>vehiculo:placa</primarykey>
  </relation>
  <function name="GetData()">select *from vehiculo</function>
  <function name="InsertarVehiculo()">INSERT INTO vehiculo (placa, matricula, marca, modelo)
    VALUES (@placa, @matricula, @marca, @modelo)</function>
</table>

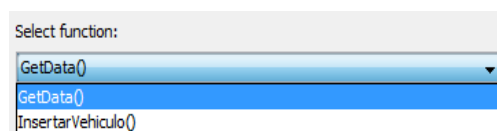
```

De esa manera finaliza la tarea Add Query. Ahora como comprobación de lo que hemos realizado, vamos a la segunda opción de este asistente gráfico de acceso a datos JDataSet.

Damos clic derecho sobre la tabla Vehículo y seleccionamos la opción Data Preview, nos mostrará el siguiente cuadro.



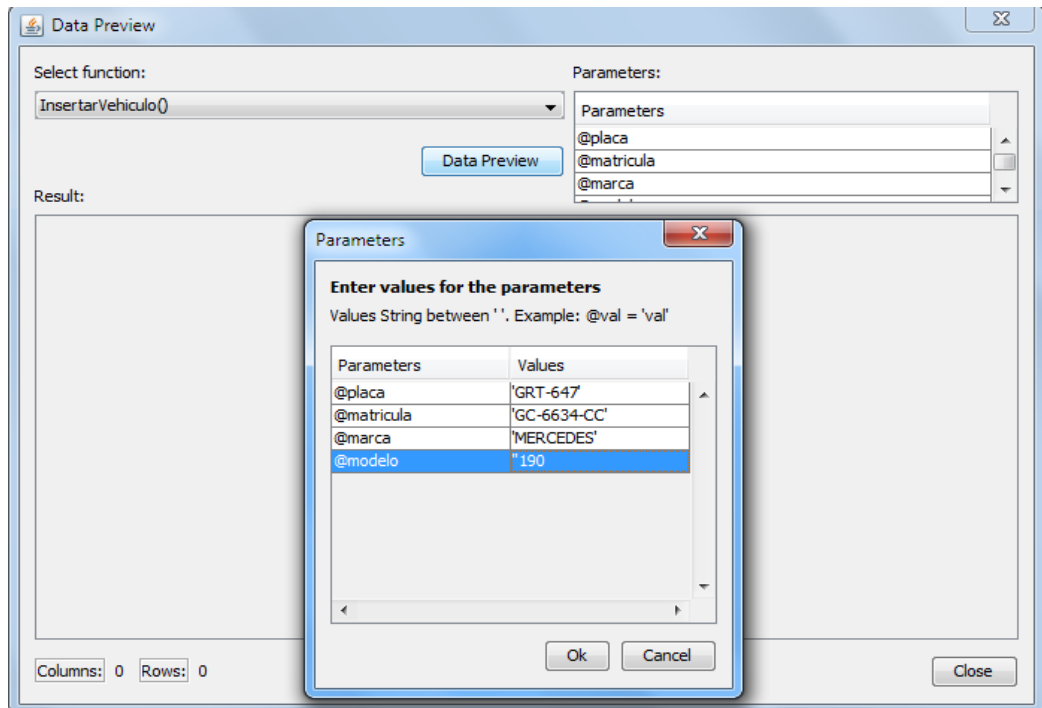
Aquí vamos a poder tener una vista previa de las sentencias ya realizadas anteriormente en la opción Add Query. En primer lugar desplegamos el combo para visualizar la lista de funciones disponibles.



Observamos que existen dos sentencias. GetData() selecciona todos los datos de la tabla según la siguiente sentencia SELECT *FROM VEHÍCULO, este ya viene por defecto.

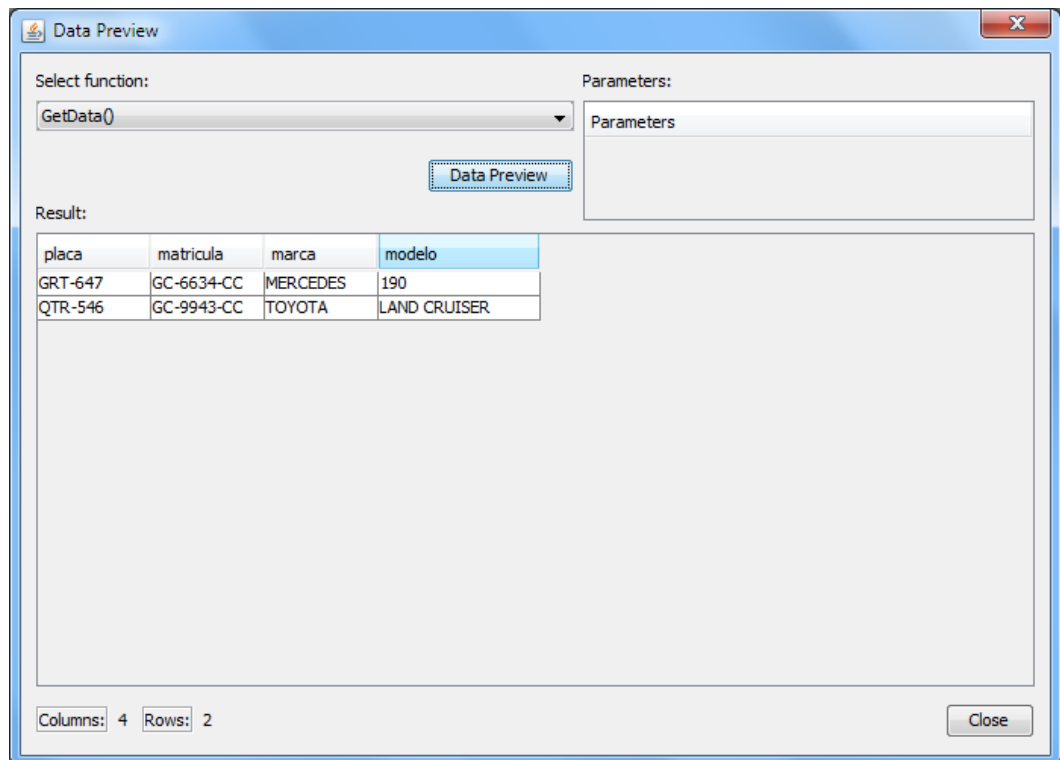
Y también está la sentencia que acabamos de construir InsertarVehículo(), el cual lo seleccionamos. Una vez seleccionado se actualiza la tabla Parameters que se encuentra a la derecha, este muestra la lista de parámetros que la sentencia contiene y a los cuales tenemos que darle un valor. En esta ventana podemos probar cada una de las sentencias que han sido agregadas a cada

tabla. A continuación damos clic en el botón Data Preview. Como ésta sentencia contiene parámetros, nos mostrará el cuadro para ingresar los valores.



A continuación muestra un mensaje "Query executed successfully", que quiere decir que la sentencia fue ejecutada exitosamente.

Para comprobar este suceso ahora escogemos la sentencia GetData() del combo, damos clic en el botón Data Preview, como mencionamos anteriormente este mostrará todos los datos de la tabla, en este caso solo nos mostrará los dos registros insertados. En la parte inferior de la ventana Data Preview se muestran el número de columnas y filas que representa la consulta de datos actual.



Vemos que el número de columnas es de cuatro y el número de filas es de dos, esto nos servirá cuando necesitemos llamar a una sentencia de tipo SELECT en nuestro proyecto, pero eso lo veremos más adelante.

Ahora podemos entender de una mejor manera las tareas de este componente `JDataSet`, pero de nada serviría hacer todo este procedimiento sin poder adaptarlos para el desarrollo de nuestro proyecto web en Java. En la siguiente sección explicaré cómo trabajar en el código JAVA para llamar el procedimiento realizado.

5.- Uso de la librería `jdataset-lib` para el llamado de funciones

Ahora vamos a trabajar sobre la clase Vehículo creada anteriormente. En esta clase debemos aprovechar toda la información del archivo `JDsConcesionaria.xml`, en el cual se guardan todas las funciones con las sentencias realizadas en el asistente gráfico, y como mencionamos antes, de este archivo XML depende todo el funcionamiento del `JDataSet`.

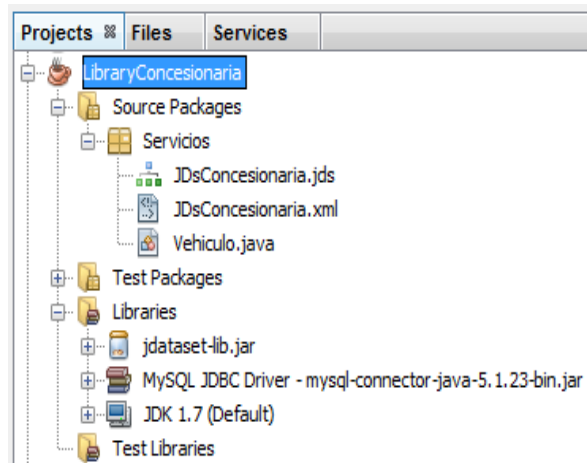
Para esto tenemos que descargar el archivo `jdataset-lib.jar`. Como lo mencionamos anteriormente, este lo podemos descargar del siguiente link.

Una vez descargado lo agregamos a nuestro proyecto LibraryConcesionaria de la siguiente manera.

Damos clic derecho en el nodo Libraries del proyecto, seleccionamos Add JAR/Folder, nos mostrará un cuadro de dialogo en donde tenemos que buscar el directorio del archivo jdataset-lib.jar y agregarlo.

El archivo jdataset-lib.jar es una librería de clases que trabaja con el conector de MySQL, de esa manera, también tenemos que agregar el driver de MySQL. Si quieres lo puedes descargar desde el sitio web de MySQL, pero las últimas versiones de Netbeans ya lo traen contenido dentro la carpeta de instalación.

Este driver lo agregamos dando clic derecho sobre el nodo Libraries, seleccionamos Add Library.., nos mostrará una ventana en donde Netbeans tiene empaquetado varios archivos JAR, nosotros tipiaremos MySQL, y escogemos MySQL JDBC Driver, y damos clic en Add Library. Para este momento debemos de tener la siguiente vista en el proyecto LibraryConcesionaria.



Abrimos la clase Vehiculo y agregamos el siguiente código.

```

public class Vehiculo {

    public String mensaje;

    public int insertarVehiculo(String placa, String matricula, String marca, String modelo){
        int filasafectadas = 0;
        try {
            String path = "C:\\Users\\Daniel\\Documents\\NetBeansProjects\\LibraryConcesionaria"
                + "\\src\\Servicios\\JDsConcesionaria.xml";
            JDataset ds1 = new JDataset(path);
            ds1.SetFunctionName("InsertarVehiculo()");
            ds1.SetParameterValue("@placa", "" + placa + "");
            ds1.SetParameterValue("@matricula", "" + matricula + "");
            ds1.SetParameterValue("@marca", "" + marca + "");
            ds1.SetParameterValue("@modelo", "" + modelo + "");
            filasafectadas = ds1.executeUpdate();
        } catch (ParserConfigurationException ex) {
            mensaje = ex.getMessage();
        } catch (IOException ex) {
            mensaje = ex.getMessage();
        } catch (SAXException ex) {
            mensaje = ex.getMessage();
        } catch (SQLException ex) {
            mensaje = ex.getMessage();
        } catch (ClassNotFoundException ex) {
            mensaje = ex.getMessage();
        }
        return filasafectadas;
    }
}

```

Antes del código mostrado tenemos que agregar la siguiente importación: `import org.jdataset.JDataset`. Como podemos observar en el código, hacemos uso de la clase `JDataset` contenido en el archivo `jdataset-lib.jar` agregado.

Primero comenzamos creando un método el cual tiene por nombre `insertarVehiculo` con sus respectivos parámetros `placa`, `matricula`, `marca`, `modelo` todos tipo `String` y del cual retorna un valor tipo `int`.

Luego de eso creamos un objeto `JDataset ds1`, el cual nos pide el `path` o directorio en que se encuentra el archivo XML creado por el `JDataSet`, en este caso sería:

`C:\\Users\\Daniel\\Documents\\NetBeansProjects\\LibraryConcesionaria\\src\\Servicios\\JDsConcesionaria.xml`

Podremos este directorio si queremos llamar desde donde se crea la aplicación, en mi caso esta aplicación se encuentra en mi carpeta Documents dentro de una carpeta NetbeansProjects. Pero al momento de subirla a un servidor web no funcionará ya que el directorio tendría que cambiar rotundamente. Este problema lo estaré aclarando más adelante.

Después de instanciar un objeto `JDataSet`, en este caso le hemos puesto como variable `ds1`, llamamos al método `ds1.SetFunctionName` el cual pide el nombre de la función a llamar tipo `String`, aquí escribimos `"InsertarVehiculo()"` entre comillas dobles porque es tipo `String`, y el nombre de la función finalizado por paréntesis de abierto y cerrado.

Ahora tenemos que agregar cada uno de sus parámetros en el caso que los contenga, escribimos `ds1.SetParameterValue("@placa", "" + placa + "")` el cual nos pide primero el nombre del parámetro en este caso escribimos `"@placa"` y en segundo lugar el valor que le vamos a dar, como se explicó anteriormente el valor del parámetro depende del tipo de dato del campo al que este asociado, es decir, si el campo al que está asociado el parámetro es de tipo `VARCHAR` tendríamos que escribir el valor `" + placa + "` entre comillas dobles porque es un tipo `String` y dentro de comillas simples con la variable `placa` entre signos `+`. Hacemos lo mismo para dar valor a todos los parámetros.

Finalmente llamamos al método `ds1.executeUpdate()` el cual sirve para ejecutar las sentencias de tipo `INSERT`, `UPDATE` y `DELETE`. Este retorna un valor tipo entero el cual especifica el número de filas afectadas. Este valor lo va a recibir la variable `filasafectadas` ya declarado anteriormente.

Si podemos observar todo este bloque de código se encuentra dentro de un `try catch` con diferentes excepciones, estas deben ir de obligación ya que al encontrar un error disparan un mensaje de especificación del error, estos pueden ser por mala sintaxis, o por no encontrar el archivo XML según el directorio especificado, o por no encontrar los archivo JAR que deben estar en el proyecto.

6.- Llamado de datos desde una aplicación web

Una vez lista la clase Vehículo del proyecto LibraryConcesionaria, crearemos una aplicación web llamada WebConcesionaria, al cual le agregamos un archivo JSP con el nombre agregarvehiculo.jsp

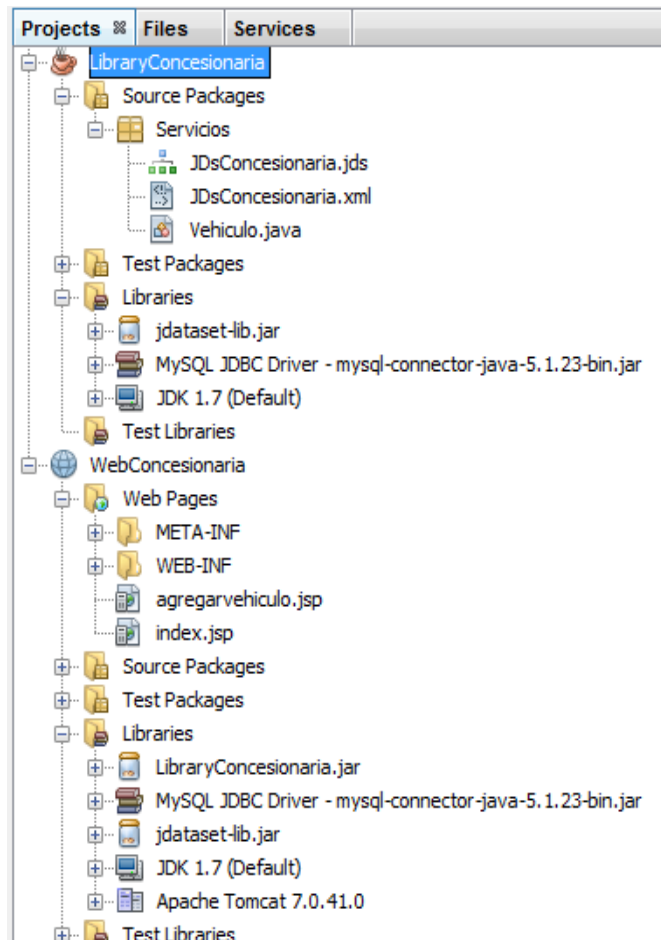
Ahora tenemos que agregar el proyecto LibraryConcesionaria como un archivo LibraryConcesionaria.jar a la aplicación WebConcesionaria.

Para lograr que la proyecto LibraryConcesionaria se convierta en un archivo JAR LibraryConcesionaria.jar, damos clic derecho sobre el nodo principal del proyecto LibraryConcesionaria y seleccionamos Clean and Build, de esa manera se crea un archivo LibraryConcesionaria.jar dentro de la carpeta dist de este proyecto.

Ahora en la aplicación WebConcesionaria damos clic derecho sobre el nodo Libraries, seleccionamos Add JAR/Folder, en el cuadro de dialogo buscamos el directorio en donde se encuentra el archivo LibraryConcesionaria.jar y lo agregamos.

También tenemos que agregar el archivo jdataset-lib.jar y el conector MySQL JDBC Driver a la aplicación WebConcesionaria, del mismo modo como ya lo habíamos explicado antes.

Deberíamos tener la siguiente vista de los proyectos LibraryConcesionaria y WebConcesionaria:



En el archivo agregarvehiculo.jsp agregamos el siguiente código HTML.

```

<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <title>Agregar vehículo</title>
  </head>
  <body>
    <h4>Ingresa nuevo vehículo</h4>
    <form name="frmagregarvehiculo" action="index.jsp" method="POST">
      Placa:<input type="text" name="txtplaca" value="" size="20" />
      Matricula:<input type="text" name="txtmatricula" value="" size="20" />
      Marca:<input type="text" name="txtmarca" value="" size="20" />
      Modelo:<input type="text" name="txtmodelo" value="" size="20" />
      <input type="submit" value="Guardar" name="btnguardar" />
    </form>
  </body>
</html>

```

Lo que hicimos en el archivo agregarvehiculo.jsp fue agregar un formulario, dentro de este formulario agregamos cuatro cuadros de textos para ingresar los

valores de placa, matrícula, marca y modelo, también un botón para guardar los datos ingresados.

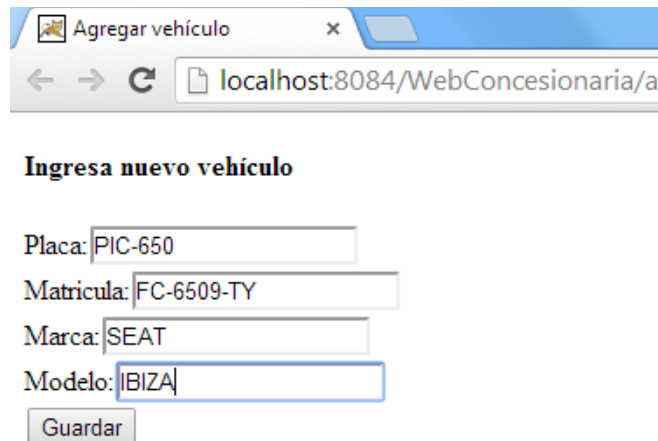
Como el formulario tiene un action a la página index.jsp que ya viene por defecto al crear un proyecto web, en esta página index.jsp escribiremos el siguiente código JAVA.

```
<%@page import="Servicios.Vehiculo"%>
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <title>JSP Page</title>
  </head>
  <body>
    <%
      String placa = request.getParameter("txtplaca");
      String matricula = request.getParameter("txtmatricula");
      String marca = request.getParameter("txtmarca");
      String modelo = request.getParameter("txtmodelo");

