

Este documento foi preparado por:

José Sahle Netto	
------------------	--

Histórico de Versões

Data	Versão	Histórico da Revisão	Autor/Revisor
17/07/2013	1.0		José Sahle Netto

Objetivo

Este documento tem por objetivo apresentar as funcionalidades do **SHLStudio** e do *plug-in Gerador de Código C# e Stored Procedures*.

O projeto do SHLStudio está escrito em C++ utilizando as classes MFC.

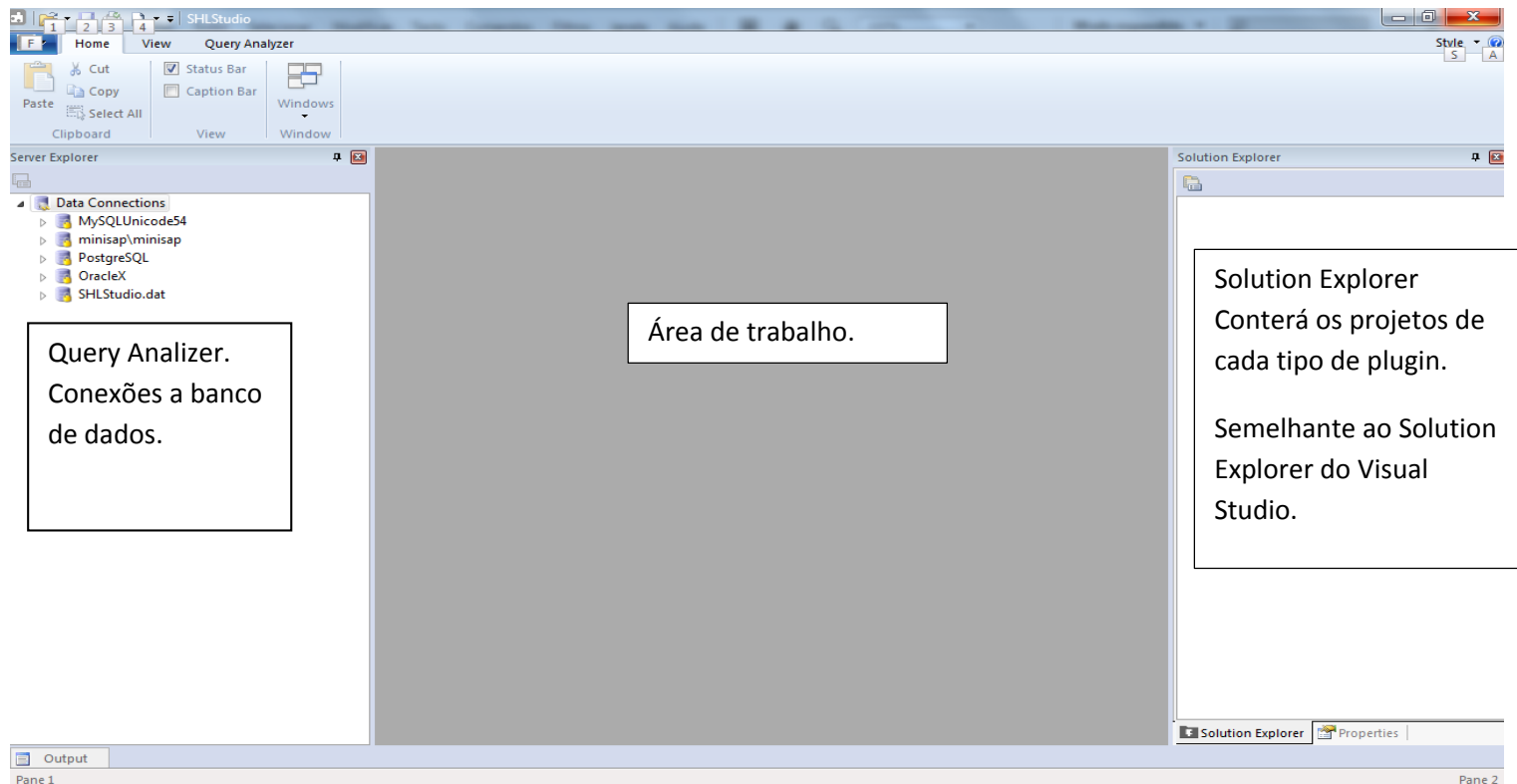
Este projeto está em fase de testes, onde estou testando e corrigindo eventuais falhas.

Optei em utilizar, inicialmente, *menus* e *descrições* em inglês.

SHLStudio

O sistema **SHLStudio** é um IDE que tem por finalidade a acoplagem de funcionalidades via plug-ins.

O sistema é um gerenciador de *projetos* possuindo um **Query Analyzer** embutido.



SHLStudio contém um **Query Analyzer** embutido. Na área à esquerda indicada são as conexões feitas aos banco (MS-SQL Server, Oracle, MySQL, PostGreSQL, e qualquer outro via OLE DB).

Possui uma área de trabalho, onde os editores de **script** serão abertos. Neste local também será usado para abrir as funcionalidades dos **plugins**.

Por fim, à direita está o **SOLUTION EXPLORER**, que gerenciará os projetos de cada tipo de plugin. Estes projetos serão agrupados dentro de uma *solução* semelhante ao **VISUAL STUDIO**.

Query Analyzer

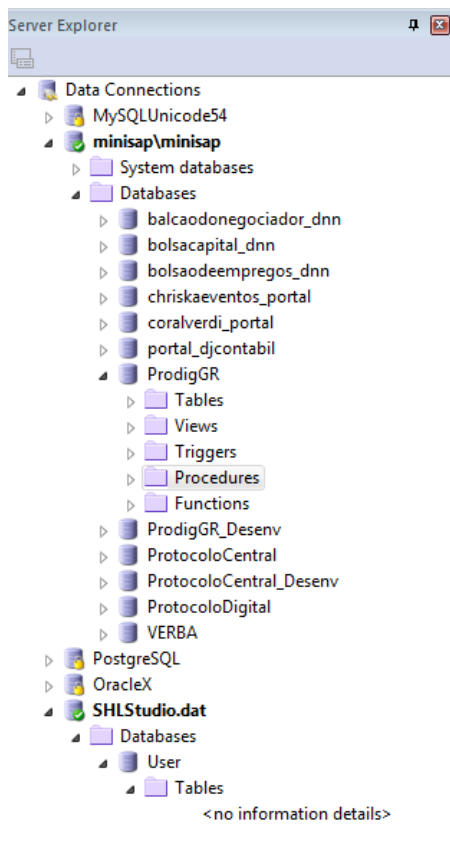
The screenshot displays the SQL Server Query Analyzer window. The top menu bar includes 'Home', 'View', and 'Query Analyzer'. The 'Database' dropdown is set to 'ProdigGR'. The 'SQL' tab is active, showing a query with two lines: '1 SELECT * FROM GP_ROTINA' and '2 SELECT * FROM GP_DLL'. The 'Result' tab is also visible, showing a table with columns 'idRotina' and 'Nome'. The table contains 12 rows of data, including 'TJBAProjudi', 'Distribuidor-PainelAutomator', 'SaneamentoPilotoPCES', and various 'TJ' Automator entries. The 'Server Explorer' on the left shows the database structure, including 'GP_ROTINA' and 'GP_DLL'. The 'Output' pane at the bottom is empty.

	idRotina	Nome
		Executed query
		SELECT * FROM GP_ROTINA
		Result
1	2	TJBAProjudi
2	4	Distribuidor-PainelAutomator
3	6	SaneamentoPilotoPCES
4	7	TJSudesteAutomator
5	8	TJNorteAutomator
6	9	TJNordesteAutomator
7	10	TJCentroOesteAutomator
8	11	TJSulAutomator
9	15	Gerador de Código
10	16	Gerador de Código
11	17	Gerador de Código
12	18	Gerador de Código

O **Query Analyzer** possui muitas funcionalidades semelhantes ao **SQL Server Management Studio**, porém estendido para quaisquer bancos de dados suportados por OLE DB.

