

RELATÓRIO DE ESTÁGIO:
Tânia Cristina Leite RA: 046567
Orientador: Prof. Dr. Aurelio Ribeiro Leite de Oliveira

FLUXO DE CAIXA: Módulo BI (Business Intelligence)

**Universidade Estadual de Campinas
Instituto de Matemática, Estatística e Computação Científica**

1º Semestre de 2012

APRESENTAÇÃO DA EMPRESA EM QUE ESTAGIEI

A empresa está no mercado há aproximadamente 15 anos, tendo como sua principal atividade a criação e implantação de *ERPs* para empresas em diversas áreas como: Clínicas médicas, Laboratórios clínicos, Área alimentícia, entre outros. Sendo seu principal nicho de mercado a área industrial (metalúrgicas, caldeirarias, etc.).

Sistemas Integrados de Gestão Empresarial (SIGE ou SIG), em inglês *Enterprise Resource Planning (ERP)*, são sistemas de informação que integram todos os dados e processos de uma organização em um único sistema. A integração pode ser vista sob a perspectiva funcional (sistemas de: finanças, contabilidade, recursos humanos, fabricação, *marketing*, vendas, compras, etc.) e sob a perspectiva sistêmica (sistema de processamento de transações, sistemas de informações gerenciais, sistemas de apoio à decisão, etc.).

Os *ERPs*, em termos gerais, são uma plataforma de software desenvolvida para integrar os diversos departamentos de uma empresa, possibilitando a automação e armazenamento de todas as informações de negócios.

O Programa de estágio que estou realizando é na área de personalização e suporte para o módulo *B.I. (Business Intelligence)* módulo este, responsável por fornecer informações gerenciais e cruciais para: gestores financeiros, produção e custos dos clientes.

Atividades Desenvolvidas

Minha principal atividade desenvolvida está diretamente ligada com o módulo *B.I.* (Business Intelligence).

B.I. é uma ferramenta responsável por retornar de forma rápida e de fácil entendimento dados cruciais para a administração de uma empresa. Provendo dados históricos, presentes e previsões em diversas áreas como: Custos, Produção, Faturamento, Fluxo de Caixa, etc. Tais dados são visualizados na forma de gráficos, relatórios e pivots (matrizes que cruzam dados) . Segue abaixo um exemplo de retorno de dados, que neste caso, representa o fluxo de caixa de um cliente hipotético:

“Divulgação pública não permitida”

Minhas atividades são Analisar e Personalizar os módulos financeiros da ferramenta: Fluxo de caixa, Contas a Pagar e Contas a Receber; feitas com o auxílio e supervisão de um dos analistas da empresa.

Para isso, faço utilização de ferramentas desenvolvidas internamente pela própria empresa ou acessos diretos ao banco de dados, que no caso, é o SQL Server versões 2005 e 2008.

Como a ferramenta funciona em vários clientes, e cada um possui sua peculiaridade, faço a análise da solicitação do cliente e, junto com o desenvolvedor da ferramenta, decidimos a melhor maneira de se criar/alterar a visualização dos dados para a realidade do cliente. Exemplo: um caso em que um dos clientes gostaria de ver anexo ao seu Contas a Pagar dados fabris (como horas gastas de máquinas para a produção).

Para isso, no caso do banco de dados, faço utilização/alterações em *Procedures*, *Views* e *Functions*. Segue definição de cada uma delas :

Procedure:

São subrotinas que se encontram dentro do banco de dados e que realizam algumas análises e processamento de dados requisitados pelo Software em si. No final, retornam esses dados trabalhados em formas diversas, podendo variar de simples resultados diretos até uma tabela completa de resultados. *Procedures* podem gerar alterações diretas nas tabelas do banco de dados como: incluir dados, excluir ou mesmo altear registros específicos.

