



UNIVERSIDADE ESTADUAL DO NORTE DO PARANÁ
FACULDADES LUIZ MENEGHEL



VINÍCIUS DE LARA ALVES

**SISTEMA DE GERENCIAMENTO EMPRESARIAL
PARA PNEUMÁTICA**

Bandeirantes

2009

VINÍCIUS DE LARA ALVES

**SISTEMA DE GERENCIAMENTO EMPRESARIAL
PARA PNEUMÁTICA**

Trabalho de Conclusão de Curso submetido às Faculdades Luiz Meneghel da Universidade Estadual do Norte do Paraná, como requisito parcial para a obtenção do grau de Bacharel em Sistemas de Informação.

Orientador: Prof. Msc. Ailton Sergio Bonifacio.

Bandeirantes

2009

VINICIUS DE LARA ALVES

**SISTEMA DE GERENCIAMENTO EMPRESARIAL
PARA PNEUMÁTICA**

Trabalho de Conclusão de Curso submetido às Faculdades Luiz Meneghel da Universidade Estadual do Norte do Paraná, como requisito parcial para a obtenção do grau de Bacharel em Sistemas de Informação.

COMISSÃO EXAMINADORA

Prof. Msc. Ailton Sergio Bonifacio
Faculdades Luiz Meneghel

Prof. Dra. Marilia Abrahão Amaral
Faculdades Luiz Meneghel

Prof. Msc. Glauco Carlos Silva
Faculdades Luiz Meneghel

Bandeirantes, 10 de Junho de 2009

Á minha família e namorada por toda motivação e amor demonstrado. Á vocês esta vitória é dedicada.

AGRADECIMENTOS

Primeiramente a Deus que me iluminou e deu forças para que eu confeccionasse esse trabalho.

Ao Prof. Msc. Orientador Ailton, que durante esse período me orientou com muita paciência e compreensão, e por proporcionar uma amizade que guardarei para a toda a vida.

A minha família, principalmente a José e Samira, meu pais, que em momento algum descreditaram em mim e me deram toda a força e motivação, em forma de amor e carinho.

A minha namorada, Carolina, que sempre esteve ao meu lado quando precisei, me apoiando e fazendo com que acreditasse mais em mim.

Aos amigos e colegas, pela força e pela vibração em relação a esta jornada.

Aos professores e colegas de Curso, pois juntos trilhamos uma etapa importante de nossas vidas.

Aos proprietários da Software House Cyprux, que abriram todas as portas para que conclui-se este trabalho.

A todos que, com boa intenção, colaboraram para a realização e finalização deste trabalho.

Aos que não impediram a finalização deste estudo.

“Se existe uma forma de fazer melhor,
descubra-a.”

Thomas Edison

RESUMO

O presente trabalho aborda o desenvolvimento de um sistema de gerenciamento empresarial para uma pneumática. Como características do mesmo têm-se o cadastro de clientes, contatos, vendedores, funcionários, contas à pagar, contas à receber, entre outros. O foco principal do sistema é o controle estoque, uma vez que o estoque é formado por produtos que montam kits específicos, e um controle total de toda a negociação com o cliente, desde o primeiro ao último contato. Para tal desenvolvimento foi empregado o uso de algumas tecnologias. Conceitos relacionados a estas tecnologias foram descritos de forma a comprovar sua usabilidade, e mostrar suas vantagens. O estudo teve por base o entendimento de conceitos de orientação objeto, conceitos de arquitetura de Mapeamento Objeto - Relacional apresentados através da ferramenta MyGeneration, sintaxe da linguagem de programação C# e da ferramenta de gerenciamento de banco de dados Microsoft SQL Server 2005. Ao final deste estudo, almejou-se ter um software estável e seguro para que supra todas essas dificuldades da empresa.

Palavras-Chave: Estoque, tecnologias, cliente, negociação.

ABSTRACT

This paper addresses the development of a business management system for a pneumatic. The same characteristics as has been the register of clients, contacts, vendors, employees, accounts payable, accounts receivable, among others. The main focus of the system is to control inventory, since the stock is forming for specific products that assemble kits, and a total control of all the negotiation with the customer, from first to last contact. Was used for this development the use of some technologies. Concepts related to these technologies were described in order to prove its usability, and show its advantages. The study was based on the understanding of concepts of object orientation, concepts of architecture Object - Relational Mapping) tool provided by MyGeneration, syntax of C # programming language and tool for management of database Microsoft SQL Server 2005. At the end of the study, aims to have a stable and secure software that above all these difficulties the company.

Keywords: Stock, technologies, client, negotiation.

LISTA DE FIGURAS

Figura 01: MyGeneration.	20
Figura 02: Ferramenta C1Report.....	28
Figura 03: Diagrama de Classes - Produto	33
Figura 04: Diagrama de Classes – Negociação com o Cliente.....	34
Figura 05: Diagrama de Classes - Controle Financeiro e de Estoque	35
Figura 06: Diagrama de Caso de Uso - Produtos	36
Figura 07: Diagrama de Caso de Uso – Contato Cliente e Controle Financeiro	38
Figura 08: DER - Produtos.....	40
Figura 09: DER – Negociação Cliente.....	41
Figura 10: DER – Controle Financeiro e Gerenciamento de Estoque.....	42
Figura 11: Tela de Login	43
Figura 12: Tela principal	44
Figura 13: Tela Menu Cliente.....	44
Figura 14: Tela de Cadastro do Cliente – Aba Registro.....	45
Figura 15: Tela de Cadastro do Cliente – Aba Busca.....	45
Figura 16: Tela Menu Produto	46
Figura 17: Telas para manipular os Produtos.....	47
Figura 18: Tela Menu Contas a Pagar Receber.....	48
Figura 19: Tela Gerenciar Compra.....	49
Figura 20: Tela Gerenciar Cotação	50
Figura 21: Tela Gerenciar Compra – Gera AF	50
Figura 22: Tela Gerenciar Autorização Fornecimento	51
Figura 23: Tela Gerenciar Ordem de Serviço.....	52
Figura 24: Tela Menu Lista de Oportunidades.....	52
Figura 25: Tela Gerenciar Oportunidade	53
Figura 26: Tela Gerenciar Oportunidade – Gera Cotação.....	53
Figura 27: Tela para geração de relatório de Oportunidades.....	54
Figura 28: Tela de relatório de Oportunidades	54
Figura 29: Tela de configuração do MyGeneration	56
Figura 30: Tela de configuração do MyGeneration	57
Figura 31: Tela de configuração do MyGeneration	58
Figura 32: Tela de configuração do MyGeneration	59