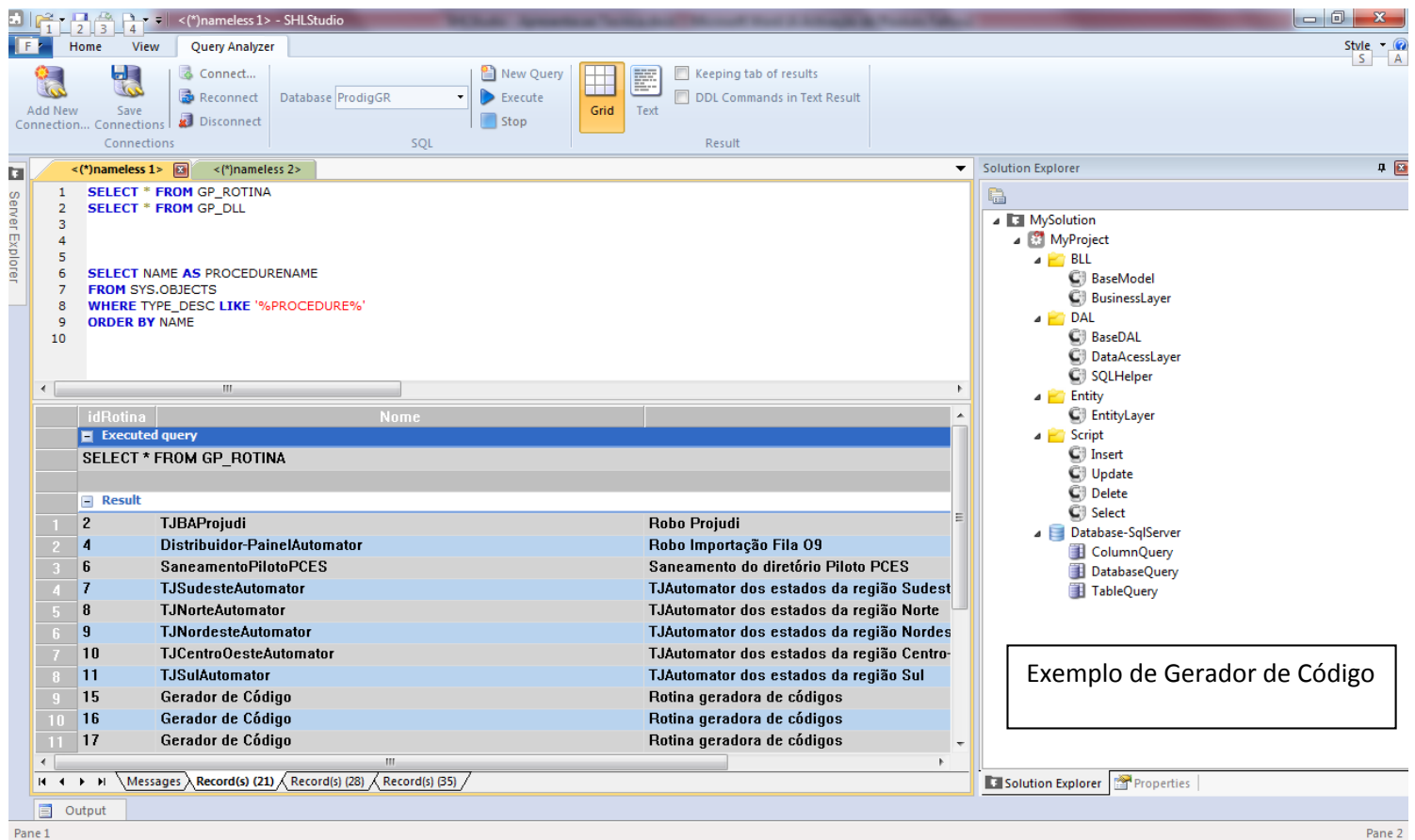
	idRotina	Nome	
Executed query			
SELECT * FROM GP_ROTINA			
Result			
1	2	TJBAProjudi	Robo Projudi
2	4	Distribuidor-PainelAutomator	Robo Importação Fila 09
3	6	SaneamentoPilotoPCES	Saneamento do diretório Piloto PCES
4	7	TJSudesteAutomator	TJAutomator dos estados da região Sudeste
5	8	TJNorteAutomator	TJAutomator dos estados da região Norte
6	9	TJNordesteAutomator	TJAutomator dos estados da região Nordeste
7	10	TJCentroOesteAutomator	TJAutomator dos estados da região Centro-Oeste
8	11	TJSulAutomator	TJAutomator dos estados da região Sul
9	15	Gerador de Código	Rotina geradora de códigos
10	16	Gerador de Código	Rotina geradora de códigos
11	17	Gerador de Código	Rotina geradora de códigos
12	18	Gerador de Código	Rotina geradora de códigos

Diferentemente do **SQL Management Studio** quando existem mais de um recordset estes não são apresentados um abaixo dos outros. Nestes casos os resultados serão apresentados em abas separadas. Outro detalhe importante, é que no topo da grade, está a sentença que gerou o resultado, assim com multiplas consultas, fica fácil identificar a dupla *consulta-resultado*.



No *Server Explorer* possui informações adicionais como:

- Tables
- Views
- Triggers
- Functions
- Procedures

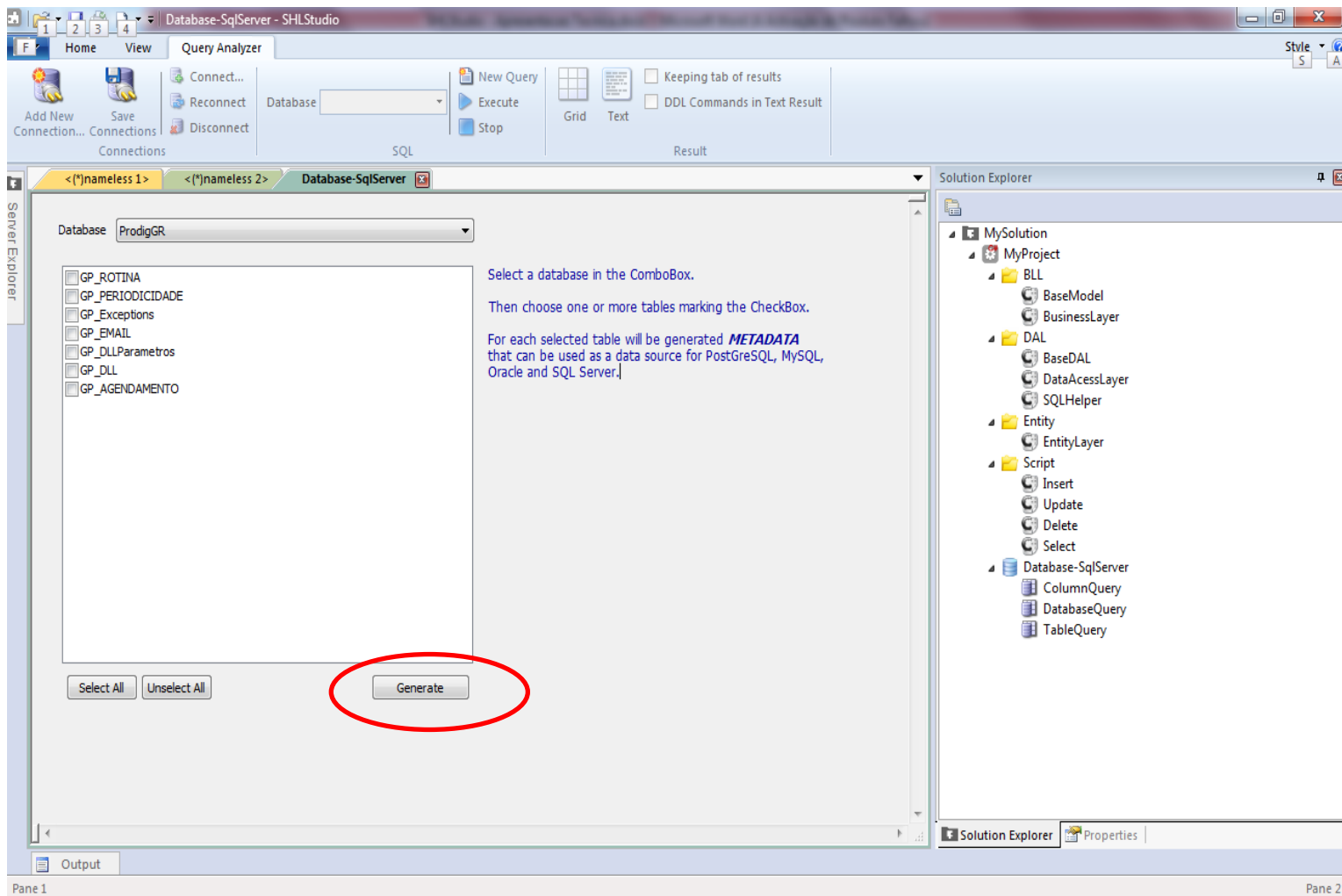


O **Solution Explorer** possui o projeto de um determinado plugin. Dentro da solução será possível possuir projetos de diferentes plugins.

Gerador de Código C#

O *plugin Gerador de Código C#* é uma rotina que gera códigos para C# nas 3 (três) principais camadas (entidade, negócio e dados). Este plugin também gera as ***stored procedures*** utilizadas na camada de dados.

É possível ampliar sua funcionalidade de gerador de código, pois ele os gera baseados em **bancos e tabelas**, utilizando de *tags* na substituição dos ***arquivos modelos***.



A entidade **Database-SqlServer** é na realidade um *metadado* gerado a partir de uma conexão com determinado banco de dados, que podem ser (Oracle, PostgreSQL, MySQL e SQL Server).

A partir destas conexões gera-se um *metadado* representando o banco de dados, que é um arquivo XML, e poderá ser editado externamente, embora na maioria das vezes não haja esta necessidade.

Database-SqlServer - SHLStudio

Home View Query Analyzer

Connect... Reconnect Database SQL New Query Execute Grid Text Keeping tab of results DDL Commands in Text Result Stop Result

Server Explorer

Database-SqlServer

```

<Settings>
<Table Name="GP_ROTINA">
<Field Name="idRotina" PrimaryKey="1" Identity="1" DBType="Int32" DBSize="0" ForeignKey="0" TableForeignKey=""/>
<Field Name="Nome" PrimaryKey="0" Identity="0" DBType="String" DBSize="50" ForeignKey="0" TableForeignKey=""/>
<Field Name="Descricao" PrimaryKey="0" Identity="0" DBType="String" DBSize="500" ForeignKey="0" TableForeignKey=""/>
<Field Name="DataCriacao" PrimaryKey="0" Identity="0" DBType="DateTime" DBSize="0" ForeignKey="0" TableForeignKey=""/>
<Field Name="Usuario" PrimaryKey="0" Identity="0" DBType="String" DBSize="20" ForeignKey="0" TableForeignKey=""/>
<Field Name="Funcional" PrimaryKey="0" Identity="0" DBType="String" DBSize="20" ForeignKey="0" TableForeignKey=""/>
<Field Name="Ativo" PrimaryKey="0" Identity="0" DBType="Int16" DBSize="0" ForeignKey="0" TableForeignKey=""/>
<Field Name="Apelido" PrimaryKey="0" Identity="0" DBType="String" DBSize="50" ForeignKey="0" TableForeignKey=""/>
</Table>
<Table Name="GP_PERIODICIDADE">
<Field Name="idPeriodicidade" PrimaryKey="1" Identity="1" DBType="Int32" DBSize="0" ForeignKey="0" TableForeignKey=""/>
<Field Name="idRotina" PrimaryKey="0" Identity="0" DBType="Int32" DBSize="0" ForeignKey="0" TableForeignKey=""/>
<Field Name="Tipo" PrimaryKey="0" Identity="0" DBType="Int32" DBSize="0" ForeignKey="0" TableForeignKey=""/>
<Field Name="Parametro" PrimaryKey="0" Identity="0" DBType="String" DBSize="100" ForeignKey="0" TableForeignKey=""/>
</Table>
<Table Name="GP_Exceptions">
<Field Name="idExceptions" PrimaryKey="1" Identity="1" DBType="Int32" DBSize="0" ForeignKey="0" TableForeignKey=""/>
<Field Name="idRotina" PrimaryKey="0" Identity="0" DBType="Int32" DBSize="0" ForeignKey="0" TableForeignKey=""/>
<Field Name="Tentativa" PrimaryKey="0" Identity="0" DBType="Int32" DBSize="0" ForeignKey="0" TableForeignKey=""/>
<Field Name="Intervalo" PrimaryKey="0" Identity="0" DBType="Int32" DBSize="0" ForeignKey="0" TableForeignKey=""/>
<Field Name="EnviarEmail" PrimaryKey="0" Identity="0" DBType="Int16" DBSize="0" ForeignKey="0" TableForeignKey=""/>
<Field Name="EnviarUltimoEmail" PrimaryKey="0" Identity="0" DBType="Int16" DBSize="0" ForeignKey="0" TableForeignKey=""/>
</Table>
<Table Name="GP_EMAIL">
<Field Name="idEMAIL" PrimaryKey="1" Identity="1" DBType="Int32" DBSize="0" ForeignKey="0" TableForeignKey=""/>
<Field Name="idRotina" PrimaryKey="0" Identity="0" DBType="Int32" DBSize="0" ForeignKey="0" TableForeignKey=""/>
<Field Name="FROM" PrimaryKey="0" Identity="0" DBType="String" DBSize="500" ForeignKey="0" TableForeignKey=""/>
<Field Name="TO" PrimaryKey="0" Identity="0" DBType="String" DBSize="500" ForeignKey="0" TableForeignKey=""/>
<Field Name="SUBJECT" PrimaryKey="0" Identity="0" DBType="String" DBSize="500" ForeignKey="0" TableForeignKey=""/>
<Field Name="TYPE" PrimaryKey="0" Identity="0" DBType="String" DBSize="5" ForeignKey="0" TableForeignKey=""/>
<Field Name="BODY" PrimaryKey="0" Identity="0" DBType="String" DBSize="8000" ForeignKey="0" TableForeignKey=""/>
<Field Name="Classificacao" PrimaryKey="0" Identity="0" DBType="Int32" DBSize="0" ForeignKey="0" TableForeignKey=""/>
<Field Name="Anexo" PrimaryKey="0" Identity="0" DBType="String" DBSize="1000" ForeignKey="0" TableForeignKey=""/>
<Field Name="Ativo" PrimaryKey="0" Identity="0" DBType="Int16" DBSize="0" ForeignKey="0" TableForeignKey=""/>

```

Solution Explorer

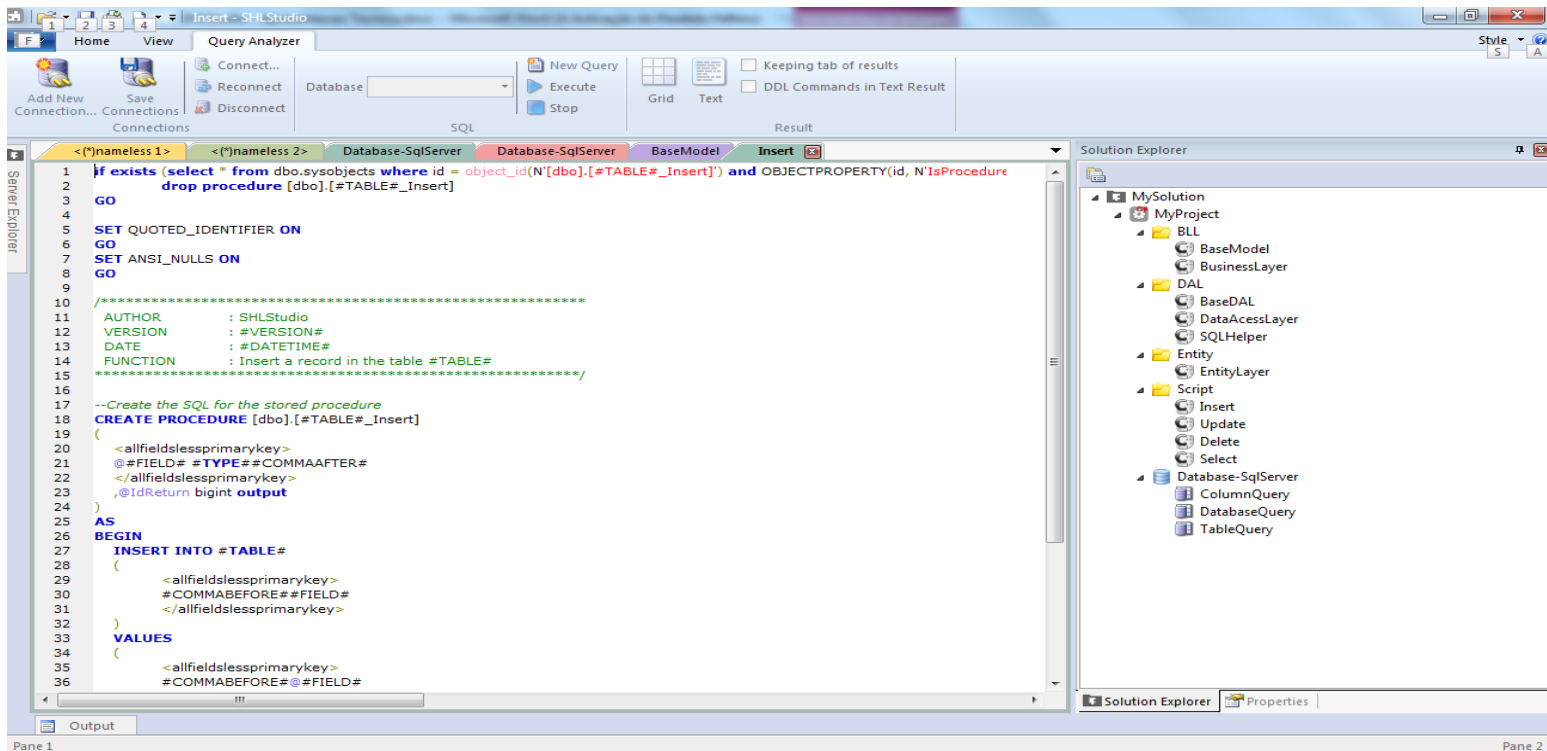
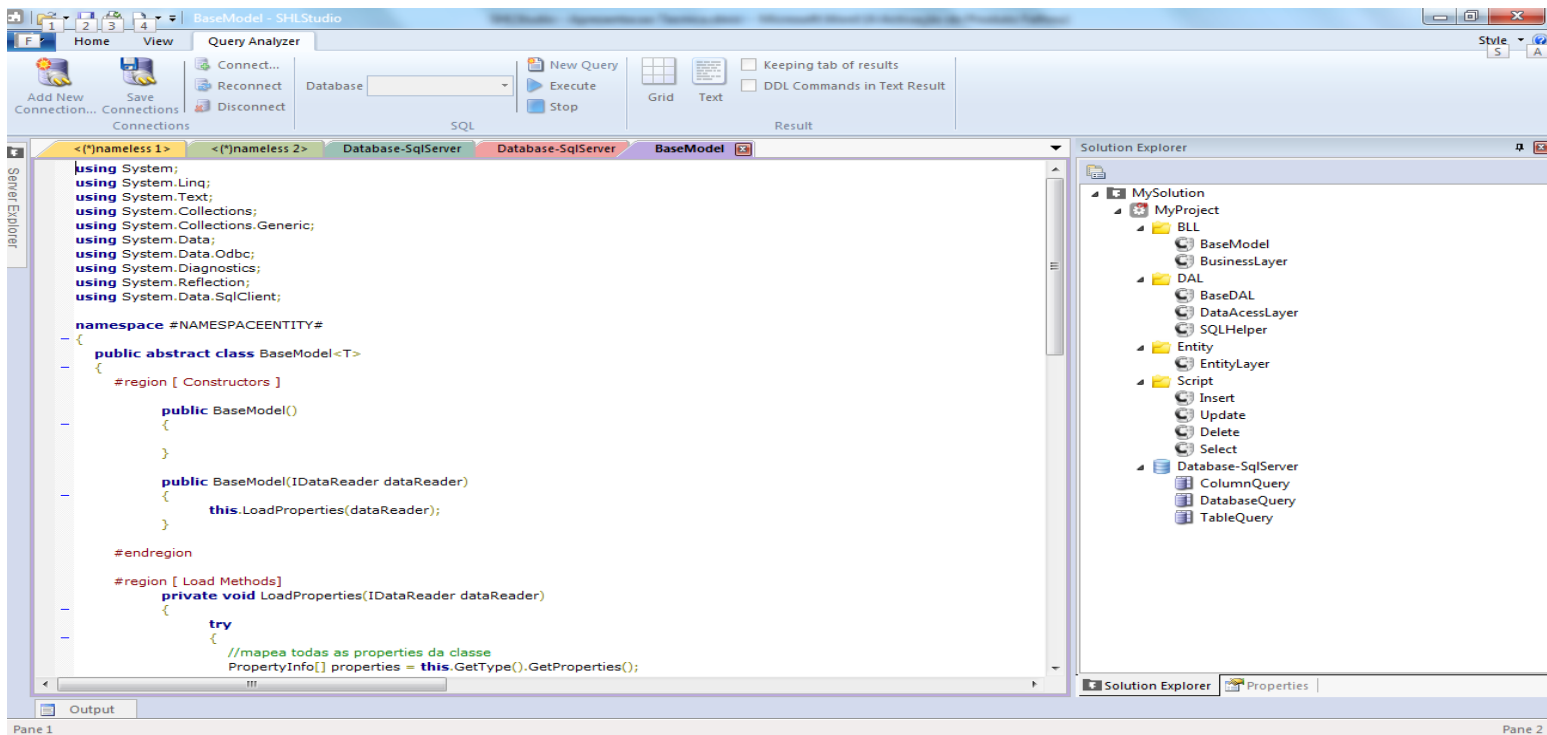
MySolution

- MyProject
 - BLL
 - BaseModel
 - BusinessLayer
 - DAL
 - BaseDAL
 - DataAccessLayer
 - SQLHelper
 - Entity
 - EntityLayer
 - Script
 - Insert
 - Update
 - Delete
 - Select
 - Database-SqlServer
 - ColumnQuery
 - DatabaseQuery
 - TableQuery

Pane 1

Pane 2

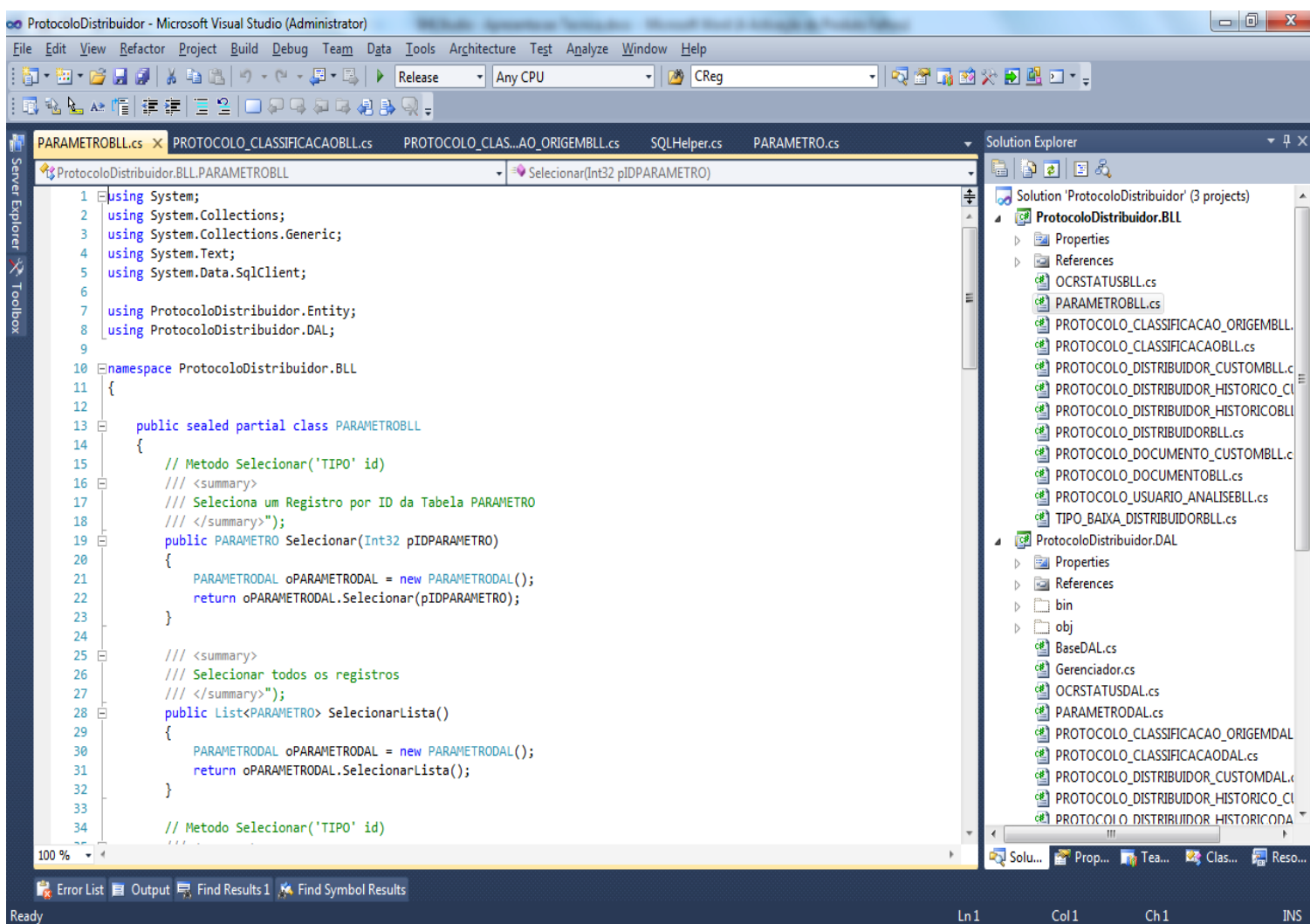
Acima está o resultado em XML do *metadado* gerado a partir de um banco de dados e suas tabelas selecionados anteriormente.



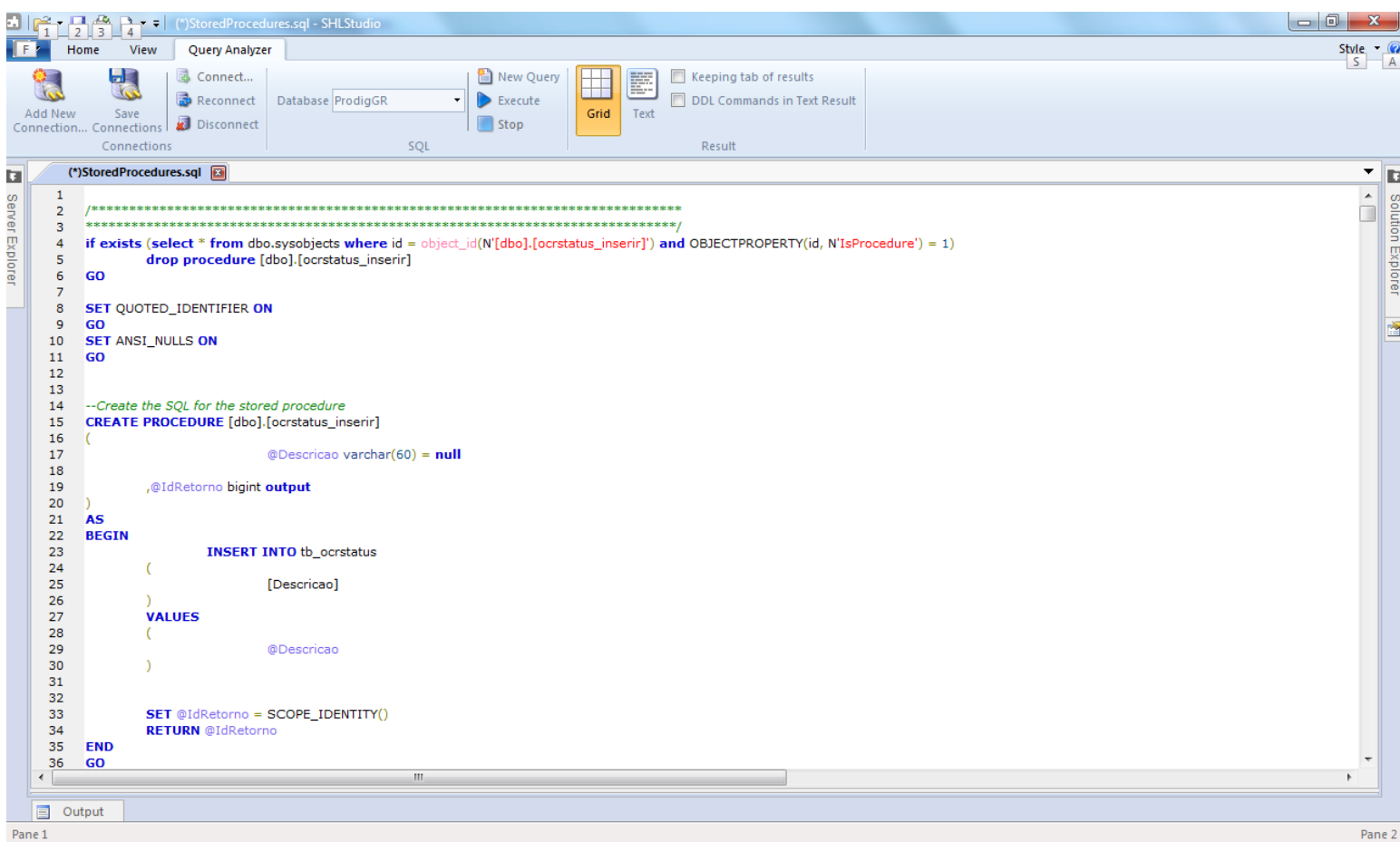
O códigos gerados para um projeto C# são feitos a partir de *templates* utilizando *tags* que substituem os elementos das tabelas selecionadas e geram o código apropriado.

Exemplo de Código Gerado

Abaixo um exemplo de códigos gerados a partir do *plugin Gerador de Código*.



As classes contantes neste projeto foram criadas pelo **gerador de código**, o que não levou nem ao menos 1 minuto, considerando a seleção das tabelas e a “ordem” ao **SHLStudio** gerar as classes.



As **stored procedures** também foram geradas ao mesmo momento.

Totalizando 12 tabelas selecionadas resultando em 36 arquivos C# (entidade, negócio, dados) e 60 procedure (inclusão, alteração, exclusão e consultas).