      Vehiculo v = new Vehiculo();
      int f = v.insetarVehiculo(placa, matricula, marca, modelo);
      String mensaje = "";
      if (f > 0){
        mensaje = "Datos del Vehiculo han sido guardados";
      }
      else{
        mensaje = v.mensaje;
      }
    %>
    <p><%=mensaje%></p>
  </body>
</html>
```

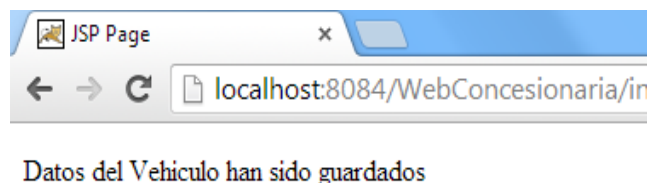
En esta página JSP recibimos los parámetros txtplaca, txtmatricula, txtmarca, txtmodelo enviados desde el formulario frmagregarvehiculo del archivo agregarvehiculo.jsp. Luego creamos un objeto de la clase Vehiculo v, el cual se encuentra en el archivo LibraryConcesionaria.jsp creado anteriormente. Llamamos al método v.insetarVehiculo en el cual enviamos como parámetros placa, matricula, marca y modelo. Luego condicionamos el valor entero que retorna de este método, si es mayor que 0 es porque realmente existen filas afectadas. Finalmente mostramos el mensaje de éxito o de error que es enviado desde la clase Vehiculo.

Creo que ahora la aplicación está lista para ser ejecutada e insertar un nuevo vehículo. Para ejecutar esta página damos clic derecho sobre el archivo agregarvehiculo.jsp, seleccionamos Run File para que la aplicación se ejecute desde esta página. En este ejemplo se tiene configurado para que se ejecute en el servidor web Apache Tomcat. Se ejecutará según el browser que tengas como predeterminado, se mostrará como en la siguiente figura.

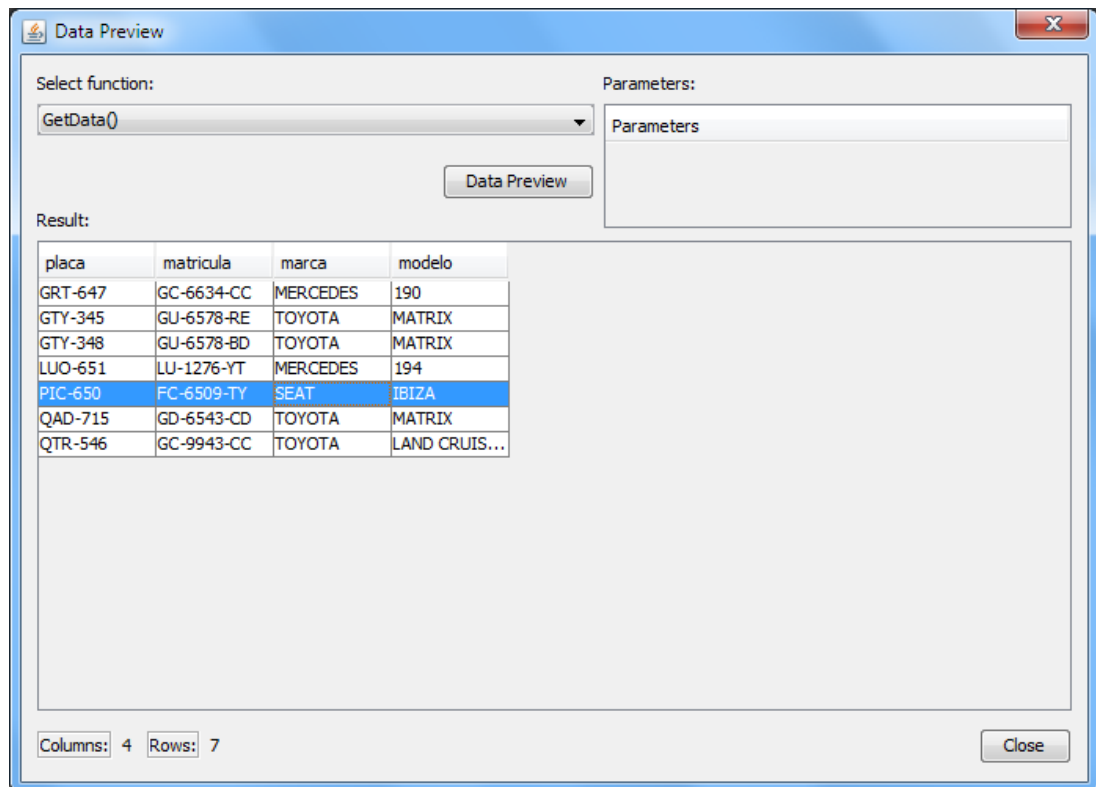


The screenshot shows a web browser window with the title 'Agregar vehículo'. The address bar displays 'localhost:8084/WebConcesionaria/a'. Below the browser, the page content includes the heading 'Ingresa nuevo vehículo' and a form with the following fields: 'Placa:' with the value 'PIC-650', 'Matricula:' with the value 'FC-6509-TY', 'Marca:' with the value 'SEAT', and 'Modelo:' with the value 'BIZA'. A 'Guardar' button is located at the bottom of the form.

Ingresamos los valores del vehículo, y damos clic en el botón Guardar. Si todo sale bien, veremos el siguiente mensaje, caso contrario el mensaje de error.



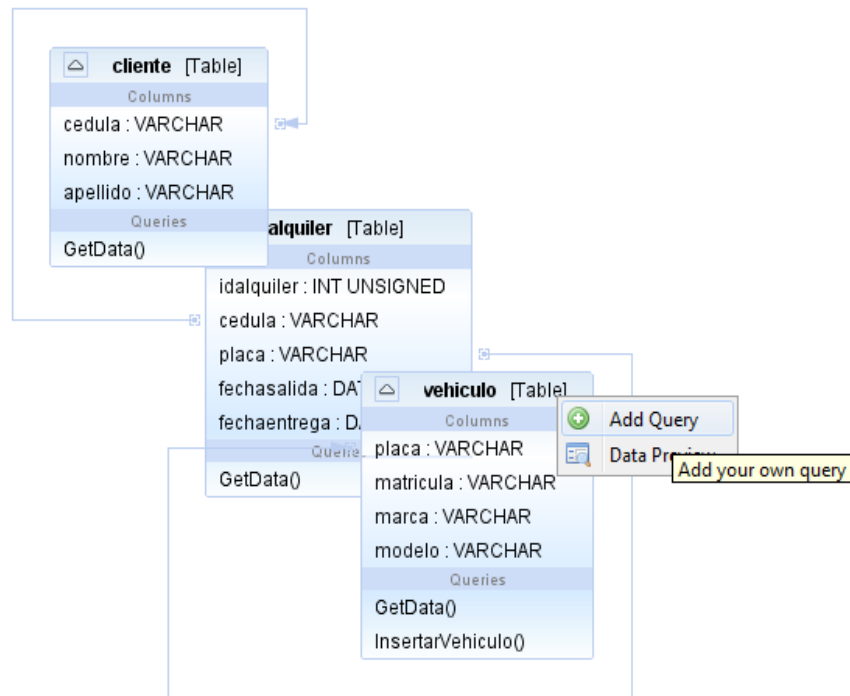
Para comprobar si los datos fueron insertados correctamente, vamos a la ventana Visual del JDsConcesionaria.xml, en la tabla Vehiculo, damos clic derecho, seleccionamos Data Preview, en la ventana que se muestra dejamos la opción GetData() del combo y damos clic en el botón Data Preview, nos mostrará el registro recién añadido, como vemos en la siguiente figura.



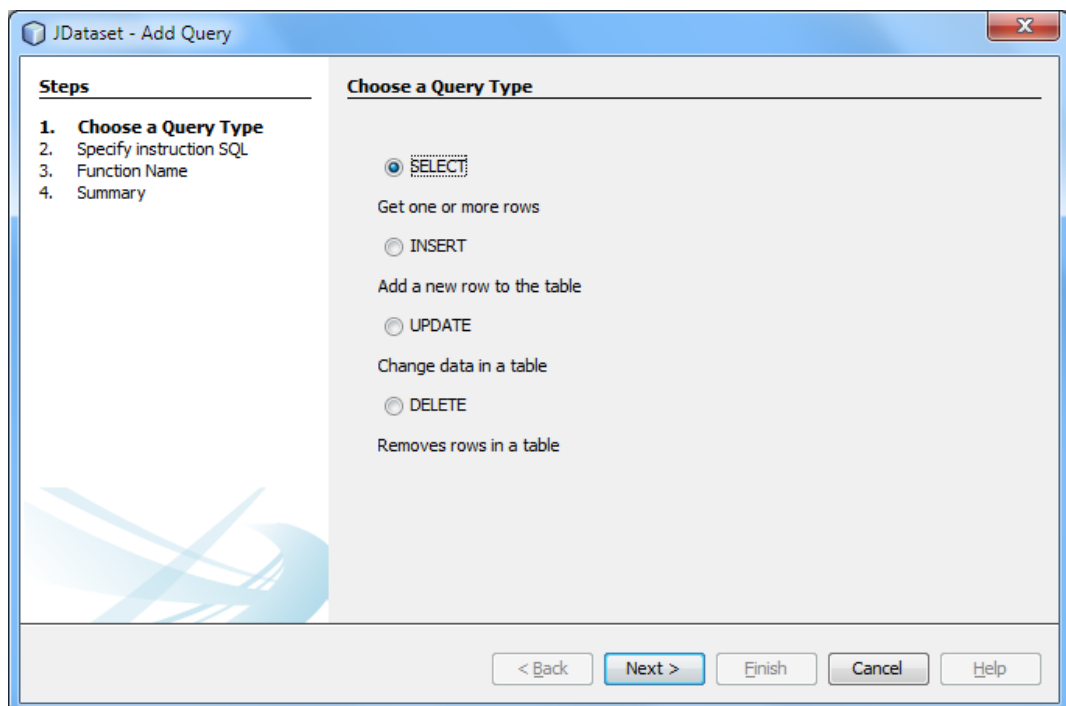
Como vemos, este componente también nos evita la verificación de los datos dentro del Gestor de base de datos, permitiendo que los datos sean observados desde el IDE.

7.- Realizar una sentencia de tipo SELECT en el JDataSet

Ahora realizaremos una sentencia de tipo SELECT y crearemos una página JSP donde el usuario pueda consultar los vehículos según la marca. Nuevamente escogemos la opción Add Query de la tabla vehículo en la ventana Visual del JDsConcesionaria.jds.



En la primera página del wizard seleccionamos el tipo de sentencia SELECT.



Luego damos clic en Next, y en la siguiente página damos clic en el botón Query Builder para que nos muestre la ventana de construcción de sentencias, aquí modificaremos la sentencia de la siguiente manera.

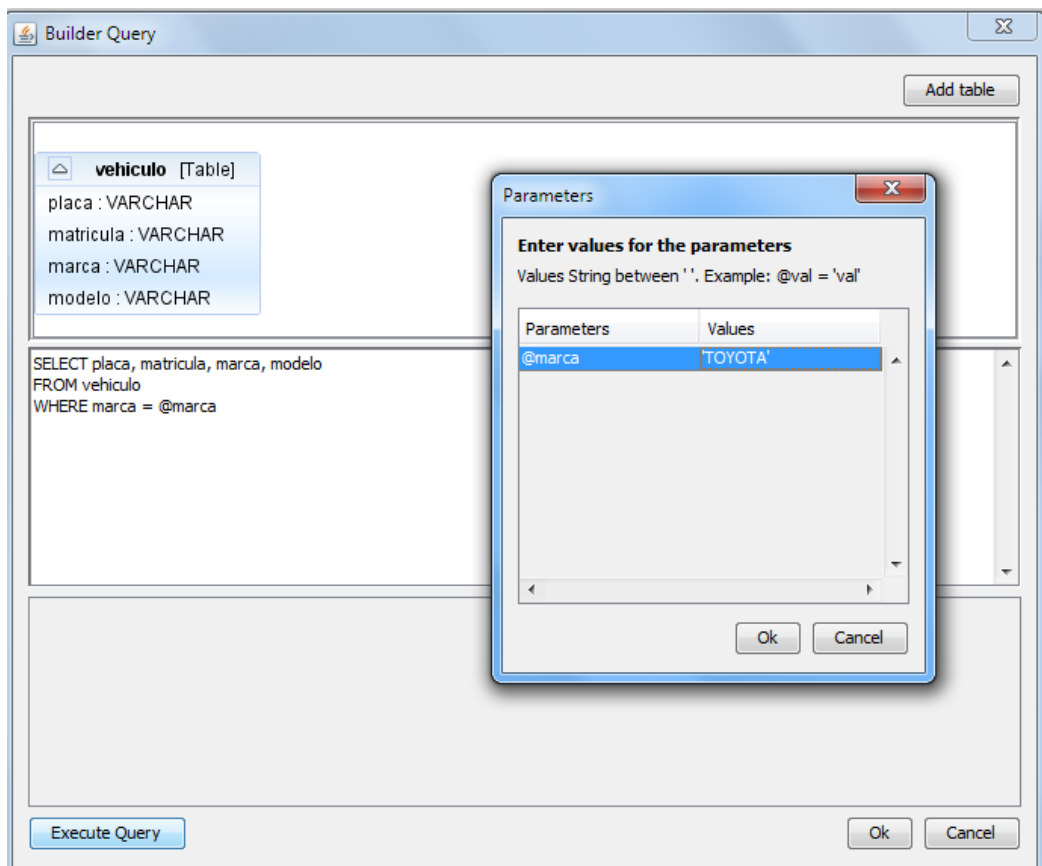
SELECT placa, matricula, marca, modelo

FROM vehiculo

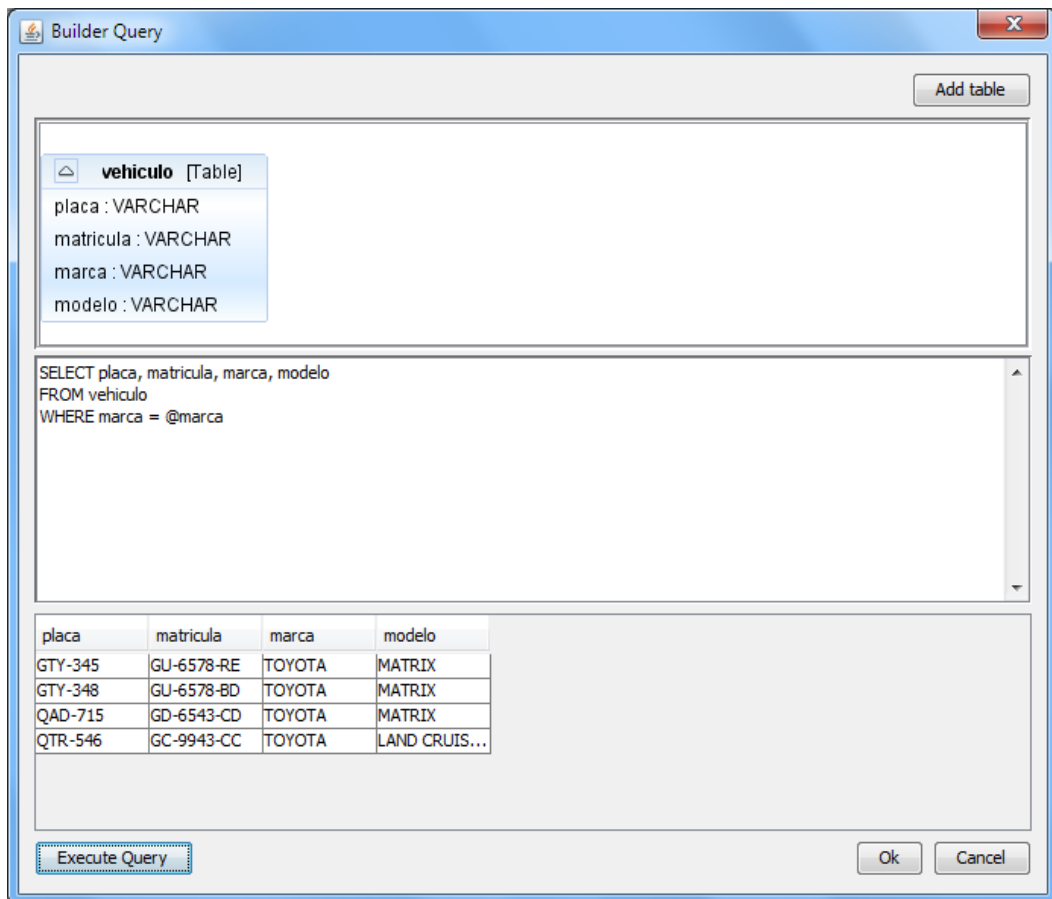
WHERE marca = @marca

Donde se tendrá que ingresar la marca para obtener una lista de todos los vehículos registrados con esa marca.

Damos clic en el botón Execute Query, como la sentencia contiene un parámetro nos saldrá el cuadro para ingresar el valor al parámetro @marca.



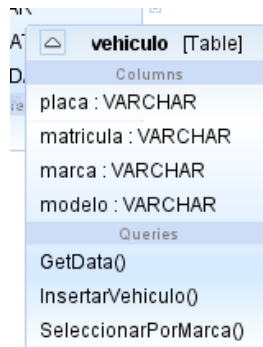
En este caso nos mostrará el siguiente resultado.



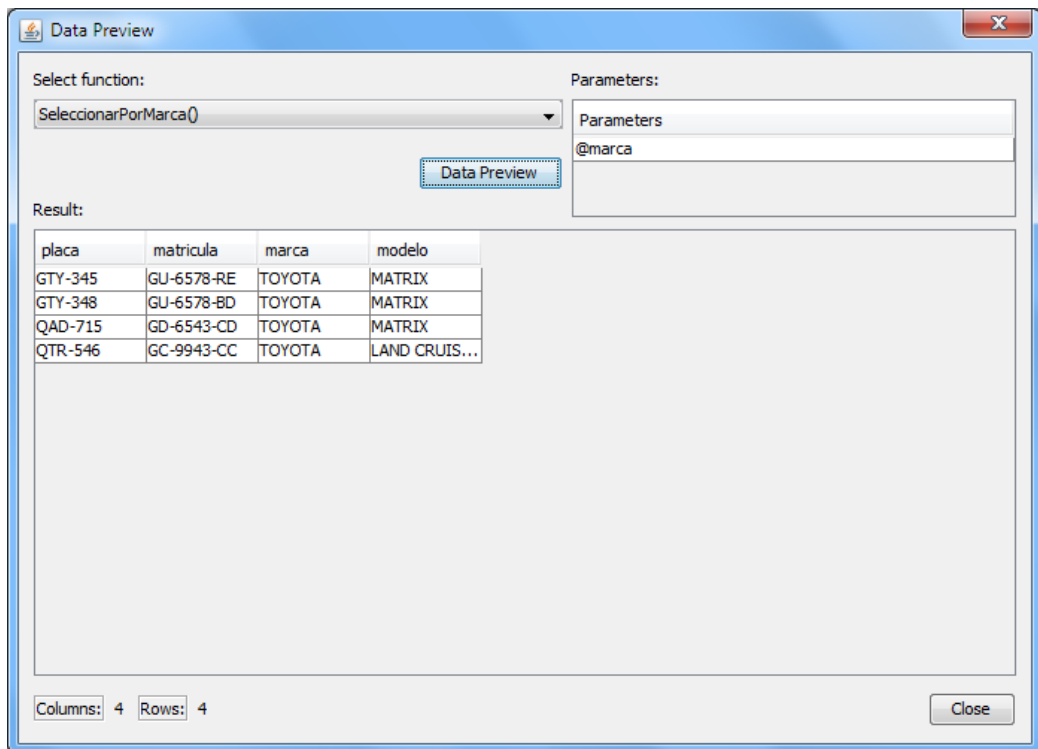
Esto quiere decir que la sentencia se ejecutó exitosamente. Aquí nos muestra la consulta de los datos según la marca ingresada 'TOYOTA'. Ahora damos clic en Ok.

La ventana del wizard se actualizará con la nueva sentencia SELECT modificada. En la siguiente página le damos un nombre a la nueva función agregada, en este caso le pondré SeleccionarPorMarca, seguimos a la siguiente página y finalmente damos clic en Finish.

Actualizamos la ventana Visual del JDsConcesionaria.jds dando clic en cualquier otra ventana editor, y nuevamente dando clic en la ventana Visual del JDsConcesionaria.jds, tal y como ya lo había explicado antes. Veremos la nueva función SeleccionarPorMarca() en la tabla vehículo.



Vamos a la opción Data Preview de la tabla vehículo, y una vez seleccionada la función `SeleccionarPorMarca()` en el combo, damos clic en el botón Data Preview, ingresamos el valor al parámetro `@marca` en el cuadro correspondiente, y nos mostrará el resultado de la siguiente manera.



Como podemos darnos cuenta en la parte inferior de esta ventana tenemos una cantidad de cuatro columnas y cuatro filas, esto nos servirá de mucho porque esta vez vamos a recibir los datos en una matriz tipo `String[][]` dentro de la clase `Vehiculo` del proyecto `LibraryConcesionaria`.

Vamos al código Java. En la clase `Vehiculo` agregamos el siguiente código.

```

public String[][] todoslosVehiculos(String marca){
    String[][] matriz = null;
    try {
        String path = "C:\\Users\\Daniel\\Documents\\NetBeansProjects\\LibraryConcesionaria"
            + "\\src\\Servicios\\JDsConcesionaria.xml";
        JDataset ds1 = new JDataset(path);
        ds1.SetFunctionName("SeleccionarPorMarca()");
        ds1.SetParameterValue("@marca", "" + marca + "");
        matriz = ds1.executeQuery();
    } catch (ParserConfigurationException ex) {
        mensaje = ex.getMessage();
    } catch (IOException ex) {
        mensaje = ex.getMessage();
    } catch (SAXException ex) {
        mensaje = ex.getMessage();
    } catch (SQLException ex) {
        mensaje = ex.getMessage();
    } catch (ClassNotFoundException ex) {
        mensaje = ex.getMessage();
    }
    return matriz;
}

```

Esta vez agregamos un método llamado todoslosVehiculos(String marca) con un parámetro, para establecer el valor al parámetro @marca. Hacemos lo mismo que el método explicado anteriormente. Esta vez enviando el nombre de la función SeleccionarPorMarca(), y un parámetro @marca, a diferencia de la siguiente línea:

```
matriz = ds1.executeQuery();
```

Esta vez utilizamos el método ds1.executeQuery(), este método sirve para ejecutar sentencias de tipo SELECT, del cual retornará una matriz tipo String[][] con los datos de la consulta.

Ahora en la aplicación WebConcesionaria agrega una página JSP para llamar al método anterior. Se puso como nombre todosvehiculos.jsp, en el cuerpo de esta página añade el siguiente código.

```

<title>Todos los Vehiculos</title>
</head>
<body>
  <h4>Todos los vehiculos</h4>
  <form name="frmtodosvehiculos" action="todosvehiculos.jsp" method="POST">
    Ingrese marca:<input type="text" name="txtmarca" value="" size="20" />
    <input type="submit" value="Mostrar lista" name="btnmostrar" />
    <%
      String[][] matriz;
      Vehiculo ve = new Vehiculo();
      String marca = request.getParameter("txtmarca");
      matriz = ve.todoslosVehiculos(marca);
    %>
    <br/>
    <table border="1">
      <%
        for (int i = 0; i < matriz.length; i++){
      %>
      <tr>
        <%
          for (int j = 0; j < matriz[i].length; j++){
        %>
        <td><font family="Calibri" color="blue" size="4"><%=matriz[i][j]></font></td>
        <%
          }
        %>
      </tr>
      <%
        }
      %>
    </table>
  </form>
</body>
</html>

```

Según este código, en la página todosvehiculos.jsp se ha agregado un formulario que tiene un action hacia esta misma página. En la primera parte del formulario se establece un cuadro de texto txtmarca donde se ingresará el valor de la marca según queramos consultar, y un botón para enviar el valor del cuadro de texto.

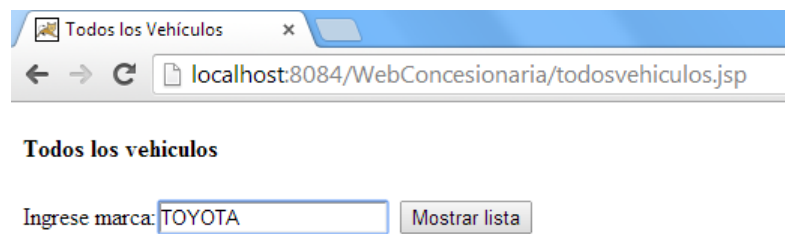
Cuando el botón haya sido pulsado se volverá a cargar la página pero esta vez recibiendo el valor de la marca, el cual llevará a ejecutar según el método todoslosVehiculos(marca) creado anteriormente, retornando una matriz con los datos consultados.

Esta matriz que retorna del método todoslosVehiculos(marca) inicializa las dimensiones de la variable matriz tipo String[[[]], estableciendo las columnas y filas de la matriz, más abajo codificamos una tabla y le damos los valores a cada celda.

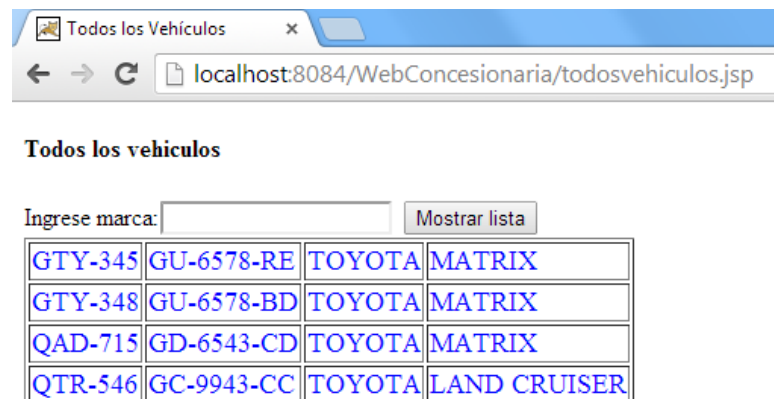
Creo que ahora ya podemos ejecutar la página todosvehiculos.jsp, pero antes actualiza al archivo LibraryConcesionaria.jar del proyecto WebConcesionaria, ya que hemos agregado un nuevo método a la clase Vehiculo del proyecto LibraryConcesionaria, si ejecutas la aplicación web sin actualizarlo la ejecución va a fallar por que no encuentra el método todoslosVehiculos(marca).

Entonces tenemos que ir al nodo Libraries, damos clic derecho sobre LibraryConcesionaria.jar y seleccionamos Remove, el archivo será eliminado, ahora damos clic derecho sobre Libraries, seleccionamos Add JAR/Folder y buscamos el archivo LibraryConcesionaria.jar y lo agregamos.

Ahora, damos clic derecho sobre el archivo todosvehiculos.jsp, seleccionamos Run File, se ejecutará el browser de la siguiente manera.



Una vez ingresada la marca en el cuadro de texto, damos clic en el botón Mostrar lista, este recargará la página de la siguiente manera.



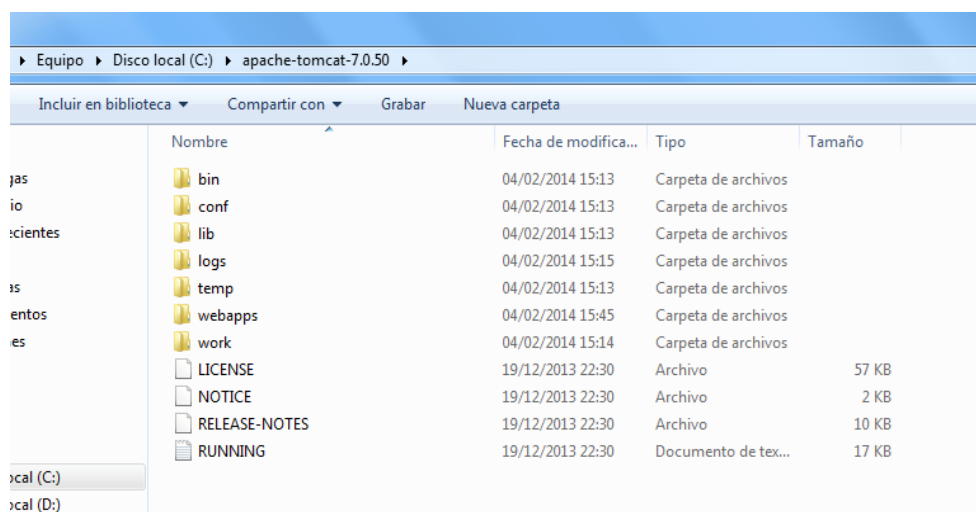
Se recargará la página mostrando la tabla con los datos de la sentencia SELECT realizada en el JDataSet.

Ahora podemos decir que el componente JDataSet abrevia el procedimiento de llamado y acceso de datos, reemplazando los redundantes métodos de conexión a través de código fuente por un asistente gráfico y un código fuente reducido.

8.- Ejecutando la aplicación web desde el servidor Apache Tomcat

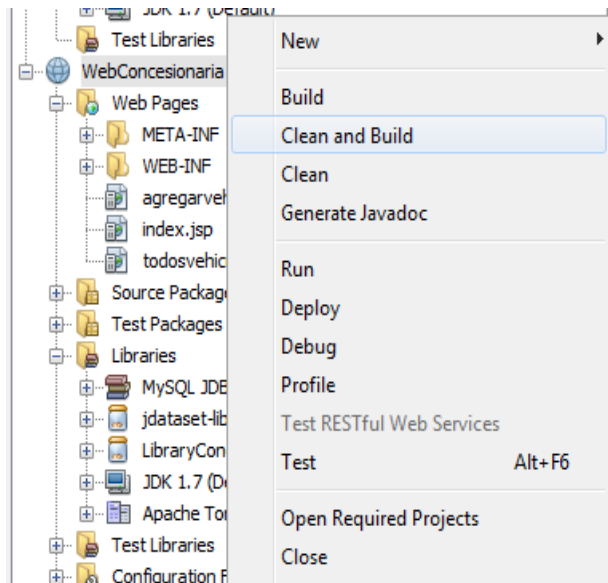
Como último punto nos queda aclarar la ejecución de la aplicación WebConcesionaria desde el servidor Apache Tomcat.

Yo tengo instalado el Apache Tomcat versión 7.0.50, fue instalado manualmente en el siguiente directorio C:\apache-tomcat-7.0.50\ después de este directorio tengo las siguientes carpetas y archivos.

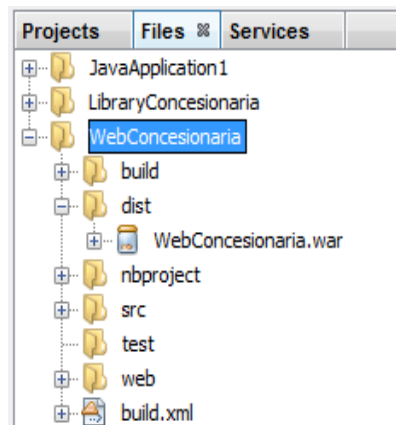


Estos archivos y carpetas permiten el funcionamiento del servidor, entre las cuales destacan la carpeta bin que tiene los archivos para iniciar y detener el servidor, también la carpeta webapps en donde agregaremos el archivo WAR que proporciona Netbeans de la aplicación web WebConcesionaria.

Un archivo WAR se refiere a la encapsulación de todos los archivos que contiene un proyecto web. Esto lo hacemos de la siguiente manera, vamos al nodo principal del proyecto WebConcesionaria, damos clic derecho y seleccionamos Clean and Build, tal y como vemos en la siguiente figura.



Al instante el IDE creará un archivo `WebConcesionaria.war`, este los podemos ver yendo a la pestaña `File`, en el proyecto `WebConcesionaria`, dentro de la carpeta `dist`.



El proyecto ha quedado listo para copiarlo a la carpeta `appwebs` de Apache Tomcat y ejecutarlo, pero va a suceder un error porque en la librería `LibraryConcesionaria`, en la clase `Vehiculo` el directorio del archivo XML no va a ser encontrado dentro de la aplicación, me refiero a la variable `path` como vemos en la siguiente figura.

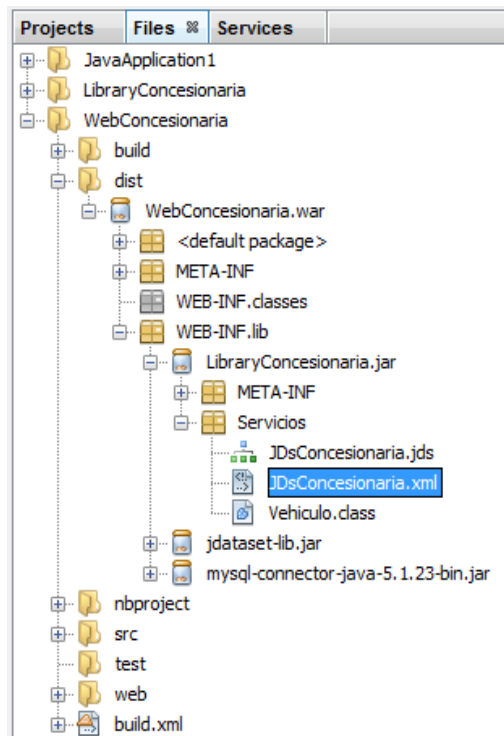
```

public String[][] todoslosVehiculos(String marca){
    String[][] matriz = null;
    try {
        String path = "C:\\Users\\Daniel\\Documents\\NetBeansProjects"
            + "\\LibraryConcesionaria\\src\\Servicios\\JDsConcesionaria.xml"
        JDataset ds1 = new JDataset(path);
        ds1.SetFunctionName("SeleccionarPorMarca()");
        ds1.SetParameterValue("@marca", "'" + marca + "'");
        matriz = ds1.executeQuery();
    } catch (ParserConfigurationException ex) {
        mensaje = ex.getMessage();
    } catch (IOException ex) {
        mensaje = ex.getMessage();
    } catch (SAXException ex) {
        mensaje = ex.getMessage();
    } catch (SQLException ex) {
        mensaje = ex.getMessage();
    } catch (ClassNotFoundException ex) {
        mensaje = ex.getMessage();
    }
    return matriz;
}

```

La variable path tiene que ser modificada designando como valor el directorio del servidor seguido por las carpetas que contienen al archivo JDsConcesionaria.xml dentro del proyecto WebConcesionaria.

Para aclarar esta parte vamos a la pestaña Files, y veamos las carpetas que contiene el proyecto WebConcesionaria, vamos a la carpeta dist, donde está el archivo WebConcesionaria.war, este vendría a ser una carpeta inicial, dentro de este vamos al paquete WEB-INF.lib, este quiere decir que hay dos carpetas WEB-INF y lib, dentro de este se encuentra el archivo LibraryConcesionaria.jar, el cual es otra carpeta, dentro de este está el paquete Servicios, esta es otra carpeta en donde finalmente encontramos el archivo JDsConcesionaria.xml, tal y como vemos en la siguiente figura.



Entonces comenzando desde el directorio del servidor Apache Tomcat, tenemos la primera parte:

C:\apache-tomcat-7.0.50\webapps\

Ya que es en la carpeta webapps donde vamos a copiar el archivo WebConcesionaria.war. Luego sigue la parte del archivo WAR:

WebConcesionaria\WEB-INF\lib\LibraryConcesionaria\Servicios\

Y finalmente agregamos: JDsConcesionaria.xml.

Ahora vamos a modificar el valor de esta variable en los dos métodos de la clase Vehiculo, tal y como vemos en la siguiente figura.

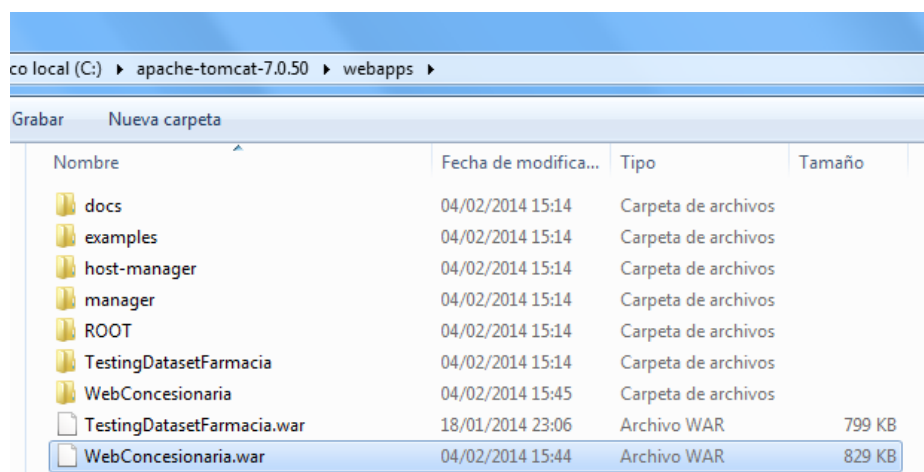
```
String path = "C:\\apache-tomcat-7.0.50\\webapps\\WebConcesionaria\\WEB-INF\\lib"
              + "\\LibraryConcesionaria\\Servicios\\JDsConcesionaria.xml";
```

Finalmente el directorio quedaría de esta manera. Ahora sobre el mismo proyecto LibraryConcesionaria dale clic derecho y selecciona Clean and

Builder. Y en el proyecto WebConcesionaria elimina el archivo LibraryConcesionaria.jds, búscalo de nuevo según el directorio en donde este guardado y agrégalo nuevamente al proyecto.

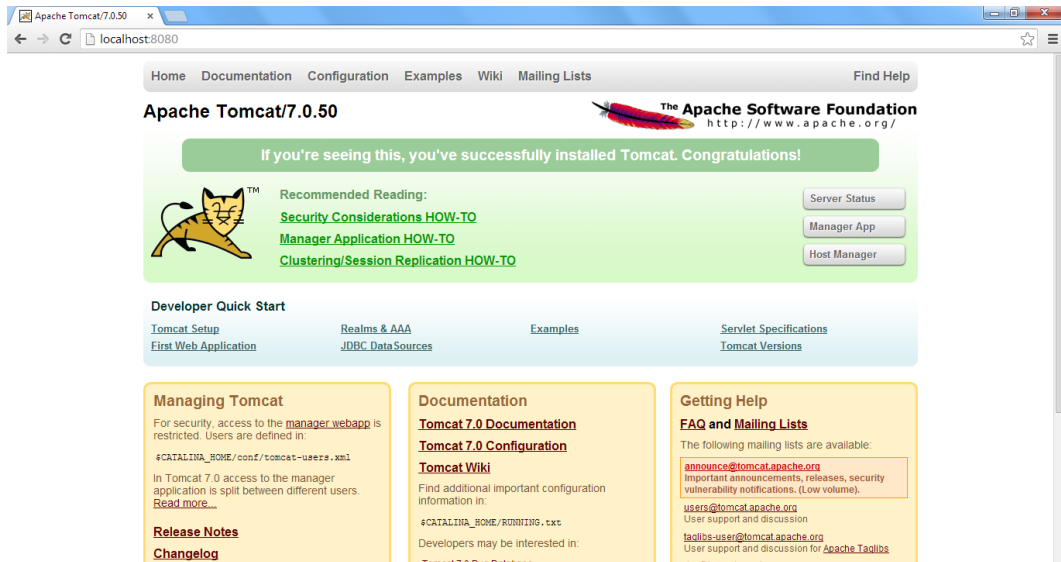
Finalmente vamos al nodo principal del proyecto WebConcesionaria, damos clic derecho sobre este, y seleccionamos Clean and Build para actualizar el archivo WebConcesionaria.war

Ahora buscamos el archivo WebConcesionaria.war y lo agregamos a la carpeta appwebs del servidor Apache Tomcat, como se ve en la siguiente figura.

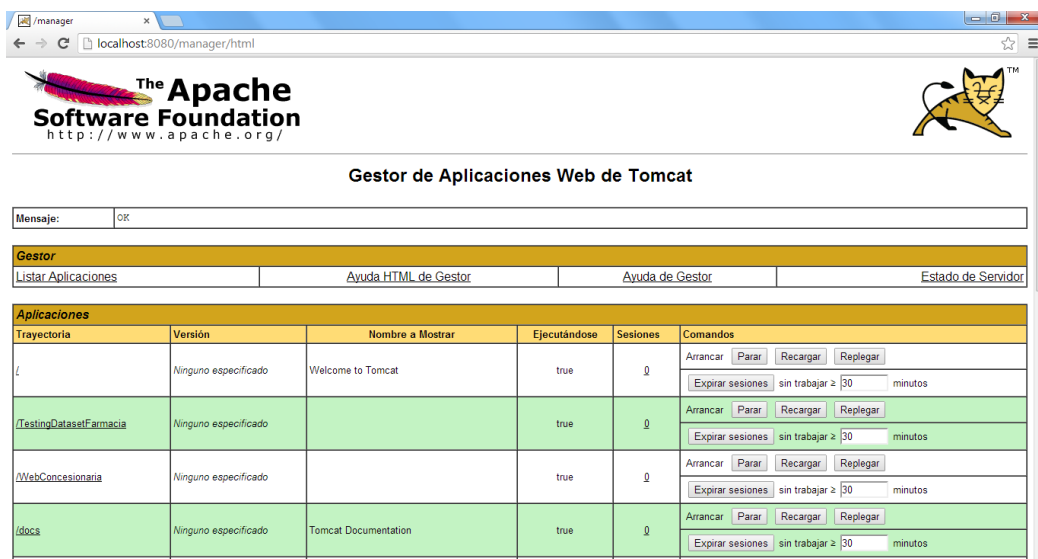


Una vez iniciado el servidor se crea una carpeta que descomprime el archivo WebConcesionaria.war como se ve en la figura anterior. La instalación, administración e inicio del servidor Apache no los voy a detallar para ahorrar tiempo ya que este tutorial ha sido un poco largo. Si no tienes idea del funcionamiento de Apache Tomcat te pido que lo investigues.

Ahora vamos a abrir el browser y escribimos localhost:8080, nos mostrará la siguiente página.



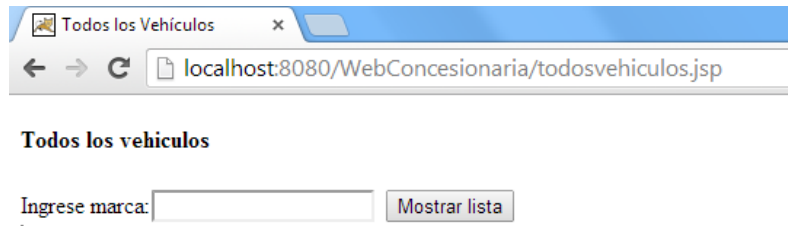
Para ver la aplicación WebConcesionaria damos clic en el botón Manager App, nos pedirá los datos del Login de administrador (esto es según como hayas configurado tu servidor), ingresar usuario y password. Después de eso veremos la siguiente página.



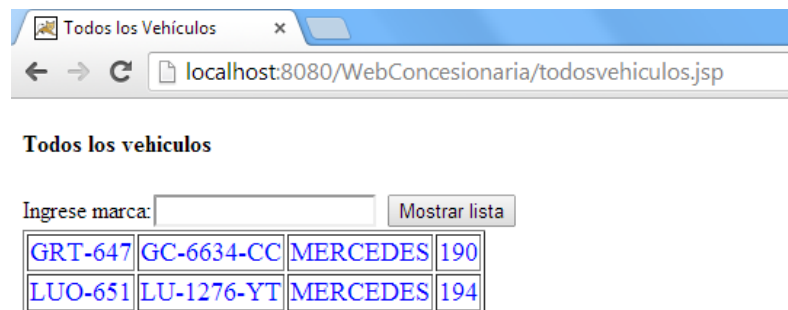
Vemos en el listado la aplicación WebConcesionaria lista para ser ejecutada. Para ir a la página todosvehiculos.jsp, escribimos en el navegador:

localhost:8080/WebConcesionaria/todosvehiculos.jsp

Nos mostrará la siguiente página.



Aquí vemos que la ejecución de la página fue satisfactoria, ahora falta verificar si el procedimiento de conexión a la base de datos a través del componente JDataSet funciona. Ingresamos un valor al cuadro de texto, yo le puse MERCEDES y damos clic en el botón Mostrar lista.



Se recargará la página. Y como muestra la figura anterior la tabla con los valores de la consulta, podemos decir que el resultado ha sido favorable.

Plugins JDataSet versión 1.0.0, elaborado por Daniel Andrés De los Santos Villao, estudiante de la carrera Ing. en Sistemas, Facultad Ciencias de la Ingeniería de la Universidad Técnica Estatal de Quevedo. Espero sus comentarios y recomendaciones acerca de este plugins a mi correo danielcls007@gmail.com

Fecha de publicación: 06-02-2014. Quevedo - Los Ríos – Ecuador

TABLA T-STUDENT

Tabla t-Student



Grados de libertad	0.25	0.1	0.05	0.025	0.01	0.005
1	1.0000	3.0777	6.3137	12.7062	31.8210	63.6559
2	0.8165	1.8856	2.9200	4.3027	6.9645	9.9250
3	0.7649	1.6377	2.3534	3.1824	4.5407	5.8408
4	0.7407	1.5332	2.1318	2.7765	3.7469	4.6041
5	0.7267	1.4759	2.0150	2.5706	3.3649	4.0321
6	0.7176	1.4398	1.9432	2.4469	3.1427	3.7074
7	0.7111	1.4149	1.8946	2.3646	2.9979	3.4995
8	0.7064	1.3968	1.8595	2.3060	2.8965	3.3554
9	0.7027	1.3830	1.8331	2.2622	2.8214	3.2498
10	0.6998	1.3722	1.8125	2.2281	2.7638	3.1693
11	0.6974	1.3634	1.7959	2.2010	2.7181	3.1058
12	0.6955	1.3562	1.7823	2.1788	2.6810	3.0545
13	0.6938	1.3502	1.7709	2.1604	2.6503	3.0123
14	0.6924	1.3450	1.7613	2.1448	2.6245	2.9768
15	0.6912	1.3406	1.7531	2.1315	2.6025	2.9467
16	0.6901	1.3368	1.7459	2.1199	2.5835	2.9208
17	0.6892	1.3334	1.7396	2.1098	2.5669	2.8982
18	0.6884	1.3304	1.7341	2.1009	2.5524	2.8784
19	0.6876	1.3277	1.7291	2.0930	2.5395	2.8609
20	0.6870	1.3253	1.7247	2.0860	2.5280	2.8453
21	0.6864	1.3232	1.7207	2.0796	2.5176	2.8314
22	0.6858	1.3212	1.7171	2.0739	2.5083	2.8188
23	0.6853	1.3195	1.7139	2.0687	2.4999	2.8073
24	0.6848	1.3178	1.7109	2.0639	2.4922	2.7970
25	0.6844	1.3163	1.7081	2.0595	2.4851	2.7874
26	0.6840	1.3150	1.7056	2.0555	2.4786	2.7787
27	0.6837	1.3137	1.7033	2.0518	2.4727	2.7707
28	0.6834	1.3125	1.7011	2.0484	2.4671	2.7633
29	0.6830	1.3114	1.6991	2.0452	2.4620	2.7564
30	0.6828	1.3104	1.6973	2.0423	2.4573	2.7500
31	0.6825	1.3095	1.6955	2.0395	2.4528	2.7440
32	0.6822	1.3086	1.6939	2.0369	2.4487	2.7385
33	0.6820	1.3077	1.6924	2.0345	2.4448	2.7333
34	0.6818	1.3070	1.6909	2.0322	2.4411	2.7284
35	0.6816	1.3062	1.6896	2.0301	2.4377	2.7238
36	0.6814	1.3055	1.6883	2.0281	2.4345	2.7195
37	0.6812	1.3049	1.6871	2.0262	2.4314	2.7154
38	0.6810	1.3042	1.6860	2.0244	2.4286	2.7116
39	0.6808	1.3036	1.6849	2.0227	2.4258	2.7079
40	0.6807	1.3031	1.6839	2.0211	2.4233	2.7045
41	0.6805	1.3025	1.6829	2.0195	2.4208	2.7012
42	0.6804	1.3020	1.6820	2.0181	2.4185	2.6981
43	0.6802	1.3016	1.6811	2.0167	2.4163	2.6951
44	0.6801	1.3011	1.6802	2.0154	2.4141	2.6923
45	0.6800	1.3007	1.6794	2.0141	2.4121	2.6896
46	0.6799	1.3002	1.6787	2.0129	2.4102	2.6870
47	0.6797	1.2998	1.6779	2.0117	2.4083	2.6846
48	0.6796	1.2994	1.6772	2.0106	2.4066	2.6822
49	0.6795	1.2991	1.6766	2.0096	2.4049	2.6800

50	0.6794	1.2987	1.6759	2.0086	2.4033	2.6778
51	0.6793	1.2984	1.6753	2.0076	2.4017	2.6757
52	0.6792	1.2980	1.6747	2.0066	2.4002	2.6737
53	0.6791	1.2977	1.6741	2.0057	2.3988	2.6718
54	0.6791	1.2974	1.6736	2.0049	2.3974	2.6700
55	0.6790	1.2971	1.6730	2.0040	2.3961	2.6682
56	0.6789	1.2969	1.6725	2.0032	2.3948	2.6665
57	0.6788	1.2966	1.6720	2.0025	2.3936	2.6649
58	0.6787	1.2963	1.6716	2.0017	2.3924	2.6633
59	0.6787	1.2961	1.6711	2.0010	2.3912	2.6618
60	0.6786	1.2958	1.6706	2.0003	2.3901	2.6603
61	0.6785	1.2956	1.6702	1.9996	2.3890	2.6589
62	0.6785	1.2954	1.6698	1.9990	2.3880	2.6575
63	0.6784	1.2951	1.6694	1.9983	2.3870	2.6561
64	0.6783	1.2949	1.6690	1.9977	2.3860	2.6549
65	0.6783	1.2947	1.6686	1.9971	2.3851	2.6536
66	0.6782	1.2945	1.6683	1.9966	2.3842	2.6524
67	0.6782	1.2943	1.6679	1.9960	2.3833	2.6512
68	0.6781	1.2941	1.6676	1.9955	2.3824	2.6501
69	0.6781	1.2939	1.6672	1.9949	2.3816	2.6490
70	0.6780	1.2938	1.6669	1.9944	2.3808	2.6479
71	0.6780	1.2936	1.6666	1.9939	2.3800	2.6469
72	0.6779	1.2934	1.6663	1.9935	2.3793	2.6458
73	0.6779	1.2933	1.6660	1.9930	2.3785	2.6449
74	0.6778	1.2931	1.6657	1.9925	2.3778	2.6439
75	0.6778	1.2929	1.6654	1.9921	2.3771	2.6430
76	0.6777	1.2928	1.6652	1.9917	2.3764	2.6421
77	0.6777	1.2926	1.6649	1.9913	2.3758	2.6412
78	0.6776	1.2925	1.6646	1.9908	2.3751	2.6403
79	0.6776	1.2924	1.6644	1.9905	2.3745	2.6395
80	0.6776	1.2922	1.6641	1.9901	2.3739	2.6387
81	0.6775	1.2921	1.6639	1.9897	2.3733	2.6379
82	0.6775	1.2920	1.6636	1.9893	2.3727	2.6371
83	0.6775	1.2918	1.6634	1.9890	2.3721	2.6364
84	0.6774	1.2917	1.6632	1.9886	2.3716	2.6356
85	0.6774	1.2916	1.6630	1.9883	2.3710	2.6349
86	0.6774	1.2915	1.6628	1.9879	2.3705	2.6342
87	0.6773	1.2914	1.6626	1.9876	2.3700	2.6335
88	0.6773	1.2912	1.6624	1.9873	2.3695	2.6329
89	0.6773	1.2911	1.6622	1.9870	2.3690	2.6322
90	0.6772	1.2910	1.6620	1.9867	2.3685	2.6316
91	0.6772	1.2909	1.6618	1.9864	2.3680	2.6309
92	0.6772	1.2908	1.6616	1.9861	2.3676	2.6303
93	0.6771	1.2907	1.6614	1.9858	2.3671	2.6297
94	0.6771	1.2906	1.6612	1.9855	2.3667	2.6291
95	0.6771	1.2905	1.6611	1.9852	2.3662	2.6286
96	0.6771	1.2904	1.6609	1.9850	2.3658	2.6280
97	0.6770	1.2903	1.6607	1.9847	2.3654	2.6275
98	0.6770	1.2903	1.6606	1.9845	2.3650	2.6269
99	0.6770	1.2902	1.6604	1.9842	2.3646	2.6264
100	0.6770	1.2901	1.6602	1.9840	2.3642	2.6259
∞	0.6745	1.2816	1.6449	1.9600	2.3263	2.5758

FICHAS DE REGISTRO DEL CUASI-EXPERIMENTO

FICHA DE PROGRAMADOR 1

Actividades del programador		O ₁ = Implementando la escritura de código fuente línea por línea	O ₂ = Implementando el componente JDataSet
		Tiempo en minutos	Tiempo en minutos
1	Desarrollar un método para la conexión con la base de datos	15	0.6
2	Desarrollar un método que realice una consulta de datos entre varias tablas relacionadas	21	2
3	Desarrollar un método para la inserción de datos	15	0.5
4	Desarrollar un método para la actualización de datos	25	0.6
5	Desarrollar un método para la eliminación de datos	15	1
6	Visualizar los registros afectados después de ejecutar una operación hacia la base de datos	1.5	0.1

FICHA DE PROGRAMADOR 2

Actividades del programador		O ₁ = Implementando la escritura de código fuente línea por línea	O ₂ = Implementando el componente JDataSet
		Tiempo en minutos	Tiempo en minutos
1	Desarrollar un método para la conexión con la base de datos	10	0.1
2	Desarrollar un método que realice una consulta de datos entre varias tablas relacionadas	29	3
3	Desarrollar un método para la inserción de datos	15	0.2
4	Desarrollar un método para la actualización de datos	22	0.5
5	Desarrollar un método para la eliminación de datos	13	0.5
6	Visualizar los registros afectados después de ejecutar una operación hacia la base de datos	3	0.2

FICHA DE PROGRAMADOR 3

Actividades del programador		O ₁ = Implementando la escritura de código fuente línea por línea	O ₂ = Implementando el componente JDataSet
		Tiempo en minutos	Tiempo en minutos
1	Desarrollar un método para la conexión con la base de datos	20	0.3
2	Desarrollar un método que realice una consulta de datos entre varias tablas relacionadas	23	2.5
3	Desarrollar un método para la inserción de datos	20	0.5
4	Desarrollar un método para la actualización de datos	20	0.6
5	Desarrollar un método para la eliminación de datos	13	0.5
6	Visualizar los registros afectados después de ejecutar una operación hacia la base de datos	2	0.1

FICHA DE PROGRAMADOR 4

Actividades del programador		O ₁ = Implementando la escritura de código fuente línea por línea	O ₂ = Implementando el componente JDataSet
		Tiempo en minutos	Tiempo en minutos
1	Desarrollar un método para la conexión con la base de datos	12	0.5
2	Desarrollar un método que realice una consulta de datos entre varias tablas relacionadas	27	4
3	Desarrollar un método para la inserción de datos	16	0.5
4	Desarrollar un método para la actualización de datos	18	0.5
5	Desarrollar un método para la eliminación de datos	14	0.3
6	Visualizar los registros afectados después de ejecutar una operación hacia la base de datos	2	0.1

FICHA DE PROGRAMADOR 5

Actividades del programador		O ₁ = Implementando la escritura de código fuente línea por línea	O ₂ = Implementando el componente JDataSet
		Tiempo en minutos	Tiempo en minutos
1	Desarrollar un método para la conexión con la base de datos	17	0.1
2	Desarrollar un método que realice una consulta de datos entre varias tablas relacionadas	25	5
3	Desarrollar un método para la inserción de datos	16	0.8
4	Desarrollar un método para la actualización de datos	20	0.5
5	Desarrollar un método para la eliminación de datos	15	0.2
6	Visualizar los registros afectados después de ejecutar una operación hacia la base de datos	1	0.5

FICHA DE PROGRAMADOR 6

Actividades del programador		O ₁ = Implementando la escritura de código fuente línea por línea	O ₂ = Implementando el componente JDataSet
		Tiempo en minutos	Tiempo en minutos
1	Desarrollar un método para la conexión con la base de datos	15	0.3
2	Desarrollar un método que realice una consulta de datos entre varias tablas relacionadas	24	3.5
3	Desarrollar un método para la inserción de datos	12	1
4	Desarrollar un método para la actualización de datos	20	0.5
5	Desarrollar un método para la eliminación de datos	12	0.1
6	Visualizar los registros afectados después de ejecutar una operación hacia la base de datos	2.5	0.2

FICHA DE PROGRAMADOR 7

Actividades del programador		O ₁ = Implementando la escritura de código fuente línea por línea	O ₂ = Implementando el componente JDataSet
		Tiempo en minutos	Tiempo en minutos
1	Desarrollar un método para la conexión con la base de datos	11	0.2
2	Desarrollar un método que realice una consulta de datos entre varias tablas relacionadas	26	1
3	Desarrollar un método para la inserción de datos	15	0.3
4	Desarrollar un método para la actualización de datos	15	0.5
5	Desarrollar un método para la eliminación de datos	13	0.4
6	Visualizar los registros afectados después de ejecutar una operación hacia la base de datos	2	0.3

FICHA DE PROGRAMADOR 8

Actividades del programador		O ₁ = Implementando la escritura de código fuente línea por línea	O ₂ = Implementando el componente JDataSet
		Tiempo en minutos	Tiempo en minutos
1	Desarrollar un método para la conexión con la base de datos	19	0.4
2	Desarrollar un método que realice una consulta de datos entre varias tablas relacionadas	23	3.5
3	Desarrollar un método para la inserción de datos	16	0.2
4	Desarrollar un método para la actualización de datos	21	0.4
5	Desarrollar un método para la eliminación de datos	11	0.5
6	Visualizar los registros afectados después de ejecutar una operación hacia la base de datos	2	0.2

FICHA DE PROGRAMADOR 9

Actividades del programador		O ₁ = Implementando la escritura de código fuente línea por línea	O ₂ = Implementando el componente JDataSet
		Tiempo en minutos	Tiempo en minutos
1	Desarrollar un método para la conexión con la base de datos	16	0.4
2	Desarrollar un método que realice una consulta de datos entre varias tablas relacionadas	27	5
3	Desarrollar un método para la inserción de datos	15	0.5
4	Desarrollar un método para la actualización de datos	20	0.4
5	Desarrollar un método para la eliminación de datos	13	0.5
6	Visualizar los registros afectados después de ejecutar una operación hacia la base de datos	2	0.1

FICHA DE PROGRAMADOR 10

Actividades del programador		O ₁ = Implementando la escritura de código fuente línea por línea	O ₂ = Implementando el componente JDataSet
		Tiempo en minutos	Tiempo en minutos
1	Desarrollar un método para la conexión con la base de datos	14	0.2
2	Desarrollar un método que realice una consulta de datos entre varias tablas relacionadas	30	2.5
3	Desarrollar un método para la inserción de datos	16	0.4
4	Desarrollar un método para la actualización de datos	20	1
5	Desarrollar un método para la eliminación de datos	12	0.6
6	Visualizar los registros afectados después de ejecutar una operación hacia la base de datos	3	0.2

FICHA DE PROGRAMADOR 11

Actividades del programador		O ₁ = Implementando la escritura de código fuente línea por línea	O ₂ = Implementando el componente JDataSet
		Tiempo en minutos	Tiempo en minutos
1	Desarrollar un método para la conexión con la base de datos	15	0.5
2	Desarrollar un método que realice una consulta de datos entre varias tablas relacionadas	20	3
3	Desarrollar un método para la inserción de datos	17	0.3
4	Desarrollar un método para la actualización de datos	19	0.5
5	Desarrollar un método para la eliminación de datos	13	0.5
6	Visualizar los registros afectados después de ejecutar una operación hacia la base de datos	2	0.2

FICHA DE PROGRAMADOR 12

Actividades del programador		O ₁ = Implementando la escritura de código fuente línea por línea	O ₂ = Implementando el componente JDataSet
		Tiempo en minutos	Tiempo en minutos
1	Desarrollar un método para la conexión con la base de datos	20	0.1
2	Desarrollar un método que realice una consulta de datos entre varias tablas relacionadas	25	3
3	Desarrollar un método para la inserción de datos	16	0.5
4	Desarrollar un método para la actualización de datos	25	0.5
5	Desarrollar un método para la eliminación de datos	13	0.2
6	Visualizar los registros afectados después de ejecutar una operación hacia la base de datos	2	0.2

FICHA DE PROGRAMADOR 13

Actividades del programador		O ₁ = Implementando la escritura de código fuente línea por línea	O ₂ = Implementando el componente JDataSet
		Tiempo en minutos	Tiempo en minutos
1	Desarrollar un método para la conexión con la base de datos	10	0.3
2	Desarrollar un método que realice una consulta de datos entre varias tablas relacionadas	21	5.5
3	Desarrollar un método para la inserción de datos	17	0.5
4	Desarrollar un método para la actualización de datos	20	0.2
5	Desarrollar un método para la eliminación de datos	15	0.4
6	Visualizar los registros afectados después de ejecutar una operación hacia la base de datos	1	0.2

FICHA DE PROGRAMADOR 14

Actividades del programador		O ₁ = Implementando la escritura de código fuente línea por línea	O ₂ = Implementando el componente JDataSet
		Tiempo en minutos	Tiempo en minutos
1	Desarrollar un método para la conexión con la base de datos	15	0.3
2	Desarrollar un método que realice una consulta de datos entre varias tablas relacionadas	22	3
3	Desarrollar un método para la inserción de datos	18	0.7
4	Desarrollar un método para la actualización de datos	18	0.2
5	Desarrollar un método para la eliminación de datos	15	0.5
6	Visualizar los registros afectados después de ejecutar una operación hacia la base de datos	2.5	0.2

FICHA DE PROGRAMADOR 15

Actividades del programador		O ₁ = Implementando la escritura de código fuente línea por línea	O ₂ = Implementando el componente JDataSet
		Tiempo en minutos	Tiempo en minutos
1	Desarrollar un método para la conexión con la base de datos	16	0.3
2	Desarrollar un método que realice una consulta de datos entre varias tablas relacionadas	29	3.5
3	Desarrollar un método para la inserción de datos	17	0.3
4	Desarrollar un método para la actualización de datos	15	0.5
5	Desarrollar un método para la eliminación de datos	10	0.5
6	Visualizar los registros afectados después de ejecutar una operación hacia la base de datos	2	0.2

FICHA DE PROGRAMADOR 16

Actividades del programador		O ₁ = Implementando la escritura de código fuente línea por línea	O ₂ = Implementando el componente JDataSet
		Tiempo en minutos	Tiempo en minutos
1	Desarrollar un método para la conexión con la base de datos	14	0.4
2	Desarrollar un método que realice una consulta de datos entre varias tablas relacionadas	28	3
3	Desarrollar un método para la inserción de datos	16	0.2
4	Desarrollar un método para la actualización de datos	20	0.1
5	Desarrollar un método para la eliminación de datos	12	0.5
6	Visualizar los registros afectados después de ejecutar una operación hacia la base de datos	2	0.3

FICHA DE PROGRAMADOR 17

Actividades del programador		O ₁ = Implementando la escritura de código fuente línea por línea	O ₂ = Implementando el componente JDataSet
		Tiempo en minutos	Tiempo en minutos
1	Desarrollar un método para la conexión con la base de datos	13	0.4
2	Desarrollar un método que realice una consulta de datos entre varias tablas relacionadas	25	3
3	Desarrollar un método para la inserción de datos	15	0.8
4	Desarrollar un método para la actualización de datos	20	0.5
5	Desarrollar un método para la eliminación de datos	12	0.6
6	Visualizar los registros afectados después de ejecutar una operación hacia la base de datos	1.5	0.2

FICHA DE PROGRAMADOR 18

Actividades del programador		O ₁ = Implementando la escritura de código fuente línea por línea	O ₂ = Implementando el componente JDataSet
		Tiempo en minutos	Tiempo en minutos
1	Desarrollar un método para la conexión con la base de datos	17	0.3
2	Desarrollar un método que realice una consulta de datos entre varias tablas relacionadas	21	0.5
3	Desarrollar un método para la inserción de datos	16	0.7
4	Desarrollar un método para la actualización de datos	22	0.5
5	Desarrollar un método para la eliminación de datos	12	0.5
6	Visualizar los registros afectados después de ejecutar una operación hacia la base de datos	3	0.2

FICHA DE PROGRAMADOR 19

Actividades del programador		O ₁ = Implementando la escritura de código fuente línea por línea	O ₂ = Implementando el componente JDataSet
		Tiempo en minutos	Tiempo en minutos
1	Desarrollar un método para la conexión con la base de datos	21	0.2
2	Desarrollar un método que realice una consulta de datos entre varias tablas relacionadas	29	1
3	Desarrollar un método para la inserción de datos	17	0.5
4	Desarrollar un método para la actualización de datos	21	0.6
5	Desarrollar un método para la eliminación de datos	13	0.9
6	Visualizar los registros afectados después de ejecutar una operación hacia la base de datos	2	0.1

FICHA DE PROGRAMADOR 20

Actividades del programador		O ₁ = Implementando la escritura de código fuente línea por línea	O ₂ = Implementando el componente JDataSet
		Tiempo en minutos	Tiempo en minutos
1	Desarrollar un método para la conexión con la base de datos	9	0.1
2	Desarrollar un método que realice una consulta de datos entre varias tablas relacionadas	21	2
3	Desarrollar un método para la inserción de datos	15	0.3
4	Desarrollar un método para la actualización de datos	20	0.5
5	Desarrollar un método para la eliminación de datos	14	0.4
6	Visualizar los registros afectados después de ejecutar una operación hacia la base de datos	2	0.5

FICHA DE PROGRAMADOR 21

Actividades del programador		O ₁ = Implementando la escritura de código fuente línea por línea	O ₂ = Implementando el componente JDataSet
		Tiempo en minutos	Tiempo en minutos
1	Desarrollar un método para la conexión con la base de datos	15	0.2
2	Desarrollar un método que realice una consulta de datos entre varias tablas relacionadas	30	3
3	Desarrollar un método para la inserción de datos	16	0.2
4	Desarrollar un método para la actualización de datos	20	0.8
5	Desarrollar un método para la eliminación de datos	11	0.5
6	Visualizar los registros afectados después de ejecutar una operación hacia la base de datos	2	0.3

FICHA DE PROGRAMADOR 22

Actividades del programador		O ₁ = Implementando la escritura de código fuente línea por línea	O ₂ = Implementando el componente JDataSet
		Tiempo en minutos	Tiempo en minutos
1	Desarrollar un método para la conexión con la base de datos	17	0.5
2	Desarrollar un método que realice una consulta de datos entre varias tablas relacionadas	29	5
3	Desarrollar un método para la inserción de datos	16	0.6
4	Desarrollar un método para la actualización de datos	19	0.5
5	Desarrollar un método para la eliminación de datos	14	0.5
6	Visualizar los registros afectados después de ejecutar una operación hacia la base de datos	1.5	0.2

FICHA DE PROGRAMADOR 23

Actividades del programador		O ₁ = Implementando la escritura de código fuente línea por línea	O ₂ = Implementando el componente JDataSet
		Tiempo en minutos	Tiempo en minutos
1	Desarrollar un método para la conexión con la base de datos	13	0.5
2	Desarrollar un método que realice una consulta de datos entre varias tablas relacionadas	20	3
3	Desarrollar un método para la inserción de datos	16	0.8
4	Desarrollar un método para la actualización de datos	24	0.5
5	Desarrollar un método para la eliminación de datos	15	0.6
6	Visualizar los registros afectados después de ejecutar una operación hacia la base de datos	1	0.2

FICHA DE PROGRAMADOR 24

Actividades del programador		O ₁ = Implementando la escritura de código fuente línea por línea	O ₂ = Implementando el componente JDataSet
		Tiempo en minutos	Tiempo en minutos
1	Desarrollar un método para la conexión con la base de datos	19	0.3
2	Desarrollar un método que realice una consulta de datos entre varias tablas relacionadas	24	4
3	Desarrollar un método para la inserción de datos	15	0.5
4	Desarrollar un método para la actualización de datos	20	0.5
5	Desarrollar un método para la eliminación de datos	13	0.5
6	Visualizar los registros afectados después de ejecutar una operación hacia la base de datos	2	0.2

FICHA DE PROGRAMADOR 25

Actividades del programador		O ₁ = Implementando la escritura de código fuente línea por línea	O ₂ = Implementando el componente JDataSet
		Tiempo en minutos	Tiempo en minutos
1	Desarrollar un método para la conexión con la base de datos	11	0.2
2	Desarrollar un método que realice una consulta de datos entre varias tablas relacionadas	25	3
3	Desarrollar un método para la inserción de datos	15	0.5
4	Desarrollar un método para la actualización de datos	16	0.2
5	Desarrollar un método para la eliminación de datos	13	0.6
6	Visualizar los registros afectados después de ejecutar una operación hacia la base de datos	2.5	0.2

FICHA DE PROGRAMADOR 26

Actividades del programador		O ₁ = Implementando la escritura de código fuente línea por línea	O ₂ = Implementando el componente JDataSet
		Tiempo en minutos	Tiempo en minutos
1	Desarrollar un método para la conexión con la base de datos	12	0.1
2	Desarrollar un método que realice una consulta de datos entre varias tablas relacionadas	25	2
3	Desarrollar un método para la inserción de datos	16	0.5
4	Desarrollar un método para la actualización de datos	18	0.5
5	Desarrollar un método para la eliminación de datos	12	0.5
6	Visualizar los registros afectados después de ejecutar una operación hacia la base de datos	2	0.2

FICHA DE PROGRAMADOR 27

Actividades del programador		O ₁ = Implementando la escritura de código fuente línea por línea	O ₂ = Implementando el componente JDataSet
		Tiempo en minutos	Tiempo en minutos
1	Desarrollar un método para la conexión con la base de datos	18	0.4
2	Desarrollar un método que realice una consulta de datos entre varias tablas relacionadas	29	3
3	Desarrollar un método para la inserción de datos	17	0.5
4	Desarrollar un método para la actualización de datos	20	0.5
5	Desarrollar un método para la eliminación de datos	13	0.5
6	Visualizar los registros afectados después de ejecutar una operación hacia la base de datos	2	0.2

FICHA DE PROGRAMADOR 28

Actividades del programador		O ₁ = Implementando la escritura de código fuente línea por línea	O ₂ = Implementando el componente JDataSet
		Tiempo en minutos	Tiempo en minutos
1	Desarrollar un método para la conexión con la base de datos	14	0.3
2	Desarrollar un método que realice una consulta de datos entre varias tablas relacionadas	19	4
3	Desarrollar un método para la inserción de datos	16	0.8
4	Desarrollar un método para la actualización de datos	20	0.4
5	Desarrollar un método para la eliminación de datos	14	0.4
6	Visualizar los registros afectados después de ejecutar una operación hacia la base de datos	2	0.1

FICHA DE PROGRAMADOR 29

Actividades del programador		O ₁ = Implementando la escritura de código fuente línea por línea	O ₂ = Implementando el componente JDataSet
		Tiempo en minutos	Tiempo en minutos
1	Desarrollar un método para la conexión con la base de datos	16	0.3
2	Desarrollar un método que realice una consulta de datos entre varias tablas relacionadas	24	3
3	Desarrollar un método para la inserción de datos	15	0.9
4	Desarrollar un método para la actualización de datos	18	0.5
5	Desarrollar un método para la eliminación de datos	15	0.8
6	Visualizar los registros afectados después de ejecutar una operación hacia la base de datos	1.5	0.3

FICHA DE PROGRAMADOR 30

Actividades del programador		O ₁ = Implementando la escritura de código fuente línea por línea	O ₂ = Implementando el componente JDataSet
		Tiempo en minutos	Tiempo en minutos
1	Desarrollar un método para la conexión con la base de datos	15	0.1
2	Desarrollar un método que realice una consulta de datos entre varias tablas relacionadas	25	3
3	Desarrollar un método para la inserción de datos	16	0.5
4	Desarrollar un método para la actualización de datos	24	0.5
5	Desarrollar un método para la eliminación de datos	16	0.2
6	Visualizar los registros afectados después de ejecutar una operación hacia la base de datos	2	0.2

FICHA DE PROGRAMADOR 31

Actividades del programador		O ₁ = Implementando la escritura de código fuente línea por línea	O ₂ = Implementando el componente JDataSet
		Tiempo en minutos	Tiempo en minutos
1	Desarrollar un método para la conexión con la base de datos	17	0.5
2	Desarrollar un método que realice una consulta de datos entre varias tablas relacionadas	31	2.5
3	Desarrollar un método para la inserción de datos	16	0.1
4	Desarrollar un método para la actualización de datos	20	0.5
5	Desarrollar un método para la eliminación de datos	13	0.5
6	Visualizar los registros afectados después de ejecutar una operación hacia la base de datos	2	0.4

FICHA DE PROGRAMADOR 32

Actividades del programador		O ₁ = Implementando la escritura de código fuente línea por línea	O ₂ = Implementando el componente JDataSet
		Tiempo en minutos	Tiempo en minutos
1	Desarrollar un método para la conexión con la base de datos	13	0.1
2	Desarrollar un método que realice una consulta de datos entre varias tablas relacionadas	26	3
3	Desarrollar un método para la inserción de datos	16	0.2
4	Desarrollar un método para la actualización de datos	17	0.4
5	Desarrollar un método para la eliminación de datos	12	0.5
6	Visualizar los registros afectados después de ejecutar una operación hacia la base de datos	3	0.2

ENCUESTAS



UNIVERSIDAD TÉCNICA ESTATAL DE QUEVEDO

Encuesta para evaluar la satisfacción de los programadores de software, en cuanto al procedimiento de acceso a datos cuando desarrollan aplicaciones.

Dirigida a: Los estudiantes de la Carrera de Sistemas de la Facultad de Ciencias de la Ingeniería de la UTQEQ

Responda: Según las siguientes declaraciones seleccione una opción, asegurando total honestidad en base a la experiencia obtenida a lo largo de sus estudios universitarios.

1. El programador obtiene completa confianza y seguridad al emplear solamente código fuente para realizar las operaciones transaccionales hacia la base de datos.
Totalmente de acuerdo
De acuerdo
Ni en acuerdo, ni en desacuerdo
En desacuerdo
Totalmente en desacuerdo

2. Clasificar los métodos o funciones para la selección, inserción, actualización y eliminación de datos, permiten que el desarrollo de la aplicación sea mantenido, repetitivo y a la vez complejo.
Totalmente de acuerdo
De acuerdo
Ni en acuerdo, ni en desacuerdo
En desacuerdo
Totalmente en desacuerdo

3. Los componentes gráficos que ayudan al programador a desarrollar aplicaciones, permiten que la construcción del software sea más sencilla y rápida.
Totalmente de acuerdo
De acuerdo
Ni en acuerdo, ni en desacuerdo
En desacuerdo
Totalmente en desacuerdo

4. El desarrollo de los métodos de selección, inserción, actualización y eliminación de datos sería más fácil y rápido si se realizara con la ayuda de componentes de asistencia gráfica.
Totalmente de acuerdo
De acuerdo
Ni en acuerdo, ni en desacuerdo
En desacuerdo
Totalmente en desacuerdo

5. El procedimiento para realizar las transacciones con la base de datos sería más fácil si existiera un módulo desde el SCE (sistema de desarrollo integrado) el diagrama del modelo relacional de la base de datos.

Totalmente de acuerdo
De acuerdo
Ni en acuerdo, ni en desacuerdo
En desacuerdo
Totalmente en desacuerdo

6. Emplear componentes gráficos especializados en las operaciones transaccionales con base de datos influyen total comodidad y facilidad al desarrollar aplicaciones.

Totalmente de acuerdo
De acuerdo
Ni en acuerdo, ni en desacuerdo
En desacuerdo
Totalmente en desacuerdo

7. La implementación de código fuente y componentes de asistencia gráfica debe ser equilibrada al momento de desarrollar aplicaciones.

Totalmente de acuerdo
De acuerdo
Ni en acuerdo, ni en desacuerdo
En desacuerdo
Totalmente en desacuerdo

24 de Julio



UNIVERSIDAD TÉCNICA ESTATAL DE QUEVEDO

Encuesta para evaluar la satisfacción de los programadores de software, en cuanto al procedimiento de acceso a datos cuando desarrollan aplicaciones.

Dirigida a: Los estudiantes de la Carrera de Sistemas de la Facultad de Ciencias de la Ingeniería de la UTQEQ

Responda: Según las siguientes declaraciones seleccione una opción, asegurando total honestidad en base a la experiencia obtenida a lo largo de sus estudios universitarios.

1. El programador obtiene completa confianza y seguridad al emplear solamente código fuente para realizar las operaciones transaccionales hacia la base de datos.
Totalmente de acuerdo
De acuerdo
Ni en acuerdo, ni en desacuerdo
En desacuerdo
Totalmente en desacuerdo

2. Clasificar los métodos o funciones para la selección, inserción, actualización y eliminación de datos, permiten que el desarrollo de la aplicación sea mantenido, repetitivo y a la vez complejo.
Totalmente de acuerdo
De acuerdo
Ni en acuerdo, ni en desacuerdo
En desacuerdo
Totalmente en desacuerdo

3. Los componentes gráficos que ayudan al programador a desarrollar aplicaciones, permiten que la construcción del software sea más sencilla y rápida.
Totalmente de acuerdo
De acuerdo
Ni en acuerdo, ni en desacuerdo
En desacuerdo
Totalmente en desacuerdo

4. El desarrollo de los métodos de selección, inserción, actualización y eliminación de datos sería más fácil y rápido si se realizara con la ayuda de componentes de asistencia gráfica.
Totalmente de acuerdo
De acuerdo
Ni en acuerdo, ni en desacuerdo
En desacuerdo
Totalmente en desacuerdo

5. El procedimiento para realizar las transacciones con la base de datos sería más fácil si existiera un módulo desde el SCE (sistema de desarrollo integrado) el diagrama del modelo relacional de la base de datos.

Totalmente de acuerdo
De acuerdo
Ni en acuerdo, ni en desacuerdo
En desacuerdo
Totalmente en desacuerdo

6. Emplear componentes gráficos especializados en las operaciones transaccionales con base de datos influyen total comodidad y facilidad al desarrollar aplicaciones.

Totalmente de acuerdo
De acuerdo
Ni en acuerdo, ni en desacuerdo
En desacuerdo
Totalmente en desacuerdo

7. La implementación de código fuente y componentes de asistencia gráfica debe ser equilibrada al momento de desarrollar aplicaciones.

Totalmente de acuerdo
De acuerdo
Ni en acuerdo, ni en desacuerdo
En desacuerdo
Totalmente en desacuerdo

24 de Julio



UNIVERSIDAD TÉCNICA ESTATAL DE QUEVEDO

Encuesta para evaluar la satisfacción de los programadores de software, en cuanto al procedimiento de acceso a datos cuando desarrollan aplicaciones.

Dirigida a: Los estudiantes de la Carrera de Sistemas de la Facultad de Ciencias de la Ingeniería de la UTQEQ

Responda: Según las siguientes declaraciones seleccione una opción, asegurando total honestidad en base a la experiencia obtenida a lo largo de sus estudios universitarios.

1. El programador obtiene completa confianza y seguridad al emplear solamente código fuente para realizar las operaciones transaccionales hacia la base de datos.
Totalmente de acuerdo
De acuerdo
Ni en acuerdo, ni en desacuerdo
En desacuerdo
Totalmente en desacuerdo

2. Clasificar los métodos o funciones para la selección, inserción, actualización y eliminación de datos, permiten que el desarrollo de la aplicación sea mantenido, repetitivo y a la vez complejo.
Totalmente de acuerdo
De acuerdo
Ni en acuerdo, ni en desacuerdo
En desacuerdo
Totalmente en desacuerdo

3. Los componentes gráficos que ayudan al programador a desarrollar aplicaciones, permiten que la construcción del software sea más sencilla y rápida.
Totalmente de acuerdo
De acuerdo
Ni en acuerdo, ni en desacuerdo
En desacuerdo
Totalmente en desacuerdo

4. El desarrollo de los métodos de selección, inserción, actualización y eliminación de datos sería más fácil y rápido si se realizara con la ayuda de componentes de asistencia gráfica.
Totalmente de acuerdo
De acuerdo
Ni en acuerdo, ni en desacuerdo
En desacuerdo
Totalmente en desacuerdo

5. El procedimiento para realizar las operaciones con la base de datos sería más fácil si existiera un módulo desde el SCE (sistema de desarrollo integrado) el diagrama del modelo relacional de la base de datos.

Totalmente de acuerdo
De acuerdo
Ni en acuerdo, ni en desacuerdo
En desacuerdo
Totalmente en desacuerdo

6. Emplear componentes gráficos especializados en las operaciones transaccionales con base de datos influyen total comodidad y facilidad al desarrollar aplicaciones.

Totalmente de acuerdo
De acuerdo
Ni en acuerdo, ni en desacuerdo
En desacuerdo
Totalmente en desacuerdo

7. La implementación de código fuente y componentes de asistencia gráfica debe ser equilibrada al momento de desarrollar aplicaciones.

Totalmente de acuerdo
De acuerdo
Ni en acuerdo, ni en desacuerdo
En desacuerdo
Totalmente en desacuerdo

24 de Julio



UNIVERSIDAD TÉCNICA ESTATAL DE QUEVEDO

Encuesta para evaluar la satisfacción de los programadores de software, en cuanto al procedimiento de acceso a datos cuando desarrollan aplicaciones.

Dirigida a: Los estudiantes de la Carrera de Sistemas de la Facultad de Ciencias de la Ingeniería de la UTEQ.

Responda: Según las siguientes observaciones seleccione una opción, marcando total honestidad en base a la experiencia obtenida a lo largo de sus estudios universitarios.

- 1. El programador obtiene completa confianza y seguridad al emplear solamente código fuente para realizar las operaciones transaccionales hacia la base de datos.
2. Codificar los métodos o funciones para la selección, inserción, actualización y eliminación de datos, provoca que el desarrollo de la aplicación sea monótono, repetitivo y a la vez complejo.
3. Los componentes gráficos que ayudan al programador a desarrollar aplicaciones, permiten que la construcción del software sea más sencilla y rápida.
4. El desarrollo de los métodos de selección, inserción, actualización y eliminación de datos sería más fácil y rápido si se realizara con la ayuda de componentes de asistencia gráfica.

5. El procedimiento para realizar las operaciones con la base de datos sería más fácil si pudiéramos visualizar desde el IDE (entorno de desarrollo integrado) el diagrama del modelo relacional de la base de datos.

- 6. Emplear componentes gráficos especializados en las operaciones transaccionales con base de datos, reduce total complejidad y facilita al desarrollar aplicaciones.
7. La implementación de código fuente y componentes de asistencia gráfica debe ser equilibrada al momento de desarrollar aplicaciones.

Dr. Luis Salazar



UNIVERSIDAD TÉCNICA ESTATAL DE QUEVEDO

Encuesta para evaluar la satisfacción de los programadores de software, en cuanto al procedimiento de acceso a datos cuando desarrollan aplicaciones.

Dirigida a: Los estudiantes de la Carrera de Sistemas de la Facultad de Ciencias de la Ingeniería de la UTEQ.

Responda: Según las siguientes observaciones seleccione una opción, marcando total honestidad en base a la experiencia obtenida a lo largo de sus estudios universitarios.

- 1. El programador obtiene completa confianza y seguridad al emplear solamente código fuente para realizar las operaciones transaccionales hacia la base de datos.
2. Codificar los métodos o funciones para la selección, inserción, actualización y eliminación de datos, provoca que el desarrollo de la aplicación sea monótono, repetitivo y a la vez complejo.
3. Los componentes gráficos que ayudan al programador a desarrollar aplicaciones, permiten que la construcción del software sea más sencilla y rápida.
4. El desarrollo de los métodos de selección, inserción, actualización y eliminación de datos sería más fácil y rápido si se realizara con la ayuda de componentes de asistencia gráfica.

5. El procedimiento para realizar las operaciones con la base de datos sería más fácil si pudiéramos visualizar desde el IDE (entorno de desarrollo integrado) el diagrama del modelo relacional de la base de datos.

- 6. Emplear componentes gráficos especializados en las operaciones transaccionales con base de datos, reduce total complejidad y facilita al desarrollar aplicaciones.
7. La implementación de código fuente y componentes de asistencia gráfica debe ser equilibrada al momento de desarrollar aplicaciones.

Dr. Luis Salazar



UNIVERSIDAD TÉCNICA ESTATAL DE QUEVEDO

Encuesta para evaluar la satisfacción de los programadores de software, en cuanto al procedimiento de acceso a datos cuando desarrollan aplicaciones.

Dirigida a: Los estudiantes de la Carrera de Sistemas de la Facultad de Ciencias de la Ingeniería de la UTEQ.

Responda: Según las siguientes observaciones seleccione una opción, marcando total honestidad en base a la experiencia obtenida a lo largo de sus estudios universitarios.

- 1. El programador obtiene completa confianza y seguridad al emplear solamente código fuente para realizar las operaciones transaccionales hacia la base de datos.
2. Codificar los métodos o funciones para la selección, inserción, actualización y eliminación de datos, provoca que el desarrollo de la aplicación sea monótono, repetitivo y a la vez complejo.
3. Los componentes gráficos que ayudan al programador a desarrollar aplicaciones, permiten que la construcción del software sea más sencilla y rápida.
4. El desarrollo de los métodos de selección, inserción, actualización y eliminación de datos sería más fácil y rápido si se realizara con la ayuda de componentes de asistencia gráfica.

5. El procedimiento para realizar las operaciones con la base de datos sería más fácil si pudiéramos visualizar desde el IDE (entorno de desarrollo integrado) el diagrama del modelo relacional de la base de datos.

- 6. Emplear componentes gráficos especializados en las operaciones transaccionales con base de datos, reduce total complejidad y facilita al desarrollar aplicaciones.
7. La implementación de código fuente y componentes de asistencia gráfica debe ser equilibrada al momento de desarrollar aplicaciones.

Dr. Luis Salazar



UNIVERSIDAD TÉCNICA ESTATAL DE QUEVEDO

Encuesta para evaluar la satisfacción de los programadores de software, en cuanto al procedimiento de acceso a datos cuando desarrollan aplicaciones.

Dirigida a: Los estudiantes de la Carrera de Sistemas de la Facultad de Ciencias de la Ingeniería de la UTEQ.

Respecto: Según las siguientes declaraciones seleccione una opción, marcando total horizontal en base a la experiencia obtenida a lo largo de sus estudios universitarios.

1.- El programador obtiene completa confianza y seguridad al emplear solamente código fuente para realizar las operaciones transaccionales hacia la base de datos.
Totalmente de acuerdo
De acuerdo
Ni en acuerdo, ni en desacuerdo
En desacuerdo
Totalmente en desacuerdo

2.- Diseñar las métricas o funciones para la selección, inserción, actualización y eliminación de datos, provocan que el desarrollo de la aplicación sea monótono, repetitivo y a la vez complejo.
Totalmente de acuerdo
De acuerdo
Ni en acuerdo, ni en desacuerdo
En desacuerdo
Totalmente en desacuerdo

3.- Los componentes gráficos que ayudan al programador a desarrollar aplicaciones, permiten que la construcción del software sea más sencilla y rápida.
Totalmente de acuerdo
De acuerdo
Ni en acuerdo, ni en desacuerdo
En desacuerdo
Totalmente en desacuerdo

4.- El desarrollo de los métodos de selección, inserción, actualización y eliminación de datos sería más fácil y rápida si se realiza con la ayuda de componentes de asistencia gráfica.
Totalmente de acuerdo
De acuerdo
Ni en acuerdo, ni en desacuerdo
En desacuerdo
Totalmente en desacuerdo

5.- El procedimiento para realizar las operaciones con la base de datos sería más fácil si pudieramos visualizar desde el IDE (entorno de desarrollo integrado) el diagrama del modelo relacional de la base de datos.
Totalmente de acuerdo
De acuerdo
Ni en acuerdo, ni en desacuerdo
En desacuerdo
Totalmente en desacuerdo

6.- Emplear componentes gráficos especializados en las operaciones transaccionales con base de datos influyen total (completamente) y facilitan al desarrollar aplicaciones.
Totalmente de acuerdo
De acuerdo
Ni en acuerdo, ni en desacuerdo
En desacuerdo
Totalmente en desacuerdo

7.- La implementación de código fuente y componentes de asistencia gráfica debe ser equitativa al momento de desarrollar aplicaciones.
Totalmente de acuerdo
De acuerdo
Ni en acuerdo, ni en desacuerdo
En desacuerdo
Totalmente en desacuerdo

26 de Septiembre



UNIVERSIDAD TÉCNICA ESTATAL DE QUEVEDO

Encuesta para evaluar la satisfacción de los programadores de software, en cuanto al procedimiento de acceso a datos cuando desarrollan aplicaciones.

Dirigida a: Los estudiantes de la Carrera de Sistemas de la Facultad de Ciencias de la Ingeniería de la UTEQ.

Respecto: Según las siguientes declaraciones seleccione una opción, marcando total horizontal en base a la experiencia obtenida a lo largo de sus estudios universitarios.

1.- El programador obtiene completa confianza y seguridad al emplear solamente código fuente para realizar las operaciones transaccionales hacia la base de datos.
Totalmente de acuerdo
De acuerdo
Ni en acuerdo, ni en desacuerdo
En desacuerdo
Totalmente en desacuerdo

2.- Diseñar los métodos o funciones para la selección, inserción, actualización y eliminación de datos, provocan que el desarrollo de la aplicación sea monótono, repetitivo y a la vez complejo.
Totalmente de acuerdo
De acuerdo
Ni en acuerdo, ni en desacuerdo
En desacuerdo
Totalmente en desacuerdo

3.- Los componentes gráficos que ayudan al programador a desarrollar aplicaciones, permiten que la construcción del software sea más sencilla y rápida.
Totalmente de acuerdo
De acuerdo
Ni en acuerdo, ni en desacuerdo
En desacuerdo
Totalmente en desacuerdo

4.- El desarrollo de los métodos de selección, inserción, actualización y eliminación de datos sería más fácil y rápida si se realiza con la ayuda de componentes de asistencia gráfica.
Totalmente de acuerdo
De acuerdo
Ni en acuerdo, ni en desacuerdo
En desacuerdo
Totalmente en desacuerdo

5.- El procedimiento para realizar las operaciones con la base de datos sería más fácil si pudieramos visualizar desde el IDE (entorno de desarrollo integrado) el diagrama del modelo relacional de la base de datos.
Totalmente de acuerdo
De acuerdo
Ni en acuerdo, ni en desacuerdo
En desacuerdo
Totalmente en desacuerdo

6.- Emplear componentes gráficos especializados en las operaciones transaccionales con base de datos influyen total (completamente) y facilitan al desarrollar aplicaciones.
Totalmente de acuerdo
De acuerdo
Ni en acuerdo, ni en desacuerdo
En desacuerdo
Totalmente en desacuerdo

7.- La implementación de código fuente y componentes de asistencia gráfica debe ser equitativa al momento de desarrollar aplicaciones.
Totalmente de acuerdo
De acuerdo
Ni en acuerdo, ni en desacuerdo
En desacuerdo
Totalmente en desacuerdo

26 de Septiembre



UNIVERSIDAD TÉCNICA ESTATAL DE QUEVEDO

Encuesta para evaluar la satisfacción de los programadores de software, en cuanto al procedimiento de acceso a datos cuando desarrollan aplicaciones.

Dirigida a: Los estudiantes de la Carrera de Sistemas de la Facultad de Ciencias de la Ingeniería de la UTEQ.

Respecto: Según las siguientes declaraciones seleccione una opción, marcando total horizontal en base a la experiencia obtenida a lo largo de sus estudios universitarios.

1.- El programador obtiene completa confianza y seguridad al emplear solamente código fuente para realizar las operaciones transaccionales hacia la base de datos.
Totalmente de acuerdo
De acuerdo
Ni en acuerdo, ni en desacuerdo
En desacuerdo
Totalmente en desacuerdo

2.- Diseñar los métodos o funciones para la selección, inserción, actualización y eliminación de datos, provocan que el desarrollo de la aplicación sea monótono, repetitivo y a la vez complejo.
Totalmente de acuerdo
De acuerdo
Ni en acuerdo, ni en desacuerdo
En desacuerdo
Totalmente en desacuerdo

3.- Los componentes gráficos que ayudan al programador a desarrollar aplicaciones, permiten que la construcción del software sea más sencilla y rápida.
Totalmente de acuerdo
De acuerdo
Ni en acuerdo, ni en desacuerdo
En desacuerdo
Totalmente en desacuerdo

4.- El desarrollo de los métodos de selección, inserción, actualización y eliminación de datos sería más fácil y rápida si se realiza con la ayuda de componentes de asistencia gráfica.
Totalmente de acuerdo
De acuerdo
Ni en acuerdo, ni en desacuerdo
En desacuerdo
Totalmente en desacuerdo

5.- El procedimiento para realizar las operaciones con la base de datos sería más fácil si pudieramos visualizar desde el IDE (entorno de desarrollo integrado) el diagrama del modelo relacional de la base de datos.
Totalmente de acuerdo
De acuerdo
Ni en acuerdo, ni en desacuerdo
En desacuerdo
Totalmente en desacuerdo

6.- Emplear componentes gráficos especializados en las operaciones transaccionales con base de datos influyen total (completamente) y facilitan al desarrollar aplicaciones.
Totalmente de acuerdo
De acuerdo
Ni en acuerdo, ni en desacuerdo
En desacuerdo
Totalmente en desacuerdo

7.- La implementación de código fuente y componentes de asistencia gráfica debe ser equitativa al momento de desarrollar aplicaciones.
Totalmente de acuerdo
De acuerdo
Ni en acuerdo, ni en desacuerdo
En desacuerdo
Totalmente en desacuerdo

26 de Septiembre



UNIVERSIDAD TÉCNICA ESTATAL DE QUEVEDO

Encuesta para evaluar la satisfacción de los programadores de software, en cuanto al procedimiento de acceso a datos cuando desarrollan aplicaciones.

Dirigida a: Los estudiantes de la Carrera de Sistemas de la Facultad de Ciencias de la Ingeniería de la UTEQ.

Respuesta: Según las siguientes declaraciones seleccione una opción, otorgando total honestidad en base a la experiencia obtenida a lo largo de sus estudios universitarios.

1. El programador obtiene completa confianza y seguridad al emplear solamente código fuente para realizar las operaciones transaccionales hacia la base de datos.

- Totalmente de acuerdo
De acuerdo
Ni en acuerdo, ni en desacuerdo
En desacuerdo
Totalmente en desacuerdo

2. Cuidar los métodos o funciones para la selección, inserción, actualización y eliminación de datos, provocan que el desarrollo de la aplicación sea monótona, repetitiva y a la vez compleja.

- Totalmente de acuerdo
De acuerdo
Ni en acuerdo, ni en desacuerdo
En desacuerdo
Totalmente en desacuerdo

3. Los componentes gráficos que ayudan al programador a desarrollar aplicaciones, permiten que la construcción del software sea más sencilla y rápida.

- Totalmente de acuerdo
De acuerdo
Ni en acuerdo, ni en desacuerdo
En desacuerdo
Totalmente en desacuerdo

4. El desarrollo de los métodos de selección, inserción, actualización y eliminación de datos sería más fácil y rápido si se realizara con la ayuda de componentes de asistencia gráfica.

- Totalmente de acuerdo
De acuerdo
Ni en acuerdo, ni en desacuerdo
En desacuerdo
Totalmente en desacuerdo

5. El procedimiento para realizar las operaciones con la base de datos sería más fácil si pudiéramos visualizar desde el IDE (entorno de desarrollo integrado) el diagrama del modelo relacional de la base de datos.

- Totalmente de acuerdo
De acuerdo
Ni en acuerdo, ni en desacuerdo
En desacuerdo
Totalmente en desacuerdo

6. Emplear componentes gráficos especializados en las operaciones transaccionales con base de datos refuerza total comodidad y flexibilidad al desarrollar aplicaciones.

- Totalmente de acuerdo
De acuerdo
Ni en acuerdo, ni en desacuerdo
En desacuerdo
Totalmente en desacuerdo

7. La implementación de código fuente y componentes de asistencia gráfica debe ser equilibrada al momento de desarrollar aplicaciones.

- Totalmente de acuerdo
De acuerdo
Ni en acuerdo, ni en desacuerdo
En desacuerdo
Totalmente en desacuerdo

26 de Setiembre



UNIVERSIDAD TÉCNICA ESTATAL DE QUEVEDO

Encuesta para evaluar la satisfacción de los programadores de software, en cuanto al procedimiento de acceso a datos cuando desarrollan aplicaciones.

Dirigida a: Los estudiantes de la Carrera de Sistemas de la Facultad de Ciencias de la Ingeniería de la UTEQ.

Respuesta: Según las siguientes declaraciones seleccione una opción, otorgando total honestidad en base a la experiencia obtenida a lo largo de sus estudios universitarios.

1. El programador obtiene completa confianza y seguridad al emplear solamente código fuente para realizar las operaciones transaccionales hacia la base de datos.

- Totalmente de acuerdo
De acuerdo
Ni en acuerdo, ni en desacuerdo
En desacuerdo
Totalmente en desacuerdo

2. Cuidar los métodos o funciones para la selección, inserción, actualización y eliminación de datos, provocan que el desarrollo de la aplicación sea monótona, repetitiva y a la vez compleja.

- Totalmente de acuerdo
De acuerdo
Ni en acuerdo, ni en desacuerdo
En desacuerdo
Totalmente en desacuerdo

3. Los componentes gráficos que ayudan al programador a desarrollar aplicaciones, permiten que la construcción del software sea más sencilla y rápida.

- Totalmente de acuerdo
De acuerdo
Ni en acuerdo, ni en desacuerdo
En desacuerdo
Totalmente en desacuerdo

4. El desarrollo de los métodos de selección, inserción, actualización y eliminación de datos sería más fácil y rápido si se realizara con la ayuda de componentes de asistencia gráfica.

- Totalmente de acuerdo
De acuerdo
Ni en acuerdo, ni en desacuerdo
En desacuerdo
Totalmente en desacuerdo

5. El procedimiento para realizar las operaciones con la base de datos sería más fácil si pudiéramos visualizar desde el IDE (entorno de desarrollo integrado) el diagrama del modelo relacional de la base de datos.

- Totalmente de acuerdo
De acuerdo
Ni en acuerdo, ni en desacuerdo
En desacuerdo
Totalmente en desacuerdo

6. Emplear componentes gráficos especializados en las operaciones transaccionales con base de datos refuerza total comodidad y flexibilidad al desarrollar aplicaciones.

- Totalmente de acuerdo
De acuerdo
Ni en acuerdo, ni en desacuerdo
En desacuerdo
Totalmente en desacuerdo

7. La implementación de código fuente y componentes de asistencia gráfica debe ser equilibrada al momento de desarrollar aplicaciones.

- Totalmente de acuerdo
De acuerdo
Ni en acuerdo, ni en desacuerdo
En desacuerdo
Totalmente en desacuerdo

26 de Setiembre



UNIVERSIDAD TÉCNICA ESTATAL DE QUEVEDO

Encuesta para evaluar la satisfacción de los programadores de software, en cuanto al procedimiento de acceso a datos cuando desarrollan aplicaciones.

Dirigida a: Los estudiantes de la Carrera de Sistemas de la Facultad de Ciencias de la Ingeniería de la UTEQ.

Respuesta: Según las siguientes declaraciones seleccione una opción, otorgando total honestidad en base a la experiencia obtenida a lo largo de sus estudios universitarios.

1. El programador obtiene completa confianza y seguridad al emplear solamente código fuente para realizar las operaciones transaccionales hacia la base de datos.

- Totalmente de acuerdo
De acuerdo
Ni en acuerdo, ni en desacuerdo
En desacuerdo
Totalmente en desacuerdo

2. Cuidar los métodos o funciones para la selección, inserción, actualización y eliminación de datos, provocan que el desarrollo de la aplicación sea monótona, repetitiva y a la vez compleja.

- Totalmente de acuerdo
De acuerdo
Ni en acuerdo, ni en desacuerdo
En desacuerdo
Totalmente en desacuerdo

3. Los componentes gráficos que ayudan al programador a desarrollar aplicaciones, permiten que la construcción del software sea más sencilla y rápida.

- Totalmente de acuerdo
De acuerdo
Ni en acuerdo, ni en desacuerdo
En desacuerdo
Totalmente en desacuerdo

4. El desarrollo de los métodos de selección, inserción, actualización y eliminación de datos sería más fácil y rápido si se realizara con la ayuda de componentes de asistencia gráfica.

- Totalmente de acuerdo
De acuerdo
Ni en acuerdo, ni en desacuerdo
En desacuerdo
Totalmente en desacuerdo

5. El procedimiento para realizar las operaciones con la base de datos sería más fácil si pudiéramos visualizar desde el IDE (entorno de desarrollo integrado) el diagrama del modelo relacional de la base de datos.

- Totalmente de acuerdo
De acuerdo
Ni en acuerdo, ni en desacuerdo
En desacuerdo
Totalmente en desacuerdo

6. Emplear componentes gráficos especializados en las operaciones transaccionales con base de datos refuerza total comodidad y flexibilidad al desarrollar aplicaciones.

- Totalmente de acuerdo
De acuerdo
Ni en acuerdo, ni en desacuerdo
En desacuerdo
Totalmente en desacuerdo

7. La implementación de código fuente y componentes de asistencia gráfica debe ser equilibrada al momento de desarrollar aplicaciones.

- Totalmente de acuerdo
De acuerdo
Ni en acuerdo, ni en desacuerdo
En desacuerdo
Totalmente en desacuerdo

26 de Setiembre



UNIVERSIDAD TÉCNICA ESTATAL DE QUEVEDO

Encuesta para evaluar la satisfacción de los programadores de software, en cuanto al procedimiento de acceso a datos cuando desarrollan aplicaciones.

Dirigida a: Los estudiantes de la Carrera de Sistemas de la Facultad de Ciencias de la Ingeniería de la UTEQ.

Resúmenes: Según las siguientes declaraciones seleccione una opción, asignando total honestidad en base a la experiencia obtenida a lo largo de sus estudios universitarios.

1. El programador define completa confianza y seguridad al emplear solamente código fuente para realizar las operaciones transaccionales hacia la base de datos.
Totalmente de acuerdo
De acuerdo
Ni en acuerdo, ni en desacuerdo
En desacuerdo
Totalmente en desacuerdo

2. Cuidar los métodos o funciones para la selección, inserción, actualización y eliminación de datos, previene que el desarrollo de la aplicación sea monótona, repetitiva y a la vez compleja.
Totalmente de acuerdo
De acuerdo
Ni en acuerdo, ni en desacuerdo
En desacuerdo
Totalmente en desacuerdo

3. Los componentes gráficos que ayudan al programador a desarrollar aplicaciones, permiten que la construcción del software sea más sencilla y rápida.
Totalmente de acuerdo
De acuerdo
Ni en acuerdo, ni en desacuerdo
En desacuerdo
Totalmente en desacuerdo

4. El desarrollo de los métodos de selección, inserción, actualización y eliminación de datos se realiza más fácil y rápida si se realiza con la ayuda de componentes de asistencia gráfica.
Totalmente de acuerdo
De acuerdo
Ni en acuerdo, ni en desacuerdo
En desacuerdo
Totalmente en desacuerdo

5. El procedimiento para realizar las operaciones con la base de datos se realiza más fácil si podemos visualizar desde el IDE (entorno de desarrollo integrado) el diagrama del modelo relacional de la base de datos.
Totalmente de acuerdo
De acuerdo
Ni en acuerdo, ni en desacuerdo
En desacuerdo
Totalmente en desacuerdo

6. Emplear componentes gráficos especializados en las operaciones transaccionales con base de datos influyen total comodidad y facilidad al desarrollar aplicaciones.
Totalmente de acuerdo
De acuerdo
Ni en acuerdo, ni en desacuerdo
En desacuerdo
Totalmente en desacuerdo

7. La implementación de código fuente y componentes de asistencia gráfica debe ser equitativa al momento de desarrollar aplicaciones.
Totalmente de acuerdo
De acuerdo
Ni en acuerdo, ni en desacuerdo
En desacuerdo
Totalmente en desacuerdo

2do An. Sistema



UNIVERSIDAD TÉCNICA ESTATAL DE QUEVEDO

Encuesta para evaluar la satisfacción de los programadores de software, en cuanto al procedimiento de acceso a datos cuando desarrollan aplicaciones.

Dirigida a: Los estudiantes de la Carrera de Sistemas de la Facultad de Ciencias de la Ingeniería de la UTEQ.

Resúmenes: Según las siguientes declaraciones seleccione una opción, asignando total honestidad en base a la experiencia obtenida a lo largo de sus estudios universitarios.

1. El programador define completa confianza y seguridad al emplear solamente código fuente para realizar las operaciones transaccionales hacia la base de datos.
Totalmente de acuerdo
De acuerdo
Ni en acuerdo, ni en desacuerdo
En desacuerdo
Totalmente en desacuerdo

2. Cuidar los métodos o funciones para la selección, inserción, actualización y eliminación de datos, previene que el desarrollo de la aplicación sea monótona, repetitiva y a la vez compleja.
Totalmente de acuerdo
De acuerdo
Ni en acuerdo, ni en desacuerdo
En desacuerdo
Totalmente en desacuerdo

3. Los componentes gráficos que ayudan al programador a desarrollar aplicaciones, permiten que la construcción del software sea más sencilla y rápida.
Totalmente de acuerdo
De acuerdo
Ni en acuerdo, ni en desacuerdo
En desacuerdo
Totalmente en desacuerdo

4. El desarrollo de los métodos de selección, inserción, actualización y eliminación de datos se realiza más fácil y rápida si se realiza con la ayuda de componentes de asistencia gráfica.
Totalmente de acuerdo
De acuerdo
Ni en acuerdo, ni en desacuerdo
En desacuerdo
Totalmente en desacuerdo

5. El procedimiento para realizar las operaciones con la base de datos se realiza más fácil si podemos visualizar desde el IDE (entorno de desarrollo integrado) el diagrama del modelo relacional de la base de datos.
Totalmente de acuerdo
De acuerdo
Ni en acuerdo, ni en desacuerdo
En desacuerdo
Totalmente en desacuerdo

6. Emplear componentes gráficos especializados en las operaciones transaccionales con base de datos influyen total comodidad y facilidad al desarrollar aplicaciones.
Totalmente de acuerdo
De acuerdo
Ni en acuerdo, ni en desacuerdo
En desacuerdo
Totalmente en desacuerdo

7. La implementación de código fuente y componentes de asistencia gráfica debe ser equitativa al momento de desarrollar aplicaciones.
Totalmente de acuerdo
De acuerdo
Ni en acuerdo, ni en desacuerdo
En desacuerdo
Totalmente en desacuerdo

2do An. Sistema



UNIVERSIDAD TÉCNICA ESTATAL DE QUEVEDO

Encuesta para evaluar la satisfacción de los programadores de software, en cuanto al procedimiento de acceso a datos cuando desarrollan aplicaciones.

Dirigida a: Los estudiantes de la Carrera de Sistemas de la Facultad de Ciencias de la Ingeniería de la UTEQ.

Resúmenes: Según las siguientes declaraciones seleccione una opción, asignando total honestidad en base a la experiencia obtenida a lo largo de sus estudios universitarios.

1. El programador define completa confianza y seguridad al emplear solamente código fuente para realizar las operaciones transaccionales hacia la base de datos.
Totalmente de acuerdo
De acuerdo
Ni en acuerdo, ni en desacuerdo
En desacuerdo
Totalmente en desacuerdo

2. Cuidar los métodos o funciones para la selección, inserción, actualización y eliminación de datos, previene que el desarrollo de la aplicación sea monótona, repetitiva y a la vez compleja.
Totalmente de acuerdo
De acuerdo
Ni en acuerdo, ni en desacuerdo
En desacuerdo
Totalmente en desacuerdo

3. Los componentes gráficos que ayudan al programador a desarrollar aplicaciones, permiten que la construcción del software sea más sencilla y rápida.
Totalmente de acuerdo
De acuerdo
Ni en acuerdo, ni en desacuerdo
En desacuerdo
Totalmente en desacuerdo

4. El desarrollo de los métodos de selección, inserción, actualización y eliminación de datos se realiza más fácil y rápida si se realiza con la ayuda de componentes de asistencia gráfica.
Totalmente de acuerdo
De acuerdo
Ni en acuerdo, ni en desacuerdo
En desacuerdo
Totalmente en desacuerdo

5. El procedimiento para realizar las operaciones con la base de datos se realiza más fácil si podemos visualizar desde el IDE (entorno de desarrollo integrado) el diagrama del modelo relacional de la base de datos.
Totalmente de acuerdo
De acuerdo
Ni en acuerdo, ni en desacuerdo
En desacuerdo
Totalmente en desacuerdo

6. Emplear componentes gráficos especializados en las operaciones transaccionales con base de datos influyen total comodidad y facilidad al desarrollar aplicaciones.
Totalmente de acuerdo
De acuerdo
Ni en acuerdo, ni en desacuerdo
En desacuerdo
Totalmente en desacuerdo

7. La implementación de código fuente y componentes de asistencia gráfica debe ser equitativa al momento de desarrollar aplicaciones.
Totalmente de acuerdo
De acuerdo
Ni en acuerdo, ni en desacuerdo
En desacuerdo
Totalmente en desacuerdo

2do An. Sistema



UNIVERSIDAD TÉCNICA ESTATAL DE QUEVEDO

Encuesta para evaluar la satisfacción de los programadores de software, en cuanto al procedimiento de acceso a datos cuando desarrollan aplicaciones.

Dirigida a: Los estudiantes de la Carrera de Sistemas de la Facultad de Ciencias de la Ingeniería de la UTEQ.

Responda: Según las siguientes observaciones seleccione una opción, marcando total honestidad en base a la experiencia obtenida a lo largo de sus estudios universitarios.

1.- El programador obtiene completa confianza y seguridad al emplear solamente código fuente para realizar las operaciones transaccionales hacia la base de datos.
[] Totalmente de acuerdo
[] De acuerdo
[] Ni en acuerdo, ni en desacuerdo
[] En desacuerdo
[] Totalmente en desacuerdo

2.- Codificar los métodos o funciones para la selección, inserción, actualización y eliminación de datos, provocan que el desarrollo de la aplicación sea monótona, repetitiva y a la vez compleja.
[] Totalmente de acuerdo
[] De acuerdo
[] Ni en acuerdo, ni en desacuerdo
[] En desacuerdo
[] Totalmente en desacuerdo

3.- Los componentes gráficos que ayudan al programador a desarrollar aplicaciones, permiten que la construcción del software sea más sencilla y rápida.
[] Totalmente de acuerdo
[] De acuerdo
[] Ni en acuerdo, ni en desacuerdo
[] En desacuerdo
[] Totalmente en desacuerdo

4.- El desarrollo de los métodos de selección, inserción, actualización y eliminación de datos sería más fácil y rápida si se realizara con la ayuda de componentes de asistencia gráfica.
[] Totalmente de acuerdo
[] De acuerdo
[] Ni en acuerdo, ni en desacuerdo
[] En desacuerdo
[] Totalmente en desacuerdo

5.- El procedimiento para realizar las operaciones con la base de datos sería más fácil si pudiéramos visualizar desde el ECE (botón de desarrollo integrado) el diagrama del modelo relacional de la base de datos.
[] Totalmente de acuerdo
[] De acuerdo
[] Ni en acuerdo, ni en desacuerdo
[] En desacuerdo
[] Totalmente en desacuerdo

6.- Emplear componentes gráficos especializadas en las operaciones transaccionales con base de datos influyen total confianza y facilidad al desarrollar aplicaciones.
[] Totalmente de acuerdo
[] De acuerdo
[] Ni en acuerdo, ni en desacuerdo
[] En desacuerdo
[] Totalmente en desacuerdo

7.- La implementación de código fuente y componentes de asistencia gráfica debe ser equilibrada al momento de desarrollar aplicaciones.
[] Totalmente de acuerdo
[] De acuerdo
[] Ni en acuerdo, ni en desacuerdo
[] En desacuerdo
[] Totalmente en desacuerdo

José Luis Salazar



UNIVERSIDAD TÉCNICA ESTATAL DE QUEVEDO

Encuesta para evaluar la satisfacción de los programadores de software, en cuanto al procedimiento de acceso a datos cuando desarrollan aplicaciones.

Dirigida a: Los estudiantes de la Carrera de Sistemas de la Facultad de Ciencias de la Ingeniería de la UTEQ.

Responda: Según las siguientes observaciones seleccione una opción, marcando total honestidad en base a la experiencia obtenida a lo largo de sus estudios universitarios.

1.- El programador obtiene completa confianza y seguridad al emplear solamente código fuente para realizar las operaciones transaccionales hacia la base de datos.
[] Totalmente de acuerdo
[] De acuerdo
[] Ni en acuerdo, ni en desacuerdo
[] En desacuerdo
[] Totalmente en desacuerdo

2.- Codificar los métodos o funciones para la selección, inserción, actualización y eliminación de datos, provocan que el desarrollo de la aplicación sea monótona, repetitiva y a la vez compleja.
[] Totalmente de acuerdo
[] De acuerdo
[] Ni en acuerdo, ni en desacuerdo
[] En desacuerdo
[] Totalmente en desacuerdo

3.- Los componentes gráficos que ayudan al programador a desarrollar aplicaciones, permiten que la construcción del software sea más sencilla y rápida.
[] Totalmente de acuerdo
[] De acuerdo
[] Ni en acuerdo, ni en desacuerdo
[] En desacuerdo
[] Totalmente en desacuerdo

4.- El desarrollo de los métodos de selección, inserción, actualización y eliminación de datos sería más fácil y rápida si se realizara con la ayuda de componentes de asistencia gráfica.
[] Totalmente de acuerdo
[] De acuerdo
[] Ni en acuerdo, ni en desacuerdo
[] En desacuerdo
[] Totalmente en desacuerdo

5.- El procedimiento para realizar las operaciones con la base de datos sería más fácil si pudiéramos visualizar desde el ECE (botón de desarrollo integrado) el diagrama del modelo relacional de la base de datos.
[] Totalmente de acuerdo
[] De acuerdo
[] Ni en acuerdo, ni en desacuerdo
[] En desacuerdo
[] Totalmente en desacuerdo

6.- Emplear componentes gráficos especializadas en las operaciones transaccionales con base de datos influyen total confianza y facilidad al desarrollar aplicaciones.
[] Totalmente de acuerdo
[] De acuerdo
[] Ni en acuerdo, ni en desacuerdo
[] En desacuerdo
[] Totalmente en desacuerdo

7.- La implementación de código fuente y componentes de asistencia gráfica debe ser equilibrada al momento de desarrollar aplicaciones.
[] Totalmente de acuerdo
[] De acuerdo
[] Ni en acuerdo, ni en desacuerdo
[] En desacuerdo
[] Totalmente en desacuerdo

José Luis Salazar



UNIVERSIDAD TÉCNICA ESTATAL DE QUEVEDO

Encuesta para evaluar la satisfacción de los programadores de software, en cuanto al procedimiento de acceso a datos cuando desarrollan aplicaciones.

Dirigida a: Los estudiantes de la Carrera de Sistemas de la Facultad de Ciencias de la Ingeniería de la UTEQ.

Responda: Según las siguientes observaciones seleccione una opción, marcando total honestidad en base a la experiencia obtenida a lo largo de sus estudios universitarios.

1.- El programador obtiene completa confianza y seguridad al emplear solamente código fuente para realizar las operaciones transaccionales hacia la base de datos.
[] Totalmente de acuerdo
[] De acuerdo
[] Ni en acuerdo, ni en desacuerdo
[] En desacuerdo
[] Totalmente en desacuerdo

2.- Codificar los métodos o funciones para la selección, inserción, actualización y eliminación de datos, provocan que el desarrollo de la aplicación sea monótona, repetitiva y a la vez compleja.
[] Totalmente de acuerdo
[] De acuerdo
[] Ni en acuerdo, ni en desacuerdo
[] En desacuerdo
[] Totalmente en desacuerdo

3.- Los componentes gráficos que ayudan al programador a desarrollar aplicaciones, permiten que la construcción del software sea más sencilla y rápida.
[] Totalmente de acuerdo
[] De acuerdo
[] Ni en acuerdo, ni en desacuerdo
[] En desacuerdo
[] Totalmente en desacuerdo

4.- El desarrollo de los métodos de selección, inserción, actualización y eliminación de datos sería más fácil y rápida si se realizara con la ayuda de componentes de asistencia gráfica.
[] Totalmente de acuerdo
[] De acuerdo
[] Ni en acuerdo, ni en desacuerdo
[] En desacuerdo
[] Totalmente en desacuerdo

5.- El procedimiento para realizar las operaciones con la base de datos sería más fácil si pudiéramos visualizar desde el ECE (botón de desarrollo integrado) el diagrama del modelo relacional de la base de datos.
[] Totalmente de acuerdo
[] De acuerdo
[] Ni en acuerdo, ni en desacuerdo
[] En desacuerdo
[] Totalmente en desacuerdo

6.- Emplear componentes gráficos especializadas en las operaciones transaccionales con base de datos influyen total confianza y facilidad al desarrollar aplicaciones.
[] Totalmente de acuerdo
[] De acuerdo
[] Ni en acuerdo, ni en desacuerdo
[] En desacuerdo
[] Totalmente en desacuerdo

7.- La implementación de código fuente y componentes de asistencia gráfica debe ser equilibrada al momento de desarrollar aplicaciones.
[] Totalmente de acuerdo
[] De acuerdo
[] Ni en acuerdo, ni en desacuerdo
[] En desacuerdo
[] Totalmente en desacuerdo

José Luis Salazar



UNIVERSIDAD TÉCNICA ESTATAL DE QUEVEDO

Encuesta para evaluar la satisfacción de los programadores de software, en cuanto al procedimiento de acceso a datos cuando desarrollan aplicaciones.

Dirigida a: Los estudiantes de la Carrera de Sistemas de la Facultad de Ciencias de la Ingeniería de la UTEQ.

Resultados: Según las siguientes declaraciones seleccione una opción, sumando total honestidad en base a la experiencia obtenida a lo largo de sus estudios universitarios.

1.- El programador obtiene completa confianza y seguridad al emplear solamente código fuente para realizar las operaciones transaccionales hacia la base de datos.

- Totalmente de acuerdo
De acuerdo
Ni en acuerdo, ni en desacuerdo
En desacuerdo
Totalmente en desacuerdo

2.- Codificar los métodos o funciones para la selección, inserción, actualización y eliminación de datos, provocan que el desarrollo de la aplicación sea monótona, repetitiva y a la vez compleja.

- Totalmente de acuerdo
De acuerdo
Ni en acuerdo, ni en desacuerdo
En desacuerdo
Totalmente en desacuerdo

3.- Los componentes gráficos que ayudan al programador a desarrollar aplicaciones, permiten que la construcción del software sea más sencilla y rápida.

- Totalmente de acuerdo
De acuerdo
Ni en acuerdo, ni en desacuerdo
En desacuerdo
Totalmente en desacuerdo

4.- El desarrollo de los métodos de selección, inserción, actualización y eliminación de datos sería más fácil y rápido si se realizara con la ayuda de componentes de asistencia gráfica.

- Totalmente de acuerdo
De acuerdo
Ni en acuerdo, ni en desacuerdo
En desacuerdo
Totalmente en desacuerdo

5.- El procedimiento para realizar las operaciones con la base de datos sería más fácil si incluieramos visualizar desde el IDE (entorno de desarrollo integrado) el diagrama del modelo relacional de la base de datos.

- Totalmente de acuerdo
De acuerdo
Ni en acuerdo, ni en desacuerdo
En desacuerdo
Totalmente en desacuerdo

6.- Emplear componentes gráficos especializados en las operaciones transaccionales con base de datos influyen total confianza y facilidad al desarrollar aplicaciones.

- Totalmente de acuerdo
De acuerdo
Ni en acuerdo, ni en desacuerdo
En desacuerdo
Totalmente en desacuerdo

7.- La implementación de código fuente y componentes de asistencia gráfica debe ser equilibrada al momento de desarrollar aplicaciones.

- Totalmente de acuerdo
De acuerdo
Ni en acuerdo, ni en desacuerdo
En desacuerdo
Totalmente en desacuerdo

24 de Julio



UNIVERSIDAD TÉCNICA ESTATAL DE QUEVEDO

Encuesta para evaluar la satisfacción de los programadores de software, en cuanto al procedimiento de acceso a datos cuando desarrollan aplicaciones.

Dirigida a: Los estudiantes de la Carrera de Sistemas de la Facultad de Ciencias de la Ingeniería de la UTEQ.

Resultados: Según las siguientes declaraciones seleccione una opción, sumando total honestidad en base a la experiencia obtenida a lo largo de sus estudios universitarios.

1.- El programador obtiene completa confianza y seguridad al emplear solamente código fuente para realizar las operaciones transaccionales hacia la base de datos.

- Totalmente de acuerdo
De acuerdo
Ni en acuerdo, ni en desacuerdo
En desacuerdo
Totalmente en desacuerdo

2.- Codificar los métodos o funciones para la selección, inserción, actualización y eliminación de datos, provocan que el desarrollo de la aplicación sea monótona, repetitiva y a la vez compleja.

- Totalmente de acuerdo
De acuerdo
Ni en acuerdo, ni en desacuerdo
En desacuerdo
Totalmente en desacuerdo

3.- Los componentes gráficos que ayudan al programador a desarrollar aplicaciones, permiten que la construcción del software sea más sencilla y rápida.

- Totalmente de acuerdo
De acuerdo
Ni en acuerdo, ni en desacuerdo
En desacuerdo
Totalmente en desacuerdo

4.- El desarrollo de los métodos de selección, inserción, actualización y eliminación de datos sería más fácil y rápido si se realizara con la ayuda de componentes de asistencia gráfica.

- Totalmente de acuerdo
De acuerdo
Ni en acuerdo, ni en desacuerdo
En desacuerdo
Totalmente en desacuerdo

5.- El procedimiento para realizar las operaciones con la base de datos sería más fácil si incluieramos visualizar desde el IDE (entorno de desarrollo integrado) el diagrama del modelo relacional de la base de datos.

- Totalmente de acuerdo
De acuerdo
Ni en acuerdo, ni en desacuerdo
En desacuerdo
Totalmente en desacuerdo

6.- Emplear componentes gráficos especializados en las operaciones transaccionales con base de datos influyen total confianza y facilidad al desarrollar aplicaciones.

- Totalmente de acuerdo
De acuerdo
Ni en acuerdo, ni en desacuerdo
En desacuerdo
Totalmente en desacuerdo

7.- La implementación de código fuente y componentes de asistencia gráfica debe ser equilibrada al momento de desarrollar aplicaciones.

- Totalmente de acuerdo
De acuerdo
Ni en acuerdo, ni en desacuerdo
En desacuerdo
Totalmente en desacuerdo

24 de Julio



UNIVERSIDAD TÉCNICA ESTATAL DE QUEVEDO

Encuesta para evaluar la satisfacción de los programadores de software, en cuanto al procedimiento de acceso a datos cuando desarrollan aplicaciones.

Dirigida a: Los estudiantes de la Carrera de Sistemas de la Facultad de Ciencias de la Ingeniería de la UTEQ.

Resultados: Según las siguientes declaraciones seleccione una opción, sumando total honestidad en base a la experiencia obtenida a lo largo de sus estudios universitarios.

1.- El programador obtiene completa confianza y seguridad al emplear solamente código fuente para realizar las operaciones transaccionales hacia la base de datos.

- Totalmente de acuerdo
De acuerdo
Ni en acuerdo, ni en desacuerdo
En desacuerdo
Totalmente en desacuerdo

2.- Codificar los métodos o funciones para la selección, inserción, actualización y eliminación de datos, provocan que el desarrollo de la aplicación sea monótona, repetitiva y a la vez compleja.

- Totalmente de acuerdo
De acuerdo
Ni en acuerdo, ni en desacuerdo
En desacuerdo
Totalmente en desacuerdo

3.- Los componentes gráficos que ayudan al programador a desarrollar aplicaciones, permiten que la construcción del software sea más sencilla y rápida.

- Totalmente de acuerdo
De acuerdo
Ni en acuerdo, ni en desacuerdo
En desacuerdo
Totalmente en desacuerdo

4.- El desarrollo de los métodos de selección, inserción, actualización y eliminación de datos sería más fácil y rápido si se realizara con la ayuda de componentes de asistencia gráfica.

- Totalmente de acuerdo
De acuerdo
Ni en acuerdo, ni en desacuerdo
En desacuerdo
Totalmente en desacuerdo

5.- El procedimiento para realizar las operaciones con la base de datos sería más fácil si incluieramos visualizar desde el IDE (entorno de desarrollo integrado) el diagrama del modelo relacional de la base de datos.

- Totalmente de acuerdo
De acuerdo
Ni en acuerdo, ni en desacuerdo
En desacuerdo
Totalmente en desacuerdo

6.- Emplear componentes gráficos especializados en las operaciones transaccionales con base de datos influyen total confianza y facilidad al desarrollar aplicaciones.

- Totalmente de acuerdo
De acuerdo
Ni en acuerdo, ni en desacuerdo
En desacuerdo
Totalmente en desacuerdo

7.- La implementación de código fuente y componentes de asistencia gráfica debe ser equilibrada al momento de desarrollar aplicaciones.

- Totalmente de acuerdo
De acuerdo
Ni en acuerdo, ni en desacuerdo
En desacuerdo
Totalmente en desacuerdo

30 de Julio



UNIVERSIDAD TÉCNICA ESTATAL DE QUEVEDO

Encuesta para evaluar la satisfacción de los programadores de software, en cuanto al procedimiento de acceso a datos cuando desarrollan aplicaciones.

Dirigida a: Los estudiantes de la Carrera de Sistemas de la Facultad de Ciencias de la Ingeniería de la UTEQ.

Respecto a: Según las siguientes declaraciones seleccione una opción, asumiendo total honestidad en base a la experiencia obtenida a lo largo de sus estudios universitarios.

1. El programador siente completa confianza y seguridad al emplear solamente código fuente para realizar las operaciones transaccionales hacia la base de datos.

- Totalmente de acuerdo
De acuerdo
Ni en acuerdo, ni en desacuerdo
En desacuerdo
Totalmente en desacuerdo

2. Codificar las métodos o funciones para la selección, inserción, actualización y eliminación de datos, provocan que el desarrollo de la aplicación sea monótona, repetitiva y a la vez compleja.

- Totalmente de acuerdo
De acuerdo
Ni en acuerdo, ni en desacuerdo
En desacuerdo
Totalmente en desacuerdo

3. Los componentes gráficos que ayudan al programador a desarrollar aplicaciones, permiten que la construcción del software sea más sencilla y rápida.

- Totalmente de acuerdo
De acuerdo
Ni en acuerdo, ni en desacuerdo
En desacuerdo
Totalmente en desacuerdo

4. El desarrollo de los métodos de selección, inserción, actualización y eliminación de datos sería más fácil y rápido si se realizara con la ayuda de componentes de asistencia gráfica.

- Totalmente de acuerdo
De acuerdo
Ni en acuerdo, ni en desacuerdo
En desacuerdo
Totalmente en desacuerdo

5. El procedimiento para realizar las operaciones con la base de datos sería más fácil si pudiéramos visualizar desde el IDE (entorno de desarrollo integrado) el diagrama del modelo relacional de la base de datos.

- Totalmente de acuerdo
De acuerdo
Ni en acuerdo, ni en desacuerdo
En desacuerdo
Totalmente en desacuerdo

6. Emplear componentes gráficos especializados en las operaciones transaccionales con base de datos refuerza total comodidad y facilidad al desarrollar aplicaciones.

- Totalmente de acuerdo
De acuerdo
Ni en acuerdo, ni en desacuerdo
En desacuerdo
Totalmente en desacuerdo

7. La implementación de código fuente y componentes de asistencia gráfica debe ser equilibrada al momento de desarrollar aplicaciones.

- Totalmente de acuerdo
De acuerdo
Ni en acuerdo, ni en desacuerdo
En desacuerdo
Totalmente en desacuerdo

3ra. Act. Sistema



UNIVERSIDAD TÉCNICA ESTATAL DE QUEVEDO

Encuesta para evaluar la satisfacción de los programadores de software, en cuanto al procedimiento de acceso a datos cuando desarrollan aplicaciones.

Dirigida a: Los estudiantes de la Carrera de Sistemas de la Facultad de Ciencias de la Ingeniería de la UTEQ.

Respecto a: Según las siguientes declaraciones seleccione una opción, asumiendo total honestidad en base a la experiencia obtenida a lo largo de sus estudios universitarios.

1. El programador siente completa confianza y seguridad al emplear solamente código fuente para realizar las operaciones transaccionales hacia la base de datos.

- Totalmente de acuerdo
De acuerdo
Ni en acuerdo, ni en desacuerdo
En desacuerdo
Totalmente en desacuerdo

2. Codificar las métodos o funciones para la selección, inserción, actualización y eliminación de datos, provocan que el desarrollo de la aplicación sea monótona, repetitiva y a la vez compleja.

- Totalmente de acuerdo
De acuerdo
Ni en acuerdo, ni en desacuerdo
En desacuerdo
Totalmente en desacuerdo

3. Los componentes gráficos que ayudan al programador a desarrollar aplicaciones, permiten que la construcción del software sea más sencilla y rápida.

- Totalmente de acuerdo
De acuerdo
Ni en acuerdo, ni en desacuerdo
En desacuerdo
Totalmente en desacuerdo

4. El desarrollo de los métodos de selección, inserción, actualización y eliminación de datos sería más fácil y rápido si se realizara con la ayuda de componentes de asistencia gráfica.

- Totalmente de acuerdo
De acuerdo
Ni en acuerdo, ni en desacuerdo
En desacuerdo
Totalmente en desacuerdo

5. El procedimiento para realizar las operaciones con la base de datos sería más fácil si pudiéramos visualizar desde el IDE (entorno de desarrollo integrado) el diagrama del modelo relacional de la base de datos.

- Totalmente de acuerdo
De acuerdo
Ni en acuerdo, ni en desacuerdo
En desacuerdo
Totalmente en desacuerdo

6. Emplear componentes gráficos especializados en las operaciones transaccionales con base de datos refuerza total comodidad y facilidad al desarrollar aplicaciones.

- Totalmente de acuerdo
De acuerdo
Ni en acuerdo, ni en desacuerdo
En desacuerdo
Totalmente en desacuerdo

7. La implementación de código fuente y componentes de asistencia gráfica debe ser equilibrada al momento de desarrollar aplicaciones.

- Totalmente de acuerdo
De acuerdo
Ni en acuerdo, ni en desacuerdo
En desacuerdo
Totalmente en desacuerdo

3ra. Act. Sistema



UNIVERSIDAD TÉCNICA ESTATAL DE QUEVEDO

Encuesta para evaluar la satisfacción de los programadores de software, en cuanto al procedimiento de acceso a datos cuando desarrollan aplicaciones.

Dirigida a: Los estudiantes de la Carrera de Sistemas de la Facultad de Ciencias de la Ingeniería de la UTEQ.

Respecto a: Según las siguientes declaraciones seleccione una opción, asumiendo total honestidad en base a la experiencia obtenida a lo largo de sus estudios universitarios.

1. El programador siente completa confianza y seguridad al emplear solamente código fuente para realizar las operaciones transaccionales hacia la base de datos.

- Totalmente de acuerdo
De acuerdo
Ni en acuerdo, ni en desacuerdo
En desacuerdo
Totalmente en desacuerdo

2. Codificar los métodos o funciones para la selección, inserción, actualización y eliminación de datos, provocan que el desarrollo de la aplicación sea monótona, repetitiva y a la vez compleja.

- Totalmente de acuerdo
De acuerdo
Ni en acuerdo, ni en desacuerdo
En desacuerdo
Totalmente en desacuerdo

3. Los componentes gráficos que ayudan al programador a desarrollar aplicaciones, permiten que la construcción del software sea más sencilla y rápida.

- Totalmente de acuerdo
De acuerdo
Ni en acuerdo, ni en desacuerdo
En desacuerdo
Totalmente en desacuerdo

4. El desarrollo de los métodos de selección, inserción, actualización y eliminación de datos sería más fácil y rápido si se realizara con la ayuda de componentes de asistencia gráfica.

- Totalmente de acuerdo
De acuerdo
Ni en acuerdo, ni en desacuerdo
En desacuerdo
Totalmente en desacuerdo

5. El procedimiento para realizar las operaciones con la base de datos sería más fácil si pudiéramos visualizar desde el IDE (entorno de desarrollo integrado) el diagrama del modelo relacional de la base de datos.

- Totalmente de acuerdo
De acuerdo
Ni en acuerdo, ni en desacuerdo
En desacuerdo
Totalmente en desacuerdo

6. Emplear componentes gráficos especializados en las operaciones transaccionales con base de datos refuerza total comodidad y facilidad al desarrollar aplicaciones.

- Totalmente de acuerdo
De acuerdo
Ni en acuerdo, ni en desacuerdo
En desacuerdo
Totalmente en desacuerdo

7. La implementación de código fuente y componentes de asistencia gráfica debe ser equilibrada al momento de desarrollar aplicaciones.

- Totalmente de acuerdo
De acuerdo
Ni en acuerdo, ni en desacuerdo
En desacuerdo
Totalmente en desacuerdo

3ra. Act. Sistema



UNIVERSIDAD TÉCNICA ESTATAL DE QUEVEDO

Encuesta para evaluar la satisfacción de los programadores de software, en cuanto al procedimiento de acceso a datos cuando desarrollan aplicaciones.

Dirigida a: Los estudiantes de la Carrera de Sistemas de la Facultad de Ciencias de la Ingeniería de la UTEQ.

Responda: Según las siguientes declaraciones seleccione una opción, asignando total honestidad en base a la experiencia obtenida a lo largo de sus estudios universitarios.

1. El programador obtiene completa confianza y seguridad al emplear solamente código fuente para realizar las operaciones transaccionales hacia la base de datos.
Totalmente de acuerdo
De acuerdo
Ni en acuerdo, ni en desacuerdo
En desacuerdo
Totalmente en desacuerdo

2. Cualificar los métodos o funciones para la selección, inserción, actualización y eliminación de datos, provocan que el desarrollo de la aplicación sea monótona, repetitiva y a la vez compleja.
Totalmente de acuerdo
De acuerdo
Ni en acuerdo, ni en desacuerdo
En desacuerdo
Totalmente en desacuerdo

3. Los componentes gráficos que ayudan al programador a desarrollar aplicaciones, permiten que la construcción del software sea más sencilla y rápida.
Totalmente de acuerdo
De acuerdo
Ni en acuerdo, ni en desacuerdo
En desacuerdo
Totalmente en desacuerdo

4. El desarrollo de los métodos de selección, inserción, actualización y eliminación de datos sería más fácil y rápido si se realizara con la ayuda de componentes de asistencia gráfica.
Totalmente de acuerdo
De acuerdo
Ni en acuerdo, ni en desacuerdo
En desacuerdo
Totalmente en desacuerdo

5. El procedimiento para realizar las operaciones con la base de datos sería más fácil si pudiéramos visualizar desde el IDE (entorno de desarrollo integrado) el diagrama del modelo relacional de la base de datos.
Totalmente de acuerdo
De acuerdo
Ni en acuerdo, ni en desacuerdo
En desacuerdo
Totalmente en desacuerdo

6. Emplear componentes gráficos especializados en las operaciones transaccionales con base de datos influyen total comodidad y facilidad al desarrollar aplicaciones.
Totalmente de acuerdo
De acuerdo
Ni en acuerdo, ni en desacuerdo
En desacuerdo
Totalmente en desacuerdo

7. La implementación de código fuente y componentes de asistencia gráfica debe ser equilibrada al momento de desarrollar aplicaciones.
Totalmente de acuerdo
De acuerdo
Ni en acuerdo, ni en desacuerdo
En desacuerdo
Totalmente en desacuerdo



UNIVERSIDAD TÉCNICA ESTATAL DE QUEVEDO

Encuesta para evaluar la satisfacción de los programadores de software, en cuanto al procedimiento de acceso a datos cuando desarrollan aplicaciones.

Dirigida a: Los estudiantes de la Carrera de Sistemas de la Facultad de Ciencias de la Ingeniería de la UTEQ.

Responda: Según las siguientes declaraciones seleccione una opción, asignando total honestidad en base a la experiencia obtenida a lo largo de sus estudios universitarios.

1. El programador obtiene completa confianza y seguridad al emplear solamente código fuente para realizar las operaciones transaccionales hacia la base de datos.
Totalmente de acuerdo
De acuerdo
Ni en acuerdo, ni en desacuerdo
En desacuerdo
Totalmente en desacuerdo

2. Cualificar los métodos o funciones para la selección, inserción, actualización y eliminación de datos, provocan que el desarrollo de la aplicación sea monótona, repetitiva y a la vez compleja.
Totalmente de acuerdo
De acuerdo
Ni en acuerdo, ni en desacuerdo
En desacuerdo
Totalmente en desacuerdo

3. Los componentes gráficos que ayudan al programador a desarrollar aplicaciones, permiten que la construcción del software sea más sencilla y rápida.
Totalmente de acuerdo
De acuerdo
Ni en acuerdo, ni en desacuerdo
En desacuerdo
Totalmente en desacuerdo

4. El desarrollo de los métodos de selección, inserción, actualización y eliminación de datos sería más fácil y rápido si se realizara con la ayuda de componentes de asistencia gráfica.
Totalmente de acuerdo
De acuerdo
Ni en acuerdo, ni en desacuerdo
En desacuerdo
Totalmente en desacuerdo

5. El procedimiento para realizar las operaciones con la base de datos sería más fácil si pudiéramos visualizar desde el IDE (entorno de desarrollo integrado) el diagrama del modelo relacional de la base de datos.
Totalmente de acuerdo
De acuerdo
Ni en acuerdo, ni en desacuerdo
En desacuerdo
Totalmente en desacuerdo

6. Emplear componentes gráficos especializados en las operaciones transaccionales con base de datos influyen total comodidad y facilidad al desarrollar aplicaciones.
Totalmente de acuerdo
De acuerdo
Ni en acuerdo, ni en desacuerdo
En desacuerdo
Totalmente en desacuerdo

7. La implementación de código fuente y componentes de asistencia gráfica debe ser equilibrada al momento de desarrollar aplicaciones.
Totalmente de acuerdo
De acuerdo
Ni en acuerdo, ni en desacuerdo
En desacuerdo
Totalmente en desacuerdo



UNIVERSIDAD TÉCNICA ESTATAL DE QUEVEDO

Encuesta para evaluar la satisfacción de los programadores de software, en cuanto al procedimiento de acceso a datos cuando desarrollan aplicaciones.

Dirigida a: Los estudiantes de la Carrera de Sistemas de la Facultad de Ciencias de la Ingeniería de la UTEQ.

Responda: Según las siguientes declaraciones seleccione una opción, asignando total honestidad en base a la experiencia obtenida a lo largo de sus estudios universitarios.

1. El programador obtiene completa confianza y seguridad al emplear solamente código fuente para realizar las operaciones transaccionales hacia la base de datos.
Totalmente de acuerdo
De acuerdo
Ni en acuerdo, ni en desacuerdo
En desacuerdo
Totalmente en desacuerdo

2. Cualificar los métodos o funciones para la selección, inserción, actualización y eliminación de datos, provocan que el desarrollo de la aplicación sea monótona, repetitiva y a la vez compleja.
Totalmente de acuerdo
De acuerdo
Ni en acuerdo, ni en desacuerdo
En desacuerdo
Totalmente en desacuerdo

3. Los componentes gráficos que ayudan al programador a desarrollar aplicaciones, permiten que la construcción del software sea más sencilla y rápida.
Totalmente de acuerdo
De acuerdo
Ni en acuerdo, ni en desacuerdo
En desacuerdo
Totalmente en desacuerdo

4. El desarrollo de los métodos de selección, inserción, actualización y eliminación de datos sería más fácil y rápido si se realizara con la ayuda de componentes de asistencia gráfica.
Totalmente de acuerdo
De acuerdo
Ni en acuerdo, ni en desacuerdo
En desacuerdo
Totalmente en desacuerdo

5. El procedimiento para realizar las operaciones con la base de datos sería más fácil si pudiéramos visualizar desde el IDE (entorno de desarrollo integrado) el diagrama del modelo relacional de la base de datos.
Totalmente de acuerdo
De acuerdo
Ni en acuerdo, ni en desacuerdo
En desacuerdo
Totalmente en desacuerdo

6. Emplear componentes gráficos especializados en las operaciones transaccionales con base de datos influyen total comodidad y facilidad al desarrollar aplicaciones.
Totalmente de acuerdo
De acuerdo
Ni en acuerdo, ni en desacuerdo
En desacuerdo
Totalmente en desacuerdo

7. La implementación de código fuente y componentes de asistencia gráfica debe ser equilibrada al momento de desarrollar aplicaciones.
Totalmente de acuerdo
De acuerdo
Ni en acuerdo, ni en desacuerdo
En desacuerdo
Totalmente en desacuerdo



UNIVERSIDAD TÉCNICA ESTATAL DE QUEVEDO

Encuesta para evaluar la satisfacción de los programadores de software, en cuanto al procedimiento de acceso a datos cuando desarrollan aplicaciones.

Dirigida a: Los estudiantes de la Carrera de Sistemas de la Facultad de Ciencias de la Ingeniería de la UTEQ.

Responda: Según las siguientes declaraciones seleccione una opción, asumiendo total honestidad en base a la experiencia obtenida a lo largo de sus estudios universitarios.

1. El programador obtiene completa confianza y seguridad al emplear solamente código fuente para realizar las operaciones transaccionales hacia la base de datos.

- Totalmente de acuerdo
De acuerdo
Ni en acuerdo, ni en desacuerdo
En desacuerdo
Totalmente en desacuerdo

2. Clasificar los métodos o funciones para la selección, inserción, actualización y eliminación de datos, previene que el desarrollo de la aplicación sea monótona, repetitiva y a la vez compleja.

- Totalmente de acuerdo
De acuerdo
Ni en acuerdo, ni en desacuerdo
En desacuerdo
Totalmente en desacuerdo

3. Los componentes gráficos que ayudan al programador a desarrollar aplicaciones, permiten que la construcción del software sea más sencilla y rápida.

- Totalmente de acuerdo
De acuerdo
Ni en acuerdo, ni en desacuerdo
En desacuerdo
Totalmente en desacuerdo

4. El desarrollo de los métodos de selección, inserción, actualización y eliminación de datos sería más fácil y rápido si se realizara con la ayuda de componentes de asistencia gráfica.

- Totalmente de acuerdo
De acuerdo
Ni en acuerdo, ni en desacuerdo
En desacuerdo
Totalmente en desacuerdo

5. El procedimiento para realizar las operaciones con la base de datos sería más fácil si pudiéramos visualizar desde el IDE (entorno de desarrollo integrado) el diagrama del modelo relacional de la base de datos.

- Totalmente de acuerdo
De acuerdo
Ni en acuerdo, ni en desacuerdo
En desacuerdo
Totalmente en desacuerdo

6. Emplear componentes gráficos especializados en las operaciones transaccionales con base de datos influyen total confianza y facilidad al desarrollar aplicaciones.

- Totalmente de acuerdo
De acuerdo
Ni en acuerdo, ni en desacuerdo
En desacuerdo
Totalmente en desacuerdo

7. La implementación de código fuente y componentes de asistencia gráfica debe ser equilibrada al momento de desarrollar aplicaciones.

- Totalmente de acuerdo
De acuerdo
Ni en acuerdo, ni en desacuerdo
En desacuerdo
Totalmente en desacuerdo



UNIVERSIDAD TÉCNICA ESTATAL DE QUEVEDO

Encuesta para evaluar la satisfacción de los programadores de software, en cuanto al procedimiento de acceso a datos cuando desarrollan aplicaciones.

Dirigida a: Los estudiantes de la Carrera de Sistemas de la Facultad de Ciencias de la Ingeniería de la UTEQ.

Responda: Según las siguientes declaraciones seleccione una opción, asumiendo total honestidad en base a la experiencia obtenida a lo largo de sus estudios universitarios.

1. El programador obtiene completa confianza y seguridad al emplear solamente código fuente para realizar las operaciones transaccionales hacia la base de datos.

- Totalmente de acuerdo
De acuerdo
Ni en acuerdo, ni en desacuerdo
En desacuerdo
Totalmente en desacuerdo

2. Clasificar los métodos o funciones para la selección, inserción, actualización y eliminación de datos, previene que el desarrollo de la aplicación sea monótona, repetitiva y a la vez compleja.

- Totalmente de acuerdo
De acuerdo
Ni en acuerdo, ni en desacuerdo
En desacuerdo
Totalmente en desacuerdo

3. Los componentes gráficos que ayudan al programador a desarrollar aplicaciones, permiten que la construcción del software sea más sencilla y rápida.

- Totalmente de acuerdo
De acuerdo
Ni en acuerdo, ni en desacuerdo
En desacuerdo
Totalmente en desacuerdo

4. El desarrollo de los métodos de selección, inserción, actualización y eliminación de datos sería más fácil y rápido si se realizara con la ayuda de componentes de asistencia gráfica.

- Totalmente de acuerdo
De acuerdo
Ni en acuerdo, ni en desacuerdo
En desacuerdo
Totalmente en desacuerdo

5. El procedimiento para realizar las operaciones con la base de datos sería más fácil si pudiéramos visualizar desde el IDE (entorno de desarrollo integrado) el diagrama del modelo relacional de la base de datos.

- Totalmente de acuerdo
De acuerdo
Ni en acuerdo, ni en desacuerdo
En desacuerdo
Totalmente en desacuerdo

6. Emplear componentes gráficos especializados en las operaciones transaccionales con base de datos influyen total confianza y facilidad al desarrollar aplicaciones.

- Totalmente de acuerdo
De acuerdo
Ni en acuerdo, ni en desacuerdo
En desacuerdo
Totalmente en desacuerdo

7. La implementación de código fuente y componentes de asistencia gráfica debe ser equilibrada al momento de desarrollar aplicaciones.

- Totalmente de acuerdo
De acuerdo
Ni en acuerdo, ni en desacuerdo
En desacuerdo
Totalmente en desacuerdo



UNIVERSIDAD TÉCNICA ESTATAL DE QUEVEDO

Encuesta para evaluar la satisfacción de los programadores de software, en cuanto al procedimiento de acceso a datos cuando desarrollan aplicaciones.

Dirigida a: Los estudiantes de la Carrera de Sistemas de la Facultad de Ciencias de la Ingeniería de la UTEQ.

Responda: Según las siguientes declaraciones seleccione una opción, asumiendo total honestidad en base a la experiencia obtenida a lo largo de sus estudios universitarios.

1. El programador obtiene completa confianza y seguridad al emplear solamente código fuente para realizar las operaciones transaccionales hacia la base de datos.

- Totalmente de acuerdo
De acuerdo
Ni en acuerdo, ni en desacuerdo
En desacuerdo
Totalmente en desacuerdo

2. Clasificar los métodos o funciones para la selección, inserción, actualización y eliminación de datos, previene que el desarrollo de la aplicación sea monótona, repetitiva y a la vez compleja.

- Totalmente de acuerdo
De acuerdo
Ni en acuerdo, ni en desacuerdo
En desacuerdo
Totalmente en desacuerdo

3. Los componentes gráficos que ayudan al programador a desarrollar aplicaciones, permiten que la construcción del software sea más sencilla y rápida.

- Totalmente de acuerdo
De acuerdo
Ni en acuerdo, ni en desacuerdo
En desacuerdo
Totalmente en desacuerdo

4. El desarrollo de los métodos de selección, inserción, actualización y eliminación de datos sería más fácil y rápido si se realizara con la ayuda de componentes de asistencia gráfica.

- Totalmente de acuerdo
De acuerdo
Ni en acuerdo, ni en desacuerdo
En desacuerdo
Totalmente en desacuerdo

5. El procedimiento para realizar las operaciones con la base de datos sería más fácil si pudiéramos visualizar desde el IDE (entorno de desarrollo integrado) el diagrama del modelo relacional de la base de datos.

- Totalmente de acuerdo
De acuerdo
Ni en acuerdo, ni en desacuerdo
En desacuerdo
Totalmente en desacuerdo

6. Emplear componentes gráficos especializados en las operaciones transaccionales con base de datos influyen total confianza y facilidad al desarrollar aplicaciones.

- Totalmente de acuerdo
De acuerdo
Ni en acuerdo, ni en desacuerdo
En desacuerdo
Totalmente en desacuerdo

7. La implementación de código fuente y componentes de asistencia gráfica debe ser equilibrada al momento de desarrollar aplicaciones.

- Totalmente de acuerdo
De acuerdo
Ni en acuerdo, ni en desacuerdo
En desacuerdo
Totalmente en desacuerdo



UNIVERSIDAD TÉCNICA ESTATAL DE QUEVEDO

Encuesta para evaluar la satisfacción de los programadores de software en cuanto al procedimiento de acceso a datos cuando desarrollan aplicaciones.

Dirigida a: Los estudiantes de la Carrera de Sistemas de la Facultad de Ciencias de la Ingeniería de la UTEQ.

Requisitos: Según las siguientes declaraciones seleccione una opción, indicando total honestidad en base a la experiencia obtenida a lo largo de sus estudios universitarios.

- 1.- El programador obtiene completa confianza y seguridad al emplear solamente código fuente para realizar las operaciones transaccionales hacia la base de datos.
Totalmente de acuerdo
De acuerdo
Ni en acuerdo, ni en desacuerdo
En desacuerdo
Totalmente en desacuerdo

- 2.- Clasificar los métodos o funciones para la selección, inserción, actualización y eliminación de datos, previos que el desarrollo de la aplicación sea monitorea, repetitiva y a la vez compleja.
Totalmente de acuerdo
De acuerdo
Ni en acuerdo, ni en desacuerdo
En desacuerdo
Totalmente en desacuerdo

- 3.- Los componentes gráficos que ayudan al programador a desarrollar aplicaciones, permiten que la construcción del software sea más sencilla y rápida.
Totalmente de acuerdo
De acuerdo
Ni en acuerdo, ni en desacuerdo
En desacuerdo
Totalmente en desacuerdo

- 4.- El desarrollo de los métodos de selección, inserción, actualización y eliminación de datos sea más fácil y rápido si se realiza con la ayuda de componentes de asistencia gráfica.
Totalmente de acuerdo
De acuerdo
Ni en acuerdo, ni en desacuerdo
En desacuerdo
Totalmente en desacuerdo

- 5.- El procedimiento para realizar las operaciones con la base de datos sería más fácil si pudiéramos visualizar datos al ICE (sistema de desarrollo integrado) el diagrama del modelo relacional de la base de datos.
Totalmente de acuerdo
De acuerdo
Ni en acuerdo, ni en desacuerdo
En desacuerdo
Totalmente en desacuerdo

- 6.- Emplear componentes gráficos especializados en las operaciones transaccionales con base de datos influyen total confianza y facilidad al desarrollar aplicaciones.
Totalmente de acuerdo
De acuerdo
Ni en acuerdo, ni en desacuerdo
En desacuerdo
Totalmente en desacuerdo

- 7.- La implementación de código fuente y componentes de asistencia gráfica debe ser equivalentes al momento de desarrollar aplicaciones.
Totalmente de acuerdo
De acuerdo
Ni en acuerdo, ni en desacuerdo
En desacuerdo
Totalmente en desacuerdo



UNIVERSIDAD TÉCNICA ESTATAL DE QUEVEDO

Encuesta para evaluar la satisfacción de los programadores de software en cuanto al procedimiento de acceso a datos cuando desarrollan aplicaciones.

Dirigida a: Los estudiantes de la Carrera de Sistemas de la Facultad de Ciencias de la Ingeniería de la UTEQ.

Requisitos: Según las siguientes declaraciones seleccione una opción, indicando total honestidad en base a la experiencia obtenida a lo largo de sus estudios universitarios.

- 1.- El programador obtiene completa confianza y seguridad al emplear solamente código fuente para realizar las operaciones transaccionales hacia la base de datos.
Totalmente de acuerdo
De acuerdo
Ni en acuerdo, ni en desacuerdo
En desacuerdo
Totalmente en desacuerdo

- 2.- Clasificar los métodos o funciones para la selección, inserción, actualización y eliminación de datos, previos que el desarrollo de la aplicación sea monitorea, repetitiva y a la vez compleja.
Totalmente de acuerdo
De acuerdo
Ni en acuerdo, ni en desacuerdo
En desacuerdo
Totalmente en desacuerdo

- 3.- Los componentes gráficos que ayudan al programador a desarrollar aplicaciones, permiten que la construcción del software sea más sencilla y rápida.
Totalmente de acuerdo
De acuerdo
Ni en acuerdo, ni en desacuerdo
En desacuerdo
Totalmente en desacuerdo

- 4.- El desarrollo de los métodos de selección, inserción, actualización y eliminación de datos sea más fácil y rápido si se realiza con la ayuda de componentes de asistencia gráfica.
Totalmente de acuerdo
De acuerdo
Ni en acuerdo, ni en desacuerdo
En desacuerdo
Totalmente en desacuerdo

- 5.- El procedimiento para realizar las operaciones con la base de datos sería más fácil si pudiéramos visualizar datos al ICE (sistema de desarrollo integrado) el diagrama del modelo relacional de la base de datos.
Totalmente de acuerdo
De acuerdo
Ni en acuerdo, ni en desacuerdo
En desacuerdo
Totalmente en desacuerdo

- 6.- Emplear componentes gráficos especializados en las operaciones transaccionales con base de datos influyen total confianza y facilidad al desarrollar aplicaciones.
Totalmente de acuerdo
De acuerdo
Ni en acuerdo, ni en desacuerdo
En desacuerdo
Totalmente en desacuerdo

- 7.- La implementación de código fuente y componentes de asistencia gráfica debe ser equivalentes al momento de desarrollar aplicaciones.
Totalmente de acuerdo
De acuerdo
Ni en acuerdo, ni en desacuerdo
En desacuerdo
Totalmente en desacuerdo



UNIVERSIDAD TÉCNICA ESTATAL DE QUEVEDO

Encuesta para evaluar la satisfacción de los programadores de software en cuanto al procedimiento de acceso a datos cuando desarrollan aplicaciones.

Dirigida a: Los estudiantes de la Carrera de Sistemas de la Facultad de Ciencias de la Ingeniería de la UTEQ.

Requisitos: Según las siguientes declaraciones seleccione una opción, indicando total honestidad en base a la experiencia obtenida a lo largo de sus estudios universitarios.

- 1.- El programador obtiene completa confianza y seguridad al emplear solamente código fuente para realizar las operaciones transaccionales hacia la base de datos.
Totalmente de acuerdo
De acuerdo
Ni en acuerdo, ni en desacuerdo
En desacuerdo
Totalmente en desacuerdo

- 2.- Clasificar los métodos o funciones para la selección, inserción, actualización y eliminación de datos, previos que el desarrollo de la aplicación sea monitorea, repetitiva y a la vez compleja.
Totalmente de acuerdo
De acuerdo
Ni en acuerdo, ni en desacuerdo
En desacuerdo
Totalmente en desacuerdo

- 3.- Los componentes gráficos que ayudan al programador a desarrollar aplicaciones, permiten que la construcción del software sea más sencilla y rápida.
Totalmente de acuerdo
De acuerdo
Ni en acuerdo, ni en desacuerdo
En desacuerdo
Totalmente en desacuerdo

- 4.- El desarrollo de los métodos de selección, inserción, actualización y eliminación de datos sea más fácil y rápido si se realiza con la ayuda de componentes de asistencia gráfica.
Totalmente de acuerdo
De acuerdo
Ni en acuerdo, ni en desacuerdo
En desacuerdo
Totalmente en desacuerdo

- 5.- El procedimiento para realizar las operaciones con la base de datos sería más fácil si pudiéramos visualizar datos al ICE (sistema de desarrollo integrado) el diagrama del modelo relacional de la base de datos.
Totalmente de acuerdo
De acuerdo
Ni en acuerdo, ni en desacuerdo
En desacuerdo
Totalmente en desacuerdo

- 6.- Emplear componentes gráficos especializados en las operaciones transaccionales con base de datos influyen total confianza y facilidad al desarrollar aplicaciones.
Totalmente de acuerdo
De acuerdo
Ni en acuerdo, ni en desacuerdo
En desacuerdo
Totalmente en desacuerdo

- 7.- La implementación de código fuente y componentes de asistencia gráfica debe ser equivalentes al momento de desarrollar aplicaciones.
Totalmente de acuerdo
De acuerdo
Ni en acuerdo, ni en desacuerdo
En desacuerdo
Totalmente en desacuerdo



UNIVERSIDAD TÉCNICA ESTATAL DE QUEVEDO

Encuesta para evaluar la satisfacción de los programadores de software, en cuanto al procedimiento de acceso a datos cuando desarrollan aplicaciones.

Dirigida a: Los estudiantes de la Carrera de Sistemas de la Facultad de Ciencias de la Ingeniería de la UTEQ.

Responda: Según las siguientes declaraciones seleccione una opción, asignando total honestidad en base a la experiencia obtenida a lo largo de sus estudios universitarios.

- 1. El programador obtiene completa confianza y seguridad al emplear solamente código fuente para realizar las operaciones transaccionales hacia la base de datos.
2. Codificar los métodos o funciones para la selección, inserción, actualización y eliminación de datos, previene que el desarrollo de la aplicación sea monótona, repetitiva y a la vez compleja.
3. Los componentes gráficos que ayudan al programador a desarrollar aplicaciones, permiten que la construcción del software sea más sencilla y rápida.
4. El desarrollo de los métodos de selección, inserción, actualización y eliminación de datos sea más fácil y rápido si se realizan con la ayuda de componentes de asistencia gráfica.

- 5. El procedimiento para realizar las operaciones con la base de datos sería más fácil si pudiéramos visualizar desde el DDE (entorno de desarrollo integrado) el diagrama del modelo relacional de la base de datos.
6. Emplear componentes gráficos especializados en las operaciones transaccionales con base de datos influyen total o parcialmente en la comodidad y facilidad al desarrollar aplicaciones.

- 7. La implementación de código fuente y componentes de asistencia gráfica debe ser equilibrada al momento de desarrollar aplicaciones.



UNIVERSIDAD TÉCNICA ESTATAL DE QUEVEDO

Encuesta para evaluar la satisfacción de los programadores de software, en cuanto al procedimiento de acceso a datos cuando desarrollan aplicaciones.

Dirigida a: Los estudiantes de la Carrera de Sistemas de la Facultad de Ciencias de la Ingeniería de la UTEQ.

Responda: Según las siguientes declaraciones seleccione una opción, asignando total honestidad en base a la experiencia obtenida a lo largo de sus estudios universitarios.

- 1. El programador obtiene completa confianza y seguridad al emplear solamente código fuente para realizar las operaciones transaccionales hacia la base de datos.
2. Codificar los métodos o funciones para la selección, inserción, actualización y eliminación de datos, previene que el desarrollo de la aplicación sea monótona, repetitiva y a la vez compleja.
3. Los componentes gráficos que ayudan al programador a desarrollar aplicaciones, permiten que la construcción del software sea más sencilla y rápida.
4. El desarrollo de los métodos de selección, inserción, actualización y eliminación de datos sea más fácil y rápido si se realizan con la ayuda de componentes de asistencia gráfica.

- 5. El procedimiento para realizar las operaciones con la base de datos sería más fácil si pudiéramos visualizar desde el DDE (entorno de desarrollo integrado) el diagrama del modelo relacional de la base de datos.
6. Emplear componentes gráficos especializados en las operaciones transaccionales con base de datos influyen total o parcialmente en la comodidad y facilidad al desarrollar aplicaciones.

- 7. La implementación de código fuente y componentes de asistencia gráfica debe ser equilibrada al momento de desarrollar aplicaciones.



UNIVERSIDAD TÉCNICA ESTATAL DE QUEVEDO

Encuesta para evaluar la satisfacción de los programadores de software, en cuanto al procedimiento de acceso a datos cuando desarrollan aplicaciones.

Dirigida a: Los estudiantes de la Carrera de Sistemas de la Facultad de Ciencias de la Ingeniería de la UTEQ.

Responda: Según las siguientes declaraciones seleccione una opción, asignando total honestidad en base a la experiencia obtenida a lo largo de sus estudios universitarios.

- 1. El programador obtiene completa confianza y seguridad al emplear solamente código fuente para realizar las operaciones transaccionales hacia la base de datos.
2. Codificar los métodos o funciones para la selección, inserción, actualización y eliminación de datos, previene que el desarrollo de la aplicación sea monótona, repetitiva y a la vez compleja.
3. Los componentes gráficos que ayudan al programador a desarrollar aplicaciones, permiten que la construcción del software sea más sencilla y rápida.
4. El desarrollo de los métodos de selección, inserción, actualización y eliminación de datos sea más fácil y rápido si se realizan con la ayuda de componentes de asistencia gráfica.

- 5. El procedimiento para realizar las operaciones con la base de datos sería más fácil si pudiéramos visualizar desde el DDE (entorno de desarrollo integrado) el diagrama del modelo relacional de la base de datos.
6. Emplear componentes gráficos especializados en las operaciones transaccionales con base de datos influyen total o parcialmente en la comodidad y facilidad al desarrollar aplicaciones.

- 7. La implementación de código fuente y componentes de asistencia gráfica debe ser equilibrada al momento de desarrollar aplicaciones.



UNIVERSIDAD TÉCNICA ESTATAL DE QUEVEDO

Encuesta para evaluar la satisfacción de los programadores de software, en cuanto al procedimiento de acceso a datos cuando desarrollan aplicaciones.

Dirigida a: Los estudiantes de la Carrera de Sistemas de la Facultad de Ciencias de la Ingeniería de la UTEQ.

Respecto: Según las siguientes declaraciones seleccione una opción, acimienta total honestidad en base a la experiencia obtenida a lo largo de sus estudios universitarios.

1. El programador define completa confianza y seguridad al escribir solamente código fuente para realizar las operaciones transaccionales hacia la base de datos.
Totalmente de acuerdo
De acuerdo
Ni en acuerdo, ni en desacuerdo
En desacuerdo
Totalmente en desacuerdo

2. Codificar los métodos o funciones para la selección, inserción, actualización y eliminación de datos, provocan que el desarrollo de la aplicación sea monótona, repetitiva y a la vez compleja.
Totalmente de acuerdo
De acuerdo
Ni en acuerdo, ni en desacuerdo
En desacuerdo
Totalmente en desacuerdo

3. Los componentes gráficos que ayudan al programador a desarrollar aplicaciones, permiten que la construcción del software sea más sencilla y rápida.
Totalmente de acuerdo
De acuerdo
Ni en acuerdo, ni en desacuerdo
En desacuerdo
Totalmente en desacuerdo

4. El desarrollo de los métodos de selección, inserción, actualización y eliminación de datos sería más fácil y rápida si se realizara con la ayuda de componentes de asistencia gráfica.
Totalmente de acuerdo
De acuerdo
Ni en acuerdo, ni en desacuerdo
En desacuerdo
Totalmente en desacuerdo

5. El procedimiento para realizar las operaciones con la base de datos sería más fácil si pudieramos visualizar desde el IDE (entorno de desarrollo integrado) el diagrama del modelo relacional de la base de datos.
Totalmente de acuerdo
De acuerdo
Ni en acuerdo, ni en desacuerdo
En desacuerdo
Totalmente en desacuerdo

6. Emplear componentes gráficos especializados en las operaciones transaccionales con base de datos influyen total comodidad y facilidad al desarrollar aplicaciones.
Totalmente de acuerdo
De acuerdo
Ni en acuerdo, ni en desacuerdo
En desacuerdo
Totalmente en desacuerdo

7. La implementación de código fuente y componentes de asistencia gráfica debe ser equilibrada al momento de desarrollar aplicaciones.
Totalmente de acuerdo
De acuerdo
Ni en acuerdo, ni en desacuerdo
En desacuerdo
Totalmente en desacuerdo



UNIVERSIDAD TÉCNICA ESTATAL DE QUEVEDO

Encuesta para evaluar la satisfacción de los programadores de software, en cuanto al procedimiento de acceso a datos cuando desarrollan aplicaciones.

Dirigida a: Los estudiantes de la Carrera de Sistemas de la Facultad de Ciencias de la Ingeniería de la UTEQ.

Respecto: Según las siguientes declaraciones seleccione una opción, acimienta total honestidad en base a la experiencia obtenida a lo largo de sus estudios universitarios.

1. El programador define completa confianza y seguridad al escribir solamente código fuente para realizar las operaciones transaccionales hacia la base de datos.
Totalmente de acuerdo
De acuerdo
Ni en acuerdo, ni en desacuerdo
En desacuerdo
Totalmente en desacuerdo

2. Codificar los métodos o funciones para la selección, inserción, actualización y eliminación de datos, provocan que el desarrollo de la aplicación sea monótona, repetitiva y a la vez compleja.
Totalmente de acuerdo
De acuerdo
Ni en acuerdo, ni en desacuerdo
En desacuerdo
Totalmente en desacuerdo

3. Los componentes gráficos que ayudan al programador a desarrollar aplicaciones, permiten que la construcción del software sea más sencilla y rápida.
Totalmente de acuerdo
De acuerdo
Ni en acuerdo, ni en desacuerdo
En desacuerdo
Totalmente en desacuerdo

4. El desarrollo de los métodos de selección, inserción, actualización y eliminación de datos sería más fácil y rápida si se realizara con la ayuda de componentes de asistencia gráfica.
Totalmente de acuerdo
De acuerdo
Ni en acuerdo, ni en desacuerdo
En desacuerdo
Totalmente en desacuerdo

5. El procedimiento para realizar las operaciones con la base de datos sería más fácil si pudieramos visualizar desde el IDE (entorno de desarrollo integrado) el diagrama del modelo relacional de la base de datos.
Totalmente de acuerdo
De acuerdo
Ni en acuerdo, ni en desacuerdo
En desacuerdo
Totalmente en desacuerdo

6. Emplear componentes gráficos especializados en las operaciones transaccionales con base de datos influyen total comodidad y facilidad al desarrollar aplicaciones.
Totalmente de acuerdo
De acuerdo
Ni en acuerdo, ni en desacuerdo
En desacuerdo
Totalmente en desacuerdo

7. La implementación de código fuente y componentes de asistencia gráfica debe ser equilibrada al momento de desarrollar aplicaciones.
Totalmente de acuerdo
De acuerdo
Ni en acuerdo, ni en desacuerdo
En desacuerdo
Totalmente en desacuerdo



UNIVERSIDAD TÉCNICA ESTATAL DE QUEVEDO

Encuesta para evaluar la satisfacción de los programadores de software, en cuanto al procedimiento de acceso a datos cuando desarrollan aplicaciones.

Dirigida a: Los estudiantes de la Carrera de Sistemas de la Facultad de Ciencias de la Ingeniería de la UTEQ.

Respecto: Según las siguientes declaraciones seleccione una opción, acimienta total honestidad en base a la experiencia obtenida a lo largo de sus estudios universitarios.

1. El programador define completa confianza y seguridad al escribir solamente código fuente para realizar las operaciones transaccionales hacia la base de datos.
Totalmente de acuerdo
De acuerdo
Ni en acuerdo, ni en desacuerdo
En desacuerdo
Totalmente en desacuerdo

2. Codificar los métodos o funciones para la selección, inserción, actualización y eliminación de datos, provocan que el desarrollo de la aplicación sea monótona, repetitiva y a la vez compleja.
Totalmente de acuerdo
De acuerdo
Ni en acuerdo, ni en desacuerdo
En desacuerdo
Totalmente en desacuerdo

3. Los componentes gráficos que ayudan al programador a desarrollar aplicaciones, permiten que la construcción del software sea más sencilla y rápida.
Totalmente de acuerdo
De acuerdo
Ni en acuerdo, ni en desacuerdo
En desacuerdo
Totalmente en desacuerdo

4. El desarrollo de los métodos de selección, inserción, actualización y eliminación de datos sería más fácil y rápida si se realizara con la ayuda de componentes de asistencia gráfica.
Totalmente de acuerdo
De acuerdo
Ni en acuerdo, ni en desacuerdo
En desacuerdo
Totalmente en desacuerdo

5. El procedimiento para realizar las operaciones con la base de datos sería más fácil si pudieramos visualizar desde el IDE (entorno de desarrollo integrado) el diagrama del modelo relacional de la base de datos.
Totalmente de acuerdo
De acuerdo
Ni en acuerdo, ni en desacuerdo
En desacuerdo
Totalmente en desacuerdo

6. Emplear componentes gráficos especializados en las operaciones transaccionales con base de datos influyen total comodidad y facilidad al desarrollar aplicaciones.
Totalmente de acuerdo
De acuerdo
Ni en acuerdo, ni en desacuerdo
En desacuerdo
Totalmente en desacuerdo

7. La implementación de código fuente y componentes de asistencia gráfica debe ser equilibrada al momento de desarrollar aplicaciones.
Totalmente de acuerdo
De acuerdo
Ni en acuerdo, ni en desacuerdo
En desacuerdo
Totalmente en desacuerdo



UNIVERSIDAD TÉCNICA ESTATAL DE QUEVEDO

Encuesta para evaluar la satisfacción de los programadores de software, en cuanto al procedimiento de acceso a datos cuando desarrollan aplicaciones.

Dirigida a: Los estudiantes de la Carrera de Sistemas de la Facultad de Ciencias de la Ingeniería de la UTEQ.

Requisito: Según las siguientes declaraciones seleccione una opción, aumentando total honestidad en base a la experiencia obtenida a lo largo de sus estudios universitarios.

- 1. El programador obtiene completa confianza y seguridad al emplear solamente código fuente para realizar las operaciones transaccionales hacia la base de datos.
Totalmente de acuerdo
De acuerdo
Ni en acuerdo, ni en desacuerdo
En desacuerdo
Totalmente en desacuerdo

- 2. Codificar los métodos o funciones para la selección, inserción, actualización y eliminación de datos, provocan que el desarrollo de la aplicación sea monótona, repetitiva y a la vez compleja.
Totalmente de acuerdo
De acuerdo
Ni en acuerdo, ni en desacuerdo
En desacuerdo
Totalmente en desacuerdo

- 3. Los componentes gráficos que ayudan al programador a desarrollar aplicaciones, permiten que la construcción del software sea más sencilla y rápida.
Totalmente de acuerdo
De acuerdo
Ni en acuerdo, ni en desacuerdo
En desacuerdo
Totalmente en desacuerdo

- 4. El desarrollo de los métodos de selección, inserción, actualización y eliminación de datos sea más fácil y rápido si se realizan con la ayuda de componentes de asistencia gráfica.
Totalmente de acuerdo
De acuerdo
Ni en acuerdo, ni en desacuerdo
En desacuerdo
Totalmente en desacuerdo

- 5. El procedimiento para realizar las operaciones con la base de datos sería más fácil si pudiéramos visualizar desde el ICE (entorno de desarrollo integrado) el diagrama del modelo relacional de la base de datos.
Totalmente de acuerdo
De acuerdo
Ni en acuerdo, ni en desacuerdo
En desacuerdo
Totalmente en desacuerdo

- 6. Emplear componentes gráficos especializados en las operaciones transaccionales con base de datos incrementa total comodidad y facilidad al desarrollar aplicaciones.
Totalmente de acuerdo
De acuerdo
Ni en acuerdo, ni en desacuerdo
En desacuerdo
Totalmente en desacuerdo

- 7. La implementación de código fuente y componentes de asistencia gráfica debe ser equilibrada al momento de desarrollar aplicaciones.
Totalmente de acuerdo
De acuerdo
Ni en acuerdo, ni en desacuerdo
En desacuerdo
Totalmente en desacuerdo



UNIVERSIDAD TÉCNICA ESTATAL DE QUEVEDO

Encuesta para evaluar la satisfacción de los programadores de software, en cuanto al procedimiento de acceso a datos cuando desarrollan aplicaciones.

Dirigida a: Los estudiantes de la Carrera de Sistemas de la Facultad de Ciencias de la Ingeniería de la UTEQ.

Requisito: Según las siguientes declaraciones seleccione una opción, aumentando total honestidad en base a la experiencia obtenida a lo largo de sus estudios universitarios.

- 1. El programador obtiene completa confianza y seguridad al emplear solamente código fuente para realizar las operaciones transaccionales hacia la base de datos.
Totalmente de acuerdo
De acuerdo
Ni en acuerdo, ni en desacuerdo
En desacuerdo
Totalmente en desacuerdo

- 2. Codificar los métodos o funciones para la selección, inserción, actualización y eliminación de datos, provocan que el desarrollo de la aplicación sea monótona, repetitiva y a la vez compleja.
Totalmente de acuerdo
De acuerdo
Ni en acuerdo, ni en desacuerdo
En desacuerdo
Totalmente en desacuerdo

- 3. Los componentes gráficos que ayudan al programador a desarrollar aplicaciones, permiten que la construcción del software sea más sencilla y rápida.
Totalmente de acuerdo
De acuerdo
Ni en acuerdo, ni en desacuerdo
En desacuerdo
Totalmente en desacuerdo

- 4. El desarrollo de los métodos de selección, inserción, actualización y eliminación de datos sea más fácil y rápido si se realizan con la ayuda de componentes de asistencia gráfica.
Totalmente de acuerdo
De acuerdo
Ni en acuerdo, ni en desacuerdo
En desacuerdo
Totalmente en desacuerdo

- 5. El procedimiento para realizar las operaciones con la base de datos sería más fácil si pudiéramos visualizar desde el ICE (entorno de desarrollo integrado) el diagrama del modelo relacional de la base de datos.
Totalmente de acuerdo
De acuerdo
Ni en acuerdo, ni en desacuerdo
En desacuerdo
Totalmente en desacuerdo

- 6. Emplear componentes gráficos especializados en las operaciones transaccionales con base de datos incrementa total comodidad y facilidad al desarrollar aplicaciones.
Totalmente de acuerdo
De acuerdo
Ni en acuerdo, ni en desacuerdo
En desacuerdo
Totalmente en desacuerdo

- 7. La implementación de código fuente y componentes de asistencia gráfica debe ser equilibrada al momento de desarrollar aplicaciones.
Totalmente de acuerdo
De acuerdo
Ni en acuerdo, ni en desacuerdo
En desacuerdo
Totalmente en desacuerdo



UNIVERSIDAD TÉCNICA ESTATAL DE QUEVEDO

Encuesta para evaluar la satisfacción de los programadores de software, en cuanto al procedimiento de acceso a datos cuando desarrollan aplicaciones.

Dirigida a: Los estudiantes de la Carrera de Sistemas de la Facultad de Ciencias de la Ingeniería de la UTEQ.

Requisito: Según las siguientes declaraciones seleccione una opción, aumentando total honestidad en base a la experiencia obtenida a lo largo de sus estudios universitarios.

- 1. El programador obtiene completa confianza y seguridad al emplear solamente código fuente para realizar las operaciones transaccionales hacia la base de datos.
Totalmente de acuerdo
De acuerdo
Ni en acuerdo, ni en desacuerdo
En desacuerdo
Totalmente en desacuerdo

- 2. Codificar los métodos o funciones para la selección, inserción, actualización y eliminación de datos, provocan que el desarrollo de la aplicación sea monótona, repetitiva y a la vez compleja.
Totalmente de acuerdo
De acuerdo
Ni en acuerdo, ni en desacuerdo
En desacuerdo
Totalmente en desacuerdo

- 3. Los componentes gráficos que ayudan al programador a desarrollar aplicaciones, permiten que la construcción del software sea más sencilla y rápida.
Totalmente de acuerdo
De acuerdo
Ni en acuerdo, ni en desacuerdo
En desacuerdo
Totalmente en desacuerdo

- 4. El desarrollo de los métodos de selección, inserción, actualización y eliminación de datos sea más fácil y rápido si se realizan con la ayuda de componentes de asistencia gráfica.
Totalmente de acuerdo
De acuerdo
Ni en acuerdo, ni en desacuerdo
En desacuerdo
Totalmente en desacuerdo

- 5. El procedimiento para realizar las operaciones con la base de datos sería más fácil si pudiéramos visualizar desde el ICE (entorno de desarrollo integrado) el diagrama del modelo relacional de la base de datos.
Totalmente de acuerdo
De acuerdo
Ni en acuerdo, ni en desacuerdo
En desacuerdo
Totalmente en desacuerdo

- 6. Emplear componentes gráficos especializados en las operaciones transaccionales con base de datos incrementa total comodidad y facilidad al desarrollar aplicaciones.
Totalmente de acuerdo
De acuerdo
Ni en acuerdo, ni en desacuerdo
En desacuerdo
Totalmente en desacuerdo

- 7. La implementación de código fuente y componentes de asistencia gráfica debe ser equilibrada al momento de desarrollar aplicaciones.
Totalmente de acuerdo
De acuerdo
Ni en acuerdo, ni en desacuerdo
En desacuerdo
Totalmente en desacuerdo

DATOS PERSONALES				
Cédula	Nombres	Apellidos	Provincia	Cantón
092691871-5	Daniel Andrés	De los Santos Villao	Los Ríos	Quevedo
Parroquia	Dirección	Teléfono	Correo electrónico	
Venus del Río	17 de Marzo y Miguel Pozo	052796902	danielcls2009@live.com	

DATOS DEL EGRESADO			
Título a obtener	Carrera	Paralelo	Fecha de egreso
Ingeniero en Sistemas	Ingeniería en Sistemas	A	26 de Septiembre del 2013

DATOS DE TESIS			
Tema de tesis	Res. Aprob. Dirección de tesis	Res. Fecha de sustentación	Fecha de sustentación
CREACIÓN DE COMPONENTES DATASETS PARA NETBEANS IDE	017-HCD-FCI- 2013	013-HCD-FCI- 2014	21 de Marzo del 2014
Cédula del tutor	Nombre del tutor		Fecha de incorporación
099961856-1	Ing. Gleiston Cicerón Guerrero Ulloa		21 de Marzo del 2014