LISTA DE QUADROS

Quadro 1: Exemplo de Código MyGeneration	21
Quadro 2: Exemplo de Código .NET Framework.....	21
Quadro 3: Exemplo de Aplicação MyGeneration.....	60

LISTA DE SIGLAS

AF	Autorização de Fornecimento
BCL	Base Class Library
BI	Business Intelligence
CASE	Computer-Aided Software Engineering
COM	Component Object Model
CLR	Common Language Runtime
CRUD	Create, Review, Update, Delete
C#	C Sharp
C++	C Plus Plus
DER	Diagrama de Entidade e Relacionamento
HTML	HyperText Markup Language
JUDE	Java and UML Developer Environment
J#	Jey Sharp
MSIL	Microsoft Intermediate Language
OO	Orientação a Objeto
ORM	Object relational mapping
OS	Ordem de Serviço
POO	Programação Orientação a Objeto
UML	Unified Modeling Language
VB	Visual Basic
XML	Extensible Markup Language

SUMÁRIO

1 INTRODUÇÃO	13
2 OBJETIVOS.....	14
2.1 Objetivos Gerais.....	14
2.2 Objetivos Específicos	14
3 JUSTIFICATIVA.....	15
4 FUNDAMENTAÇÃO TEÓRICA	16
4.1 Orientação a Objetos	16
4.1.1 Classe.....	17
4.1.2 Objeto	17
4.1.3 Encapsulamento.....	18
4.1.4 Herança	18
4.1.5 Polimorfismo	19
4.2 ORM (Object relational mapping).....	19
4.2.1 MyGeneration	20
4.3 FRAMEWORK.NET	22
4.4 C# (C Sharp).....	25
4.5 Microsoft SQL SERVER 2005	26
4.6 Microsoft Visual Studio 2005.....	27
4.7 ComponentOne Studio Enterprise.....	28
5 SISTEMA PROPOSTO	29
5.1 Levantamento de Requisitos.....	30
5.2 Ferramentas utilizadas.....	32
5.3 Modelagens da ferramenta	33
5.3.1 Diagramas de Classes	33
5.3.2 Diagrama de Casos de Uso	34
5.3.4 Diagrama de Entidade Relacionamento	40
5.4 Implementação do Sistema	43
5.4.1 Interface Gráfica	43

5.5 Aplicação MyGeneration.....	55
5.5.1 Instalação.....	55
5.5.2 Configuração.....	56
5.5.3 Utilização.....	57
5.5.4 Aplicação.....	60
6 CONSIDERAÇÕES FINAIS.....	62
REFERÊNCIAS.....	62

1 INTRODUÇÃO

Em se tratando do mundo moderno, onde a tecnologia tem avançado de modo fantástico, e as empresas cada vez mais pensam em modernização, a informatização significa um enorme avanço para as empresas de modo geral, principalmente nos países desenvolvidos e em desenvolvimento. O processo de informatização das empresas é imprescindível para o bom andamento de todos os outros processos dentro das organizações. (Revista Pequenas Empresas Grandes Negócios, 2006)

No entanto, cada empresa possui uma metodologia de trabalho direcionada, onde suas particularidades devem ser tratadas caso a caso, por isso muitas empresas que trabalham com sistemas de gestão empresarial possuem deficiências no tratamento dessas particularidades, (Revista Pequenas Empresas Grandes Negócios, 2006) e por isso o objetivo desse projeto é desenvolver um sistema que trabalhe com um método de trabalho simples no que se diz respeito a seu controle financeiro, controlando toda a entrada e saída de capital, de acordo com suas ações, controle de estoque, uma vez que o estoque é formado por produtos que montam kits específicos, pedidos pelos clientes, e por ultimo um modulo de atendimento ao cliente, que controla toda a negociação feita com o cliente do começo ao fim, desde seu primeiro contato com o vendedor, até uma geração de orçamento, pedido, gerenciamento de estoque, dentre todas as ações que serão desenvolvidas no sistema, auxiliando assim, tomadas de decisões futuras, e o próprio desenvolvimento da empresa.

Baseado em todo o conhecimento adquirido ao decorrer do curso, pretende-se então, desenvolver um sistema que responda a todos os requisitos necessários, através de tecnologias atuais como a plataforma .NET, juntamente com a linguagem de programação C# (*C Sharp*), que se incluem como objetos de estudo desse trabalho.

2 OBJETIVOS

2.1 *Objetivos Gerais*

O objetivo geral deste trabalho é o desenvolvimento de uma aplicação desktop para gerenciamento de uma empresa de pneumática, composto por cadastro de clientes, contatos, vendedores, funcionários, contas a pagar, contas a receber, entre outros. Um controle de atendimento ao cliente, tratando do primeiro ao último contato entre vendedor e cliente, e um controle de estoque direcionado a realidade de uma empresa que atua no ramo de pneumática. Com isso, o usuário terá maior manuseio da movimentação dos produtos de sua empresa e de como seus vendedores estão atuando, podendo assim tomar decisões futuras, como compra e venda de produtos, com maior facilidade e segurança, garantindo assim maior credibilidade com o cliente.

2.2 *Objetivos Específicos*

Os objetivos específicos deste projeto são:

- Utilização das técnicas de programação orientada a objeto.
- Implementação baseada em soluções ORM (*Object relational mapping*), utilizando o gerador de classes *MyGeneration*.
- Utilização da tecnologia .Net juntamente com a linguagem C#.
- Utilização da ferramenta CASE: JUDE.
- Utilização do SGBD SQL SERVER 2005.

3 JUSTIFICATIVA

Cada empresa possui uma metodologia de trabalho direcionada, onde suas particularidades devem ser tratadas caso a caso, por isso muitas empresas que trabalham com sistemas de gestão empresarial possuem deficiências no tratamento dessas particularidades, a informatização anda de mãos dadas com a pesquisa científica.