Segue um exemplo de *Procedure*, neste caso, utilizada para retornar dados referentes a faturamento de uma empresa:

Exemplo de *Procedure*:

“Divulgação pública não permitida”

Functions:

Assim como as *procedures*, *functions* são rotinas no banco de dados que executam um ou alguns blocos de instruções, porém, diferente das *procedures*, uma *function* não pode gerar alterações como as citadas acima, ou seja, não pode gerar alterações diretamente no banco de dados. Elas somente manipulam a informação em tempo de execução e retornam um resultado simples para o módulo que fez a chamada da mesma.

O Resultado pode ser apresentado em forma de uma tabela de resultados (*table*) ou um retorno escalar, que seria o retorno de um simples campo (valor, resultado de um cálculo, registro específico como um código, etc.).

Segue um exemplo de uma *Function*, que neste caso, retorna o resultado de um cálculo para custo de horas com base em um valor informado pelo sistema (valor este podendo ser o salário base de um funcionário e a quantidade de horas gastas em determinada operação fabril).

Exemplo de uma *Function*:

“Divulgação pública não permitida”

Views:

Uma *view* nada mais é do que um método de visualização de dados, porém processado também em ambiente de servidor, assim como as instruções citadas anteriormente.

Simplificando, pode ser chamada de tabela virtual podendo ser populada por resultados de várias consultas em diferentes tabelas do banco de dados (as quais podem ou não ter relação entre si). Retornando uma tabela que pode ser consultada e trabalhada por meio de comandos SQL.

Um Exemplo Prático Do Meu Dia a Dia:

Um exemplo prático do meu dia a dia, que foi realizado recentemente, foi uma melhoria em uma das rotinas do módulo de Fluxo de Caixa, onde foi necessária a inclusão de uma nova visão para cruzamento de dados.

A princípio, os dados visualizados pelo fluxo eram principalmente compostos por: Contas a Receber (Previsto e Realizado), Contas a Pagar (Previsto e Realizado), Cheques, etc. Recentemente, surgiu a necessidade em um cliente da visualização de Contas a Pagar PROVISIONADO (que seriam, grosso modo, contas que não se tornaram efetivas no caixa da empresa), esta situação pode ocorrer por vários motivos, entre eles: Compras realizadas, porém ainda não recebidas. No entanto, para uma análise melhor de planejamento, seria necessário visualizar a situação com estes valores para que o responsável pelo financeiro da empresa tenha ideia de quanto dinheiro em caixa ele realmente dispõe (contando com todas as variáveis).

Segue visão que se tinha antes da melhoria realizada:

“Divulgação pública não permitida”

A seguir, a visão conseguida após a melhoria:

“Divulgação pública não permitida”

Comparando as duas imagens, nota-se que há uma diferença de aproximadamente R\$6000,00 negativos, que representa uma possível conta para o dia 19/07/2012

(lembrando que esta ainda não foi efetivada, porém provisionada dentro dos cálculos).

Segue parte da rotina implementada para conseguir o resultado apresentado:

“Divulgação pública não permitida”

Neste caso, esta rotina encontra-se dentro de uma *Procedure* que faz uso de uma tabela temporária (tabela criada na memória em tempo de execução, sendo que, assim que concluída a execução da *procedure* específica, esta tabela é descarregada da memória) representada pela variável de memória @ARRAY_RET que está sendo populada pelos dados de pagamento (provenientes da consulta à tabela auxiliar ao pedido de compra) que posteriormente se tornarão (ou não) contas a pagar efetivos.

Conclusão:

Como cada cliente tem uma necessidade específica, aprendi que é preciso ter paciência e saber interpretar as exigências de cada um deles. Ter muita dedicação e estar sempre atenta para absorver o que o cliente deseja (quais são suas necessidades) para que se possa, então, chegar a um resultado satisfatório.

Além do mais, o estágio me permitiu tomar contato com o banco de dados SQL Server. Entender seu funcionamento foi um grande desafio e aprendizado.