

Migración del Diagrama de Clases del Análisis al Modelo de Datos y a la Base de Datos en SQL Server 2000.x

Autor: Wilder López Meléndez¹
wlopezm@yahoo.com
<http://espanol.geocities.com/wlopezm>
Fecha: 29 de Mayo del 2008

El Rational Rose es una herramienta que puede ser utilizado para elaborar el modelado de un sistema informático en todas las fases del proyecto, con ella podemos construir todos los flujos de trabajos (workflow) que establece la metodología RUP.

El Diagrama de clases del análisis nos permite identificar las clases que utiliza el sistemas y sus relaciones, nos permite identificar también los atributos y sus métodos, estas clases normalmente se convierten en las tablas de la Base de Datos, por eso este diagrama es muy consultado en la fase de implementación, donde el desarrollador tendrá que construir las tablas con sus campos basados en este.

El Rational Rose permite la migración de este diagrama hacia el motor de base de datos que el desarrollador ha seleccionado, (DB2, Oracle, SQL Server, entre otros), sin tener que duplicar el trabajo.

En este artículo mostraremos paso a paso la creación del modelo de datos y la base de datos en SQL Server 2000, a partir del diagrama de clases, utilizaremos el Rational Rose 2003 para ello.

Se cuenta con el diagrama de clases de la figura 1 para un sistema de punto de caja, se desea migrarlo a una Base de Datos en SQL SERVER 2000.

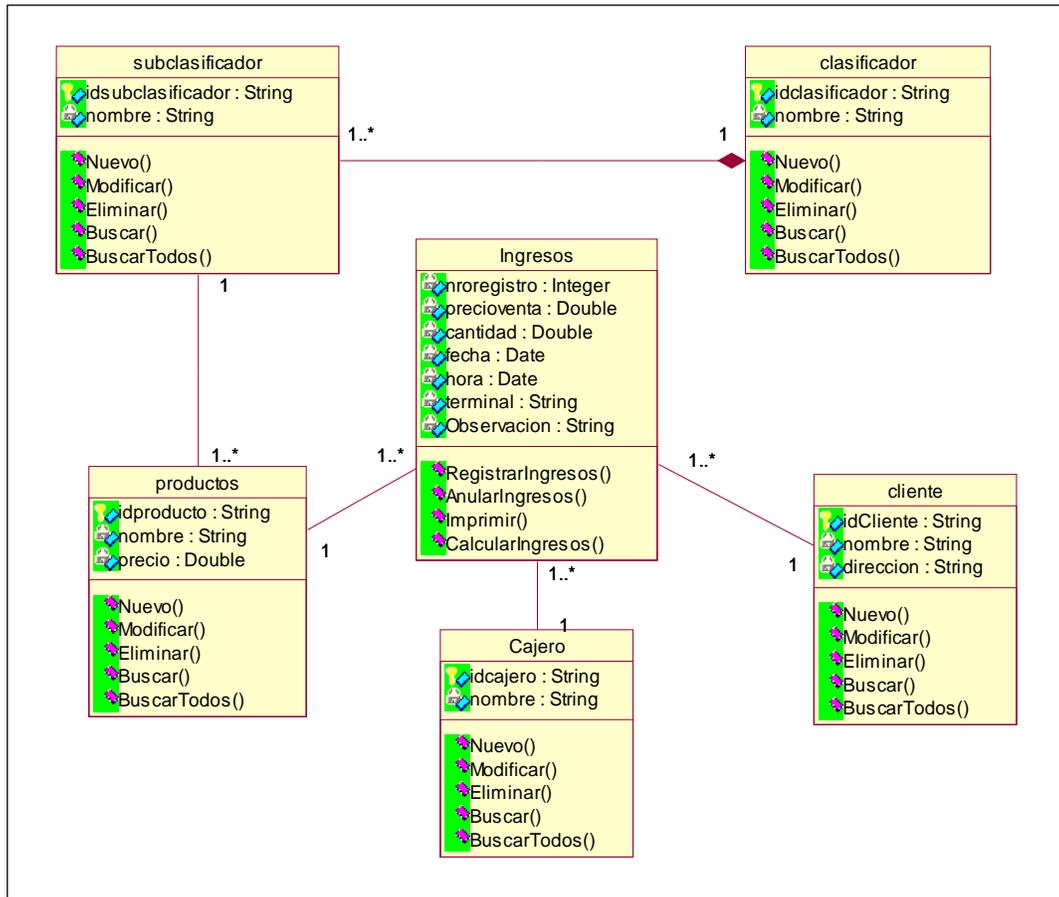


Figura 1: Diagrama de Clases en Rational Rose

Paso 1. *Cambiar todas las clases del análisis como Persistentes.*

1. Abrir el diagrama de clases.
2. Pulsar doble clic en la primera clase, se mostrará el cuadro de dialogo de *open Specification*, en ella seleccionar la pestaña *Detail*, y seleccionar la opción *Persistent*, tal como se observa en la figura 2.
3. Hacer lo mismo con las demás clases.

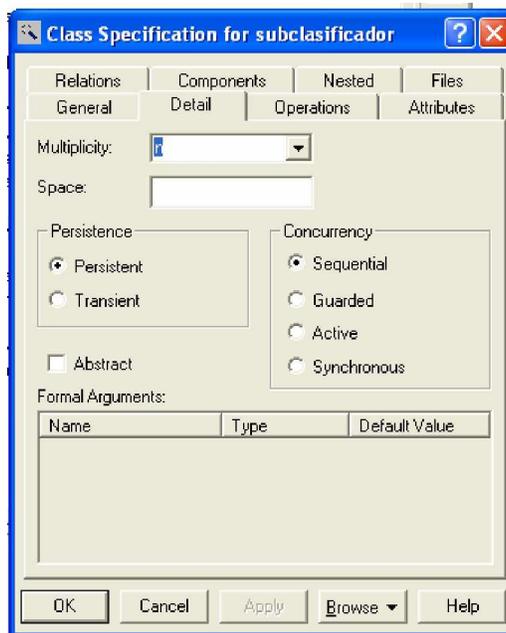


Figura 2: Cuadro de Diálogo Open Specification

PASO 2. Crear La Base de Datos en Rational (Database) ver figura3

1. Pulsar clic derecho en el Paquete Component View, seleccionar la opción *Data Modeler / New / Database*
2. Ponerle un nombre adecuado que tenga relación con el sistema.
3. Especificar el Motor de la Base de Datos a migrar, doble clic en la base de datos creado en el punto anterior, nos mostrara el cuadro de dialogo de las especificaciones de la base de datos, ubicar el campo *Target* seleccionar de la lista el motor de base de datos en nuestro caso *Microsoft SQL Server 2000.x*

PASO 3. Creación del Schema que almacenara las tablas de la Base de Datos y servirá para la elaboración el diagrama Modelo de Datos

1. Ingresar al paquete *Logical View, Schema*, clic derecho sobre *Schema*, *Data Modeler / New / Schema*
2. Ponerle un nombre

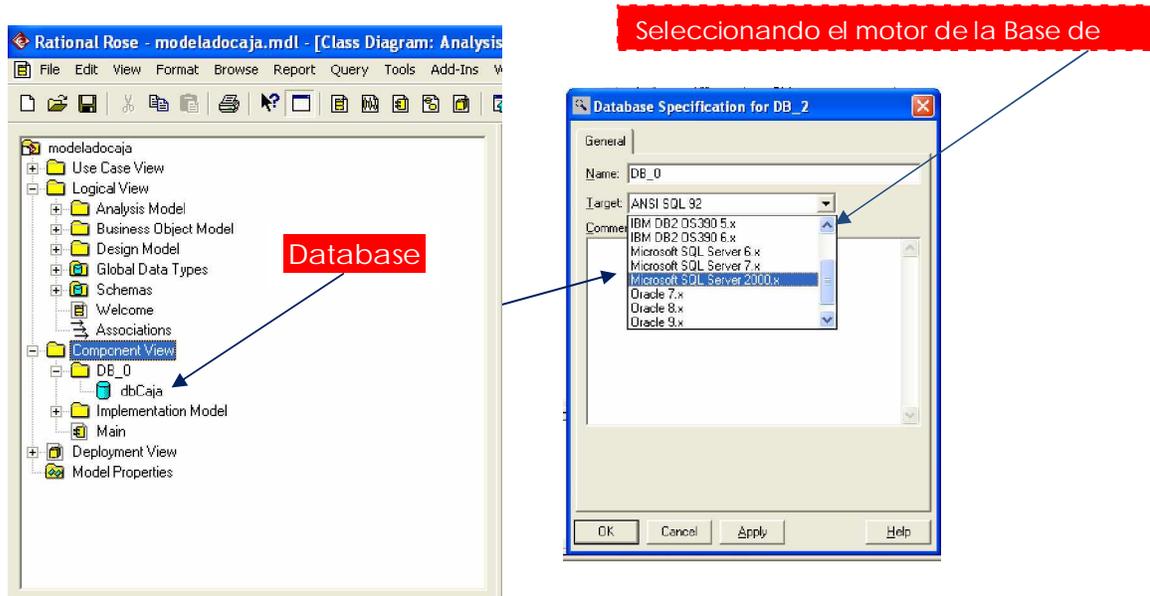


Figura 3: Database creado, seleccionando el motor de la base de datos

PASO 3. Transformar el diagrama de clases al Modelo de Datos, definiendo la base de datos y el Schema donde pondrá las tablas generadas. Ver figura 4

1. Ingresar al paquete donde este el diagrama de clases de la figura 1, en nuestro caso *Logical View / Analisis Model*.
2. Pulsar clic derecho sobre el paquete ingresado *Analiis Model, Data Modeler / Transform to Data Model*
3. En el cuadro de dialogo seleccionar los campos Destination Schema elegir el schema creado en el paso 4, Target Database elegir la base de datos creado en el paso 2. Finalmente ingresar un prefijo que se pondrán a las tablas
4. Pulse OK para finalizar.

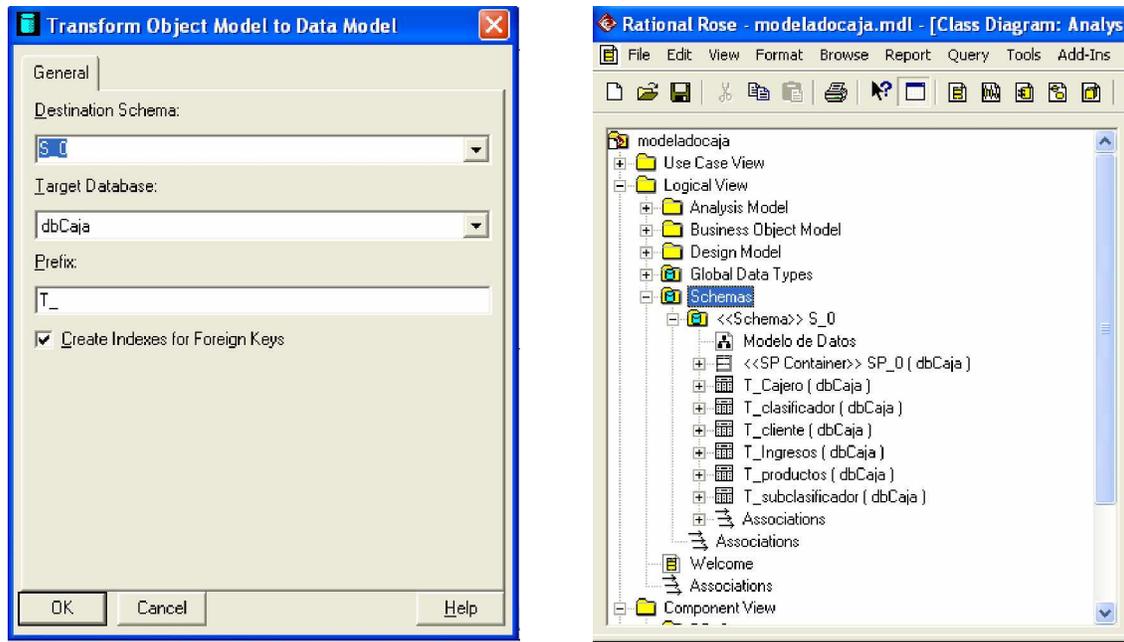


Figura 4: Transformación del diagrama de clases al Modelo de Datos

Paso 5. Creación del Modelo de Datos

1. Ubicarse en el Schema donde se encuentran las tablas generadas, clic derecho, *Data Modeler / New / Data Model Diagram*
2. Cambiarlo de nombre con la opción Rename, a un nombre que tenga relación con el proyecto
3. Pulsar doble clic sobre el modelo de datos creado y empezar a arrastrar cada una de las tablas hacia el diagrama

Observación, el Rational Rose genera automáticamente los campos claves primarias (PK) de las tablas sin importar si ya teníamos definido uno, esto debe ser arreglado manualmente de igual manera para los campos claves foráneas (FK).

Paso 6. Arreglar los campos de las tablas y definir las PK y FK ver figura 5

1. Seleccionar la tabla a corregir desde el Schema
2. Seleccionar el campo Tabla_ID generado por el Rational y eliminarlo con *clic derecho / delete*.
3. Seleccionar el campo a definir como PK y pulsar doble click sobre ella, nos mostrara la ventana de las especificaciones del campo, seleccionar

la pestaña *Type* y activar el check *Primary Key*. Ver figura 5. Pude aprovechar también para cambiar la longitud del campo.

4. Repetir los mismos pasos para todos los campos de las tablas, hasta que el diagrama quede como el de la figura 6.

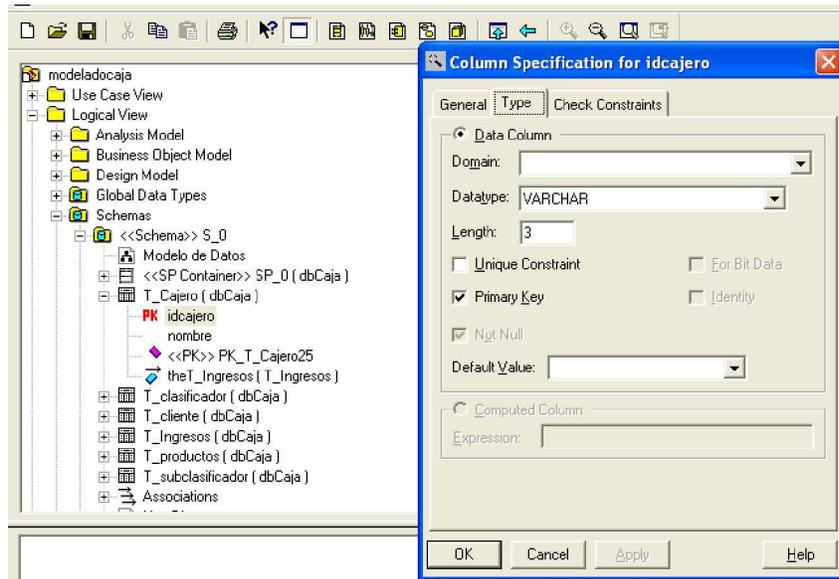


Figura 5: Corrección de los campos de las tablas del modelo de datos

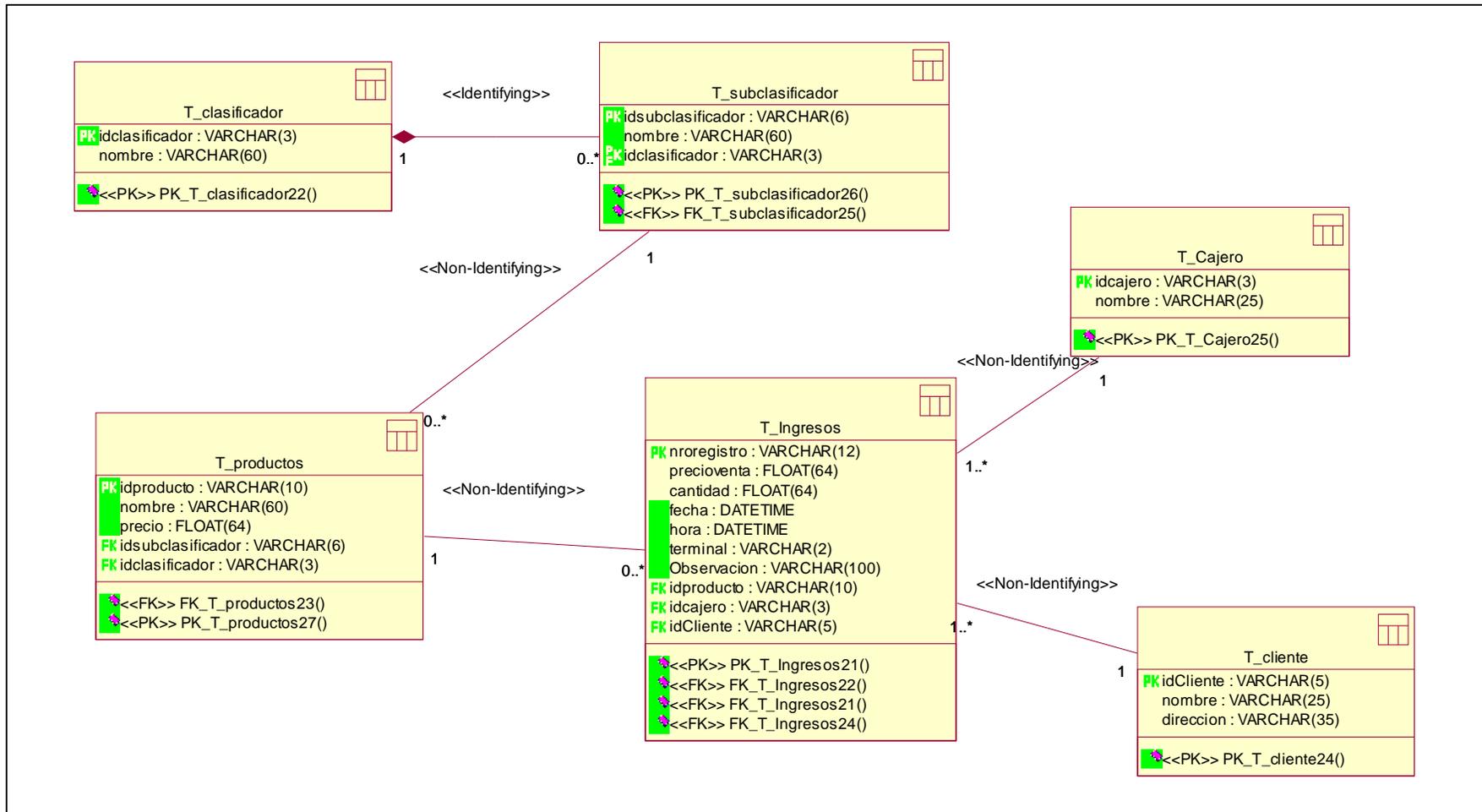
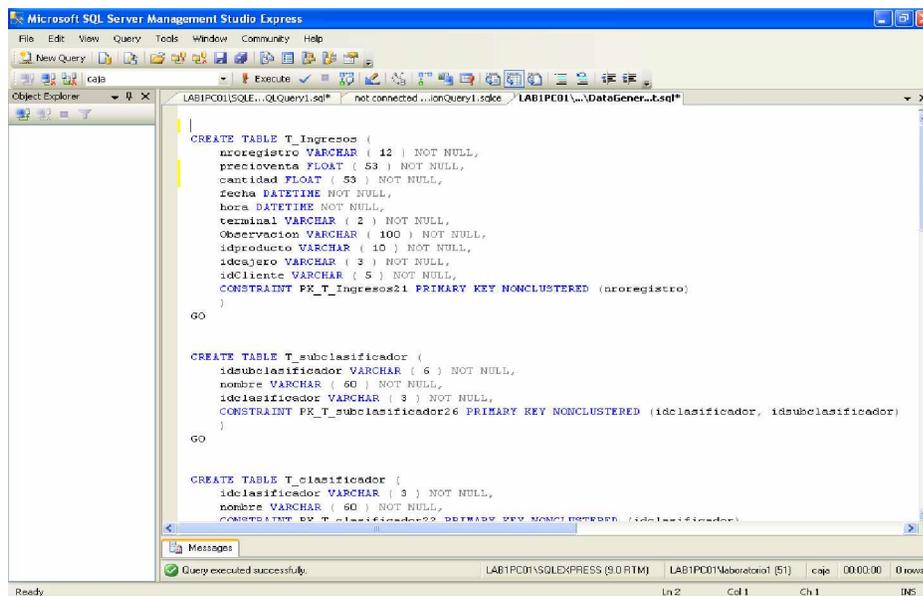


Figura 6: Diagrama Modelo de Datos / Esquema de Datos corregido

Paso 7. Convertir a la Base de Datos física

1. Seleccionar la base de datos creado en el paso 2
2. Clic derecho, *Data Model / Forward Engineer* mostrara un asistente que le guiara durante el proceso
3. En el cuadro de dialogo pulsar Next / luego mostrará las opciones a generar pulsar Next, luego ingresar el nombre del archivo script a generar, también puede seleccionar la base de datos del motor, debe estar instalado el SQL server y funcionando para que esta opción funcione. Pulsar Next, y luego finalizar
4. Ubicar el Script generado y ejecutarlo desde el Analizador de Consultas del SQL Server 2005. Ver figura 8. No olvidar de crear una Base de Datos donde alojar las tablas de este script.
5. Finalizamos la creación de la Base de Datos, ya puede utilizarlo



```
Microsoft SQL Server Management Studio Express
File Edit View Query Tools Window Community Help
New Query
Execute
Object Explorer LAB1PC01\SQLEXPRESS\QLQuery1.sql* not connected ...ionQuery1.sqke LAB1PC01\...DataGener...t.sql*
CREATE TABLE T_Ingresos (
  nroregistro VARCHAR ( 12 ) NOT NULL,
  precioventa FLOAT ( 53 ) NOT NULL,
  cantidad FLOAT ( 53 ) NOT NULL,
  fecha DATETIME NOT NULL,
  hora DATETIME NOT NULL,
  terminal VARCHAR ( 2 ) NOT NULL,
  Observacion VARCHAR ( 100 ) NOT NULL,
  idproducto VARCHAR ( 10 ) NOT NULL,
  idcajero VARCHAR ( 3 ) NOT NULL,
  idcliente VARCHAR ( 5 ) NOT NULL,
  CONSTRAINT FK_T_Ingresos21 PRIMARY KEY NONCLUSTERED (nroregistro)
)
GO

CREATE TABLE T_subclasificador (
  idsubclasificador VARCHAR ( 6 ) NOT NULL,
  nombre VARCHAR ( 60 ) NOT NULL,
  idclasificador VARCHAR ( 3 ) NOT NULL,
  CONSTRAINT FK_T_subclasificador26 PRIMARY KEY NONCLUSTERED (idclasificador, idsubclasificador)
)
GO

CREATE TABLE T_clasificador (
  idclasificador VARCHAR ( 3 ) NOT NULL,
  nombre VARCHAR ( 60 ) NOT NULL,
  CONSTRAINT PK_T_clasificador23 PRIMARY KEY NONCLUSTERED (idclasificador)
)
Messages
Query executed successfully. LAB1PC01\SQLEXPRESS (9.0 RTM) LAB1PC01\Motorola1 (51) cap 00:00:00 0 rows
Ready Ln2 Col1 Ch1 DNS
```

Figura 7: Analizador de Consultas del SQL Server 2005

1 Acerca del Autor

- ü Docente Universitario en J2EE, J2SE e Ingeniería de Software
- ü Miembro de la Sociedad Peruana de Computación
- ü Cuenta con 10 años de experiencia en el área de Desarrollo de Software, dedicando los últimos 04 años al uso exclusivo de la tecnología JAVA y a la metodología RUP
- ü Ingeniero de Sistemas titulado por la Universidad Privada del Norte – Trujillo – Perú