Pretende-se desenvolver um sistema que utilize um método de trabalho específico no que se diz respeito a controlar toda a entrada e saída de capital da empresa de acordo com suas ações, controle de estoque, uma vez que o estoque é formado por produtos que montam kits específicos, pedidos pelos clientes, e por último um controle total de atendimento ao cliente, que controla toda a negociação feita com o cliente do começo ao fim, desde seu primeiro contato com o vendedor, até uma geração de orçamento, pedido, gerenciamento de estoque, dentre todas as ações que serão desenvolvidas no sistema.

A utilização deste sistema, informando todas as negociações de uma maneira detalhada e com um controle de estoque eficaz, permite uma maior organização e facilidade para toma de decisões futuras. Satisfazendo cada vez mais o cliente e assim aumento cada vez mais o desenvolvimento da empresa.

4 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo serão relatados conceitos relacionados à área de pesquisa de projeto.

4.1 Orientação a Objetos

O termo orientação a objetos sugere uma associação entre (abstração de) coisas do mundo real e trechos de programas de computador ou objetos (Yourdon, 1999). A POO (Programação Orientada a Objetos) introduz uma abordagem na qual o desenvolvedor visualiza seu programa em execução como uma coleção de objetos cooperantes que se comunicam por meio de troca de mensagens (Vincenzi, 2004).

A OO (Orientação a Objetos) é um paradigma que ajuda os projetistas a desenvolverem sistemas com estruturas simples (mesmo em sistemas maiores) e descentralizadas (Meyer, 2000).

Embora este conceito exista há décadas, as técnicas de OO permitem reutilizar mais do que simplesmente o código, podendo fazer uso de requisitos, análise, projeto, planejamento de testes, interfaces de usuários e arquiteturas. Assim, praticamente todos os componentes do ciclo de vida da engenharia de software podem ser encapsulados como objetos reusáveis (Yourdon, 1999).

Alguns benefícios da OO estão apresentados abaixo (Yourdon, 1999):

- **Reaproveitamento:** as classes são projetadas de forma que possam ser reutilizadas em vários sistemas.
- **Estabilidade:** classes projetadas para reutilização tornam-se estáveis pois são testadas e aperfeiçoadas para várias situações.

- **Projetistas pensam no comportamento dos objetos e não nos detalhes de baixo-nível:** o encapsulamento oculta os detalhes e faz com que as classes se tornem caixas-pretas, onde somente se precisa compreender seu comportamento e como se comunicar com elas.
- **Desenvolvimento Acelerado:** os aplicativos são criados com componentes pré existentes que se adaptam a um projeto em particular.
- **Bibliotecas de classes corporativas:** as empresas podem desenvolver bibliotecas de classes próprias, que refletem os padrões internos da organização e as necessidades de suas aplicações.

Estes benefícios proporcionaram o início da reutilização de soluções no desenvolvimento de software.

4.1.1 Classe

Classes são uma descrição de um grupo de objetos com propriedades (atributos), comportamentos (operações), relacionamentos com outros objetos e semântica comum. Assim, uma classe é um gabarito para criar objetos, onde cada objeto é uma cópia de alguma classe (Terry, 2001).

4.1.2 Objeto

Objeto é uma instância de uma classe criada em tempo de execução. Cada objeto tem uma cópia dos dados definidos na classe e encapsula estado e comportamento. Os objetos interagem entre si e são ativados por meio de troca de mensagens (Vincenzi, 2004).

4.1.3 Encapsulamento

Encapsulamento é uma técnica empregada para garantir a ocultação de informações na qual a interface e a implementação de uma classe são separadas sintaticamente. Somente os métodos pertencentes ao objeto podem ter acesso aos dados encapsulados. O encapsulamento estimula a modularidade do programa e restringe possíveis interdependências com outras classes, exceto por meio de sua interface (Vincenzi, 2004).

Dessa forma, o encapsulamento evita a interdependência de um programa, fazendo com que as futuras alterações não causem grandes efeitos colaterais (ARAÚJO, 2009).

4.1.4 Herança

Herança é o mecanismo de reutilização de atributos e operações, definidos em classes gerais, por classes mais específicas, podendo ser usada para expressar tanto generalização como associação (Furlan, 1998).

Com a herança é possível trabalhar com os conceitos de superclasse e subclasse. Superclasse, chamada também de classe mãe, consistem em classes que possuem classes derivadas, já as subclasses, chamadas de classes filha, são classes derivadas. As subclasses, por ser derivada a superclasse, herdam todas suas características (GUEDES, 2008).

Apresenta como vantagem o reaproveitamento de código, isto significa que, uma subclasse herda automaticamente os atributos e métodos de sua superclasse, não sendo necessário redeclará-los, conseqüentemente permitindo a reutilização do código pronto (ARAÚJO, 2009; GUEDES, 2008).

4.1.5 Polimorfismo

Polimorfismo é qualidade ou estado de um objeto ser capaz de assumir diferentes formas. Possibilita que ao enviar uma mesma mensagem para um conjunto de objetos, cada objeto responda de maneira diferente em função da mensagem recebida (Vincenzi, 2004).

4.2 ORM (*Object relational mapping*)

Desenvolvedores de aplicações orientadas a objetos sabem que de alguma maneira precisam armazenar e recuperar informações em bancos de dados relacionais. Um ORM, nada mais é do que um *framework* ou um conjunto de classes que permite fazer este trabalho sem precisar escrever códigos de conexão com o banco, *querys* de SQL a todo o momento, preservando as características de orientação a objetos da linguagem face à natureza relacional dos bancos de dados atuais (Revista .Net Magazine, 2007).

As soluções ORM podem ser divididas em quatro níveis de maturidade: relacional puro (*pure relational*), mapeamento leve (*light object mapping*), mapeamento médio (*medium object mapping*) e mapeamento completo (*full object mapping*) (Fussel, 1997).

Para o sistema proposto foram utilizadas soluções relacionais puras, a aplicação inteira, incluindo a interface com o usuário, é afetada pelo modelo relacional e por operações envolvendo acesso direto à base de dados relacional (Alur et al., 2001).

4.2.1 MyGeneration

O MyGeneration é uma ferramenta para geração de código fonte a partir de *templates*, com suporte a programação C#, VB .NET JScript e VBScript. Pode ser usado em arquiteturas ORM e possui suporte a diversos bancos de dados: SQL Server , Oracle, Access, MySQL , FireBird, etc (MACORATTI, 2008).

Os *templates* podem ser vistos como scripts que podem ser definidos a partir de modelos prontos, ou podem ser criados desde o início, podendo gerar *Stored Procedures* , XML, HTML (HyperText Markup Language), páginas ASP .NET, entidades , DAL, etc (MACORATTI, 2008).

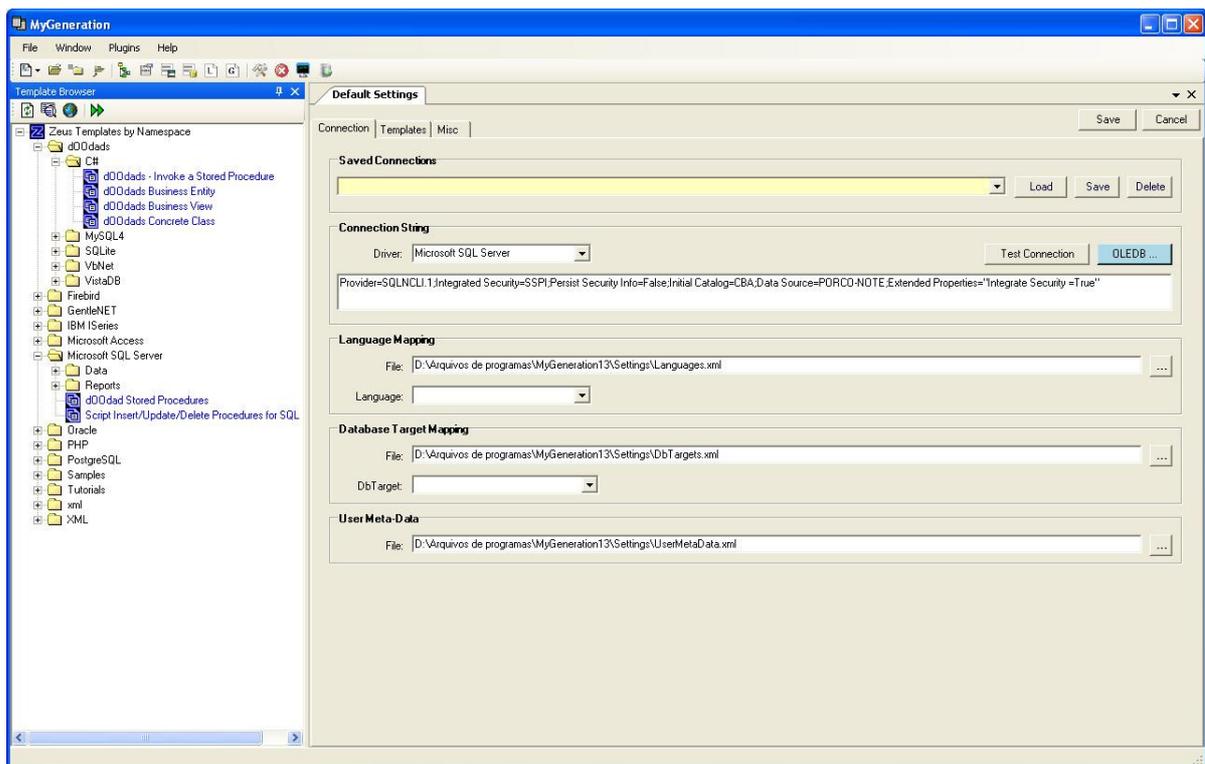


Figura 01: MyGeneration.

Fonte: Macoratti (2002)

Algumas vantagens para a utilização do MyGeneration (Revista .Net Magazine, 2007):

- Escrever menos código e programar com muito mais produtividade.
- Código fica mais elegante.
- Melhor manutenção do projeto.
- Melhora a padronização da aplicação.

Exemplo de um código em .NET usando MyGeneration:

```
Usuario.AddNew();
Usuario.FirstName = this.txtFirstName.Text;
Usuario.LastName = this.txtLastName.Text;
Usuario.Save();
```

Quadro 1: Exemplo de Código MyGeneration
Fonte: Revista .Net Magazine, (2007)

Exemplo de um código em .NET usando classes nativas do .NET Framework:

```
strConn = "string de conexão aqui";
SqlConnection objCon = new SqlConnection(strConn);
SqlCommand objComand = objCon.CreateCommand();
objComand.CommandText = "insert into Usuario (FirstName , LastName ) values (@FirstName , @LastName )";
objComand.CommandType = CommandType.Text;
objComand.Parameters.Add("@FirstName ", txtFirstName.Text );
objComand.Parameters.Add("@LastName ", txtLastName.Text );
objComand.ExecuteNonQuery();
```

Quadro 2: Exemplo de Código .NET Framework
Fonte: Revista .Net Magazine, (2007)

4.3 FRAMEWORK.NET

O .Net é uma iniciativa da Microsoft que visa uma plataforma única para desenvolvimento e execução de sistemas e aplicações. Todo e qualquer código gerado para .NET, pode ser executado em qualquer dispositivo ou plataforma que possua um framework: a "Plataforma .NET" (.NET Framework). Com idéia semelhante à plataforma Java, o programador deixa de escrever código para um sistema ou dispositivo específico, e passa a escrever para a plataforma .NET (MIRANDA, 2008).

O Lançamento .NET para o mundo foi realizado em 2000 durante o PDC (Professional Developer Professional) e a sua principal inovação e a integração com o XML (eXtensible Markup Language) o que demonstra o claro direcionamento desta tecnologia para o desenvolvimento focado no ambiente Web, pois possibilita facilmente o compartilhamento do dados neste ambiente (UCHÔA, 2006).

A plataforma .NET foi desenvolvida pela Microsoft durante 4 anos e seu lançamento oficial ocorreu em junho de 2000, a estratégia desta plataforma era de explorar ao máximo a integração de serviços através da Web, fazendo que tanto aplicações Web ou Desktop se comuniquem de uma forma simples, independente do sistema operacional e linguagem de programação.

Um diferencial significativo que a plataforma .NET oferece é o suporte a uma série de linguagens de programação, com isso uma mesma solução pode ser desenvolvida em mais de uma linguagem de programação ou até mesmo utilizar uma solução existente e integrá-la em um novo projeto, economizando em retrabalho.

Nos itens a seguir são apresentados os componentes da plataforma .NET.

- **.NET Framework:** Um framework é uma estrutura pré-definida que contém um conjunto de componentes de software oferecendo uma estrutura padrão para o desenvolvimento de uma aplicação. O .NET Framework oferece suporte a mais de 20 linguagens de programação, independentemente da linguagem escolhida, sendo ela suportada por este framework, o resultado da aplicação será o mesmo (THAI, 2001).
- **CLR (*Common Language Runtime*):** O CLR é o ambiente que faz a execução das aplicações desenvolvidas dentro do .NET Framework, é a máquina virtual similar ao JVM (*Java Virtual Machine*), ou seja, está entre a aplicação desenvolvida e o sistema operacional. Apesar de estas aplicações executarem no ambiente Windows, elas não são aplicações Win32, assim que uma aplicação é executada o Win32 chama o *runtime* .NET fazendo com que ele assuma o controle da aplicação executada podendo gerenciar a memória, diminuindo a chance de ocorrer erros durante sua execução. O CLR é o ambiente de execução da plataforma .NET, portanto independente da linguagem escolhida ele é o mesmo (THAI, 2001).
- **BCL (*Base Class Library*):** A BCL é a biblioteca de classes da plataforma .NET, nela estão as classes básicas para que uma aplicação possa ser desenvolvida. O uso dela facilita o processo de desenvolvimento, pois existe a possibilidade da reutilização de componentes de software, simplificando e agilizando toda a elaboração de uma aplicação. Nesta biblioteca é possível encontrar classes referentes a interfaces, manipulação de arquivos, gerenciamento de memória, funcionalidades de web, rede entre outras. As classes estão organizadas de forma hierárquica dentro de uma estrutura chamada *namespace* (THAI, 2001).

- **.NET Compact Framework:** É a materialização do .NET Framework, específico, para dispositivos móveis. Devido às diferenças de hardware entre um computador e um dispositivo móvel, foi necessário criar um framework exclusivamente para este segmento de dispositivos. O funcionamento da CLR e BCL, que são os principais componentes do framework, seguem os mesmos princípios, porém na BCL existem classes que foram criadas especificamente para dispositivos móveis, assim como outras foram retiradas por fugirem do contexto destes dispositivos (YANG, 2007).
- **MSIL(*Microsoft Intermediate Language*):** é um recurso que age independente da CPU e possibilita que na hora da execução gera um novo compilador (*Just-in-time compiler* ou seja compilação em tempo real) e ele converte para código nativo, ou seja, um código específico para o processador da máquina onde a aplicação será executada (UCHÔA, 2006).

4.4 C# (C Sharp)

Utilizado atualmente por centenas de milhares de programadores, o projeto C# foi iniciado há cerca de cinco anos (HEJLSBERG, 2004), através da Microsoft, mais especificamente por Anders Hejlsberg (mesmo criador do Turbo Pascal e do Delphi) (CORRÊA, 2005).

C# é uma linguagem orientada a objeto moderna que habilita os programadores a construir rápida e facilmente soluções para a plataforma Microsoft .NET (Camara, 2006).

Abaixo seguem algumas vantagens de se utilizar a linguagem C# (Camara, 2006):

- Utiliza padrões emergentes da programação web, como HTML e XML
- Elimina erros de alto custo na programação, por exemplo, libera o programador do gerenciamento de memória manual;
- Possui mapeamento entre processos de negócio e implementação;
- Possui interoperabilidade extensiva , incluindo suporte nativo para o COM (*Component Object Model*) , permite o uso restrito de ponteiros nativos.

A linguagem C# é definida como uma linguagem de programação orientada a objetos, simples, moderna e com tipificação segura, derivada da C e C++. Foi desenvolvida especificamente para a plataforma .NET, e facilita aos desenvolvedores criar, depurar e distribuir aplicações corporativas (Turtschi et AL, 2002).

4.5 Microsoft SQL SERVER 2005

Segundo a Microsoft o SQL Server 2005 é uma plataforma abrangente de banco de dados que fornece recursos de gerenciamento de dados de classe empresarial com ferramentas de BI (*Business Intelligence*) integradas (MICROSOFT ®, 2009).

É uma ferramenta extremamente complexa que traz grandes vantagens, principalmente nas ferramentas de Administração e Desenvolvimento (SILVIO, 2007).

Um dos grandes atrativos dessa versão é a forte integração com o CLR que é o ambiente gerenciado da plataforma .NET e inclui uma série de recursos para gerenciamento de recursos de sua aplicação (SILVIO, 2007).

Essa solução de análise de gerenciamento de dados de última geração fornece mais segurança, escalabilidade e disponibilidade de dados empresariais e aplicativos analíticos, tornando-os mais fáceis de criar, implementar, gerenciar e são mais confiáveis e seguros (SILVIO, 2007).

São compartilhados dados entre múltiplas plataformas, aplicações e dispositivos para facilitar a conexão entre sistemas internos e externos, integrando soluções robustas de inteligência empresarial BI, que ajudam na tomada de decisões e aumentam a produtividade em toda a empresa; controlando custos sem sacrificar o desempenho, a disponibilidade e a escalabilidade ou a segurança (SILVIO, 2007).

Ele também inova ao oferecer preço e licenciamento acessível. O SQL Server 2005 faz a sua infra-estrutura de dados avançar em 3 (Três) principais areas que são: Gerenciamento de Dados Corporativos, Produtividade do Desenvolvedor e Inteligência Empresarial (SILVIO, 2007).

A total integração com o Microsoft Visual Studio, o Microsoft Office System e um conjunto de novas ferramentas de desenvolvimento, incluindo o *Business Intelligence Development Studio*, torna o SQL Server 2005 único. Seja você um desenvolvedor, administrador de banco de dados, operador de informações ou tomador de decisões, o SQL Server 2005 fornece soluções inovadoras que podem ajudá-lo a obter mais valor de seus dados (MICROSOFT®, 2009).

4.6 Microsoft Visual Studio 2005

O Microsoft Visual Studio é um pacote de programas da Microsoft, para desenvolvimento de Software, especialmente dedicado, ao *framework* .NET e às linguagens VB (Visual Basic), C, C++ (C Plus Plus), C# e J# (Jey Sharp). Também é um grande produto de desenvolvimento na área web, usando a plataforma do ASP.NET. As linguagens com maior freqüência nessa plataforma são: VB.NET (Visual Basic .Net) e o C# (lê-se C Sharp). (Wikipédia, 2009)

O Visual Studio 2005 é uma ferramenta de ambiente de desenvolvimento integrado (IDE) com o intuito de ajudar na criação, documentação, execução e depuração de programas implementados em varias linguagens do .NET (DEITEL, 2003).

4.7 ComponentOne Studio Enterprise

ComponentOne Studio Enterprise é um dos maiores e mais completos *toolset* de componentes para o desenvolvimento no Windows, Web e aplicações Móveis. ComponentOne Studio acelera o desenvolvimento através de recursos visual com recursos *drag and drop* que simplificam o tempo e aumenta os recursos do Visual Studio. Podem-se adicionar os mais sofisticados recursos para o Windows, Web, e aplicações Móveis com um número menor de códigos e com um tempo menor. O ComponentOne apresenta melhorias para as suas aplicações, focando em simplicidade, economia de investimento e recursos novos e adicionais para o Microsoft Visual Studio. O *Studio Enterprise* fornece componentes para quatro subscrições, disponibilizando mais de 180 componentes para .NET, ASP.NET, ActiveX, e Dispositivos Mobile. (Stock Soft, 2009)

O ComponentOne oferece todas as ferramentas necessárias para satisfazer o seu relato, a impressão, visualização, e exportação de arquivos. Criar relatórios através de sua base de dados se torna rápido e fácil usando o componente C1Report. Visualizar, exportar (como PDF, HTML, Excel) ou imprimir seus relatórios é simples e de fácil manuseio, pois o ComponentOne adiciona vários tipos de controles na ferramenta de desenvolvimento Visual Studio. (ComponentOne, 2009)

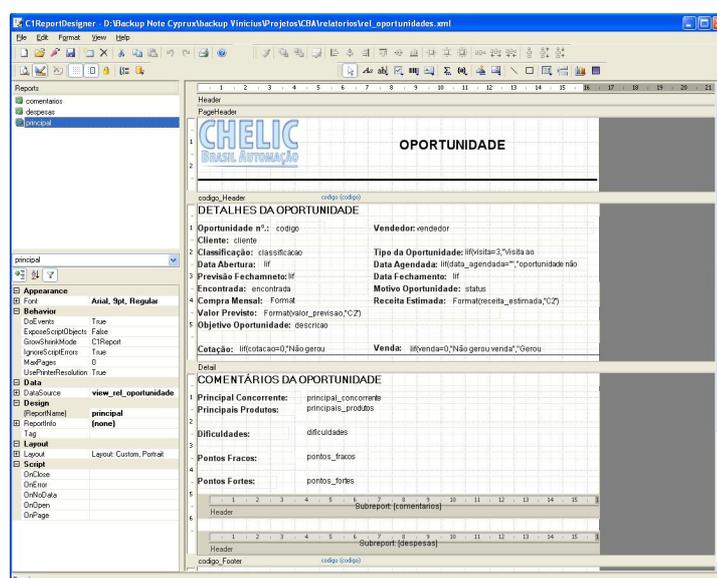


Figura 02: Ferramenta C1Report

5 SISTEMA PROPOSTO

Para melhor comprovar o uso e a eficácia do estudo, um software foi desenvolvido visando um controle de estoque funcional e objetivo para a realidade da empresa e uma maior eficácia na parte de negociação da empresa, facilitando assim, tomadas de decisão que contribuam para a satisfação do cliente. Tal ferramenta contém cadastros em geral para que o usuário possa armazenar as informações conforme sua necessidade. Para uma maior visualização, o software emite diversos relatórios que informa de uma maneira simples a situação da empresa. A interação com o sistema é através de uma interface gráfica de fácil utilização que lhe proporciona uma maior comodidade e flexibilidade em seu uso.

A motivação para o desenvolvimento desta ferramenta deu-se através da facilidade no gerenciamento da empresa, com maior facilidade e segurança.

Alguns métodos de desenvolvimento foram adotados otimizando o desenvolvimento resultando em uma maior qualidade dos resultados obtidos, utilizando o modelo de processo de software baseado numa arquitetura ORM e unificação de paradigma adotando o padrão de OO desde a fase de análise até a fase final de implementação.

Softwares que se baseiam na arquitetura ORM se tornam mais simples, pois, associam todas suas tabelas do banco de dados através de classes que são utilizadas para fazer todos os tipos de operações, sem que precise escrever códigos de comunicação, tornando assim o código mais elegante e manutenível.

A notação OO foi escolhida devido a seu grande desempenho, oferecendo facilidades como reaproveitamento de código, padronização e manutenção.

Durante a fase de análise e modelagem do software foi utilizada a ferramenta case Jude Community 5.4.1, segue a notação UML (*Unified Modeling Language*), sendo modeladas as funcionalidades do sistema facilitando o processo de desenvolvimento do mesmo.

O software foi desenvolvido utilizando a linguagem C#. O desenvolvimento utilizou o modelo de programação *desktop*, sendo manipulada através do uso de métodos contidos nas APIs inclusas na plataforma .Net. Para lidar com a manipulação da informação entre sistema e a base de dados, foi utilizado o MyGeneration. Os dados foram armazenados na base de dados Microsoft SQL Server 2005.

5.1 Levantamento de Requisitos

1 Visão Geral.

O sistema deve conter um controle de estoque direcionando a realidade da empresa, de uma forma eficaz e segura. Desenvolver uma interface simples e objetiva para que seja armazenada toda a parte de negociação do cliente. Conter cadastros que ofereçam um controle simples da parte financeira da empresa.

2 Objetivos.

Facilitar o controle de estoque da empresa, tornando-o mais organizado e seguro. Armazenar todo o tipo de contato com o cliente para que possa tomar decisões que atinjam a satisfação total do cliente. Poder visualizar de forma simples a entrada e saída de dinheiro da empresa.

3 Requisitos funcionais.

- Cadastro de clientes.
- Cadastro de contatos.
- Cadastro de marca de produto.
- Cadastro de categoria de produto.
- Cadastro de hastes e camisas.
- Cadastro de produtos.
- Cadastro de preço por curso.
- Cadastro de montagem de Cilindro.
- Cadastro de contas a pagar.
- Cadastro de contas a receber.
- Cadastro de vendedor.
- Cadastro de fornecedor.
- Cadastro de usuários.
- Emissão de relatórios diversos.
- Gerenciamento de compra.
- Gerenciamento de OS.
- Gerenciamento de cotação.
- Gerenciamento de autorização de fornecimento.
- Gerenciamento de oportunidades.

5.2 Ferramentas utilizadas

A tabela 01 descreve as ferramentas necessárias com suas respectivas finalidades para o desenvolvimento deste trabalho, proporcionando concluir os objetivos propostos.

Microsoft Visual Studio	Ferramenta de desenvolvimento de aplicações (Desktop/Web).
SQL Server Management Studio Express	Ferramenta gráfica de gerenciamento para SQL Server 2005 Express Edition.
Jude Community 5. 4. 1 Community Edition	Ferramenta de diagramas UML.
Microsoft SQL Server 2005	Ferramenta de gerenciamento de dados.
MyGeneration 1.3.0.3	Ferramenta para geração de códigos.
ComponentOne	Ferramenta para desenvolvimento de relatórios.
Windows XP SP. 2	Sistema operacional base para o desenvolvimento da ferramenta.

Tabela 01: Ferramentas Utilizadas

5.3 Modelagens da ferramenta

O uso da ferramenta de modelagem representou um processo importante facilitando o desenvolvimento do sistema sendo modelados seus diagramas de casos de uso e de classe.

5.3.1 Diagramas de Classes

Com o uso da ferramenta MyGeneration todas as classes são geradas automaticamente, sendo assim representadas através dos diagramas abaixo.

A ausência de relacionamentos é pertinente a soluções de ORM da ferramenta que se utiliza de um relacionamento puro, citado anteriormente.

A Figura 03 representa as classes que envolvem o gerenciamento de produtos.

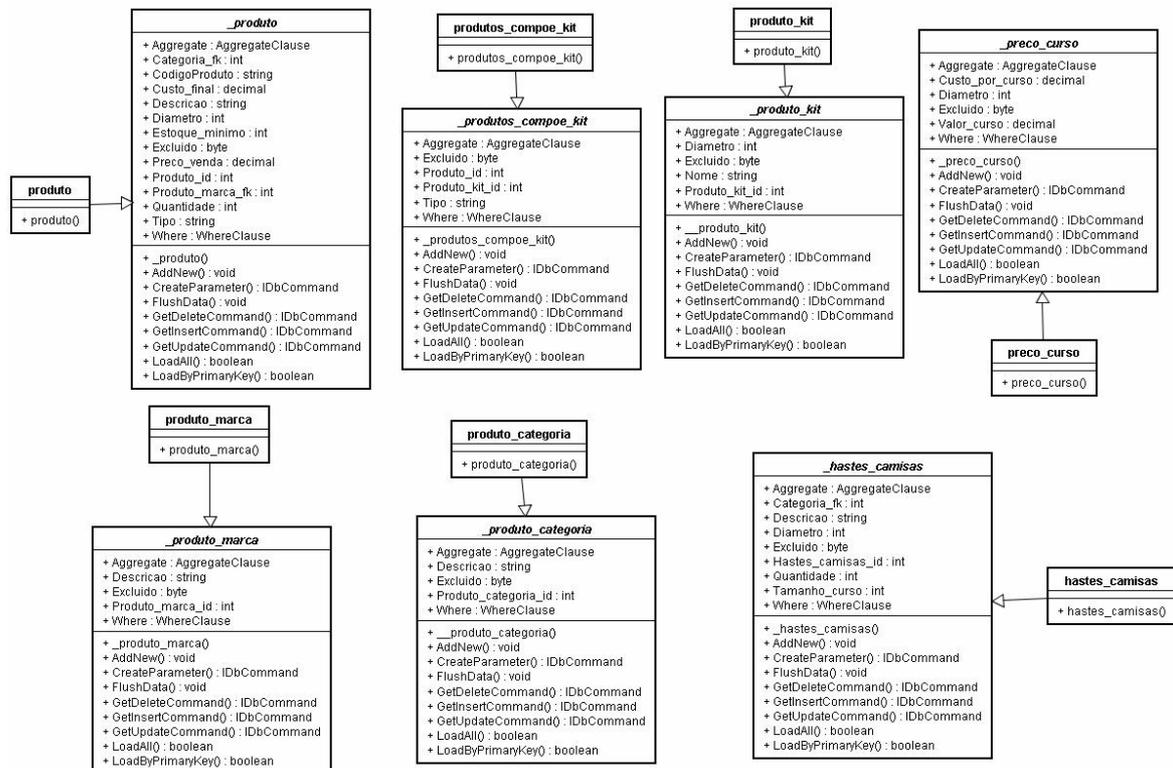


Figura 03: Diagrama de Classes - Produto

A Figura 04 representa as classes que envolvem gerenciamento da negociação com o cliente.

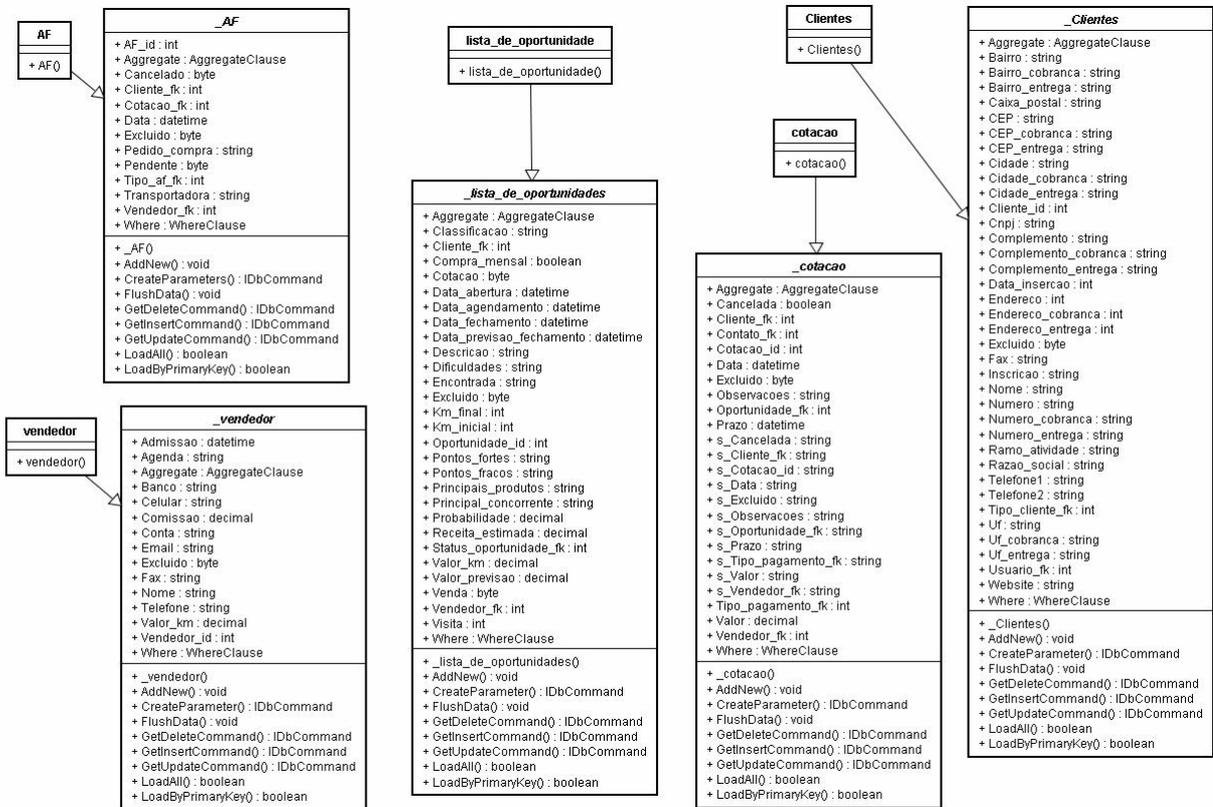


Figura 04: Diagrama de Classes – Negociação com o Cliente

A Figura 05 representa as classes que envolvem gerenciamento do controle financeiro e controle de estoque.

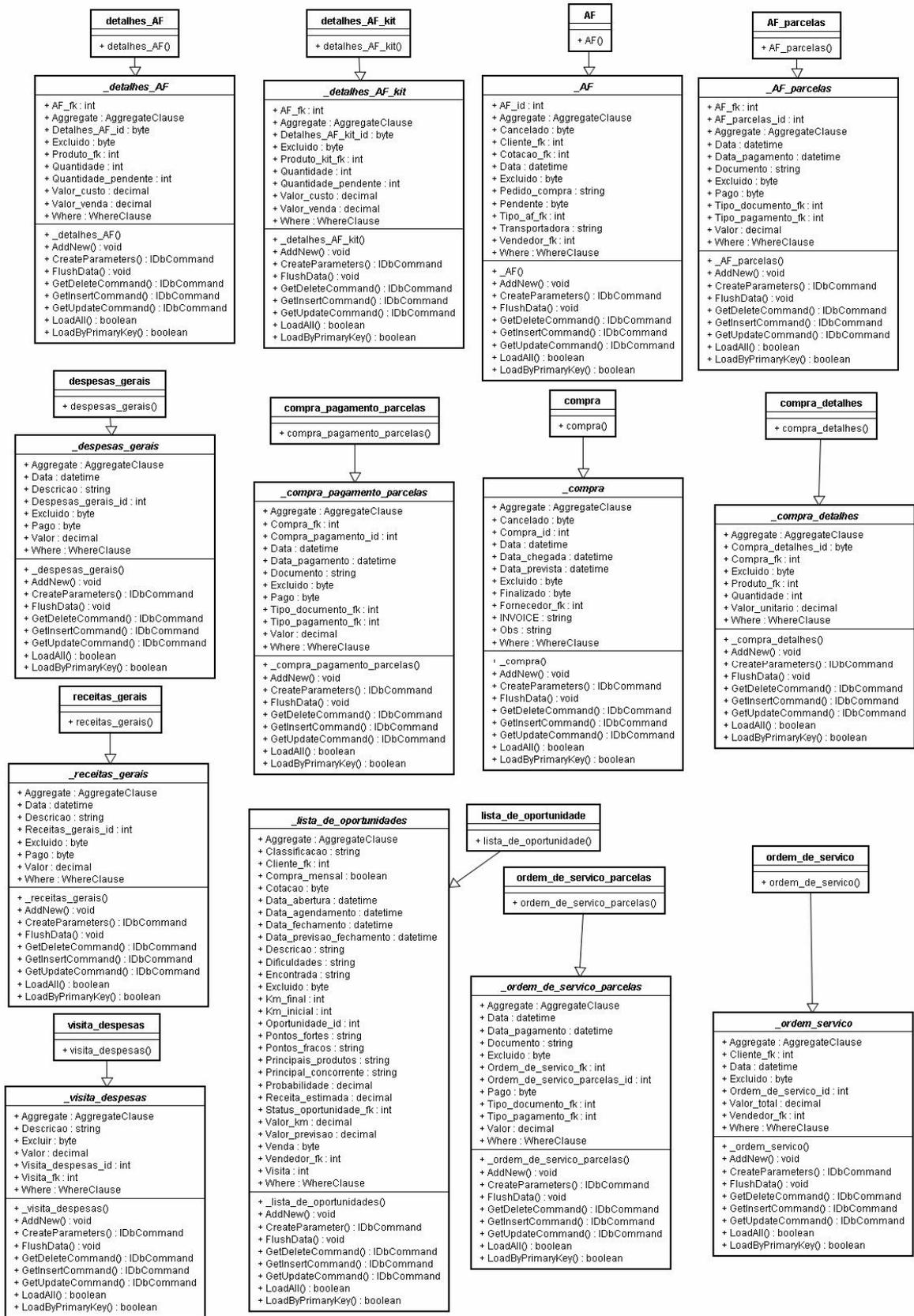


Figura 05: Diagrama de Classes - Controle Financeiro e de Estoque

5.3.2 Diagrama de Casos de Uso

Na Figura 06 o diagrama que envolve o a parte de cadastramento dos produtos e o controle de estoque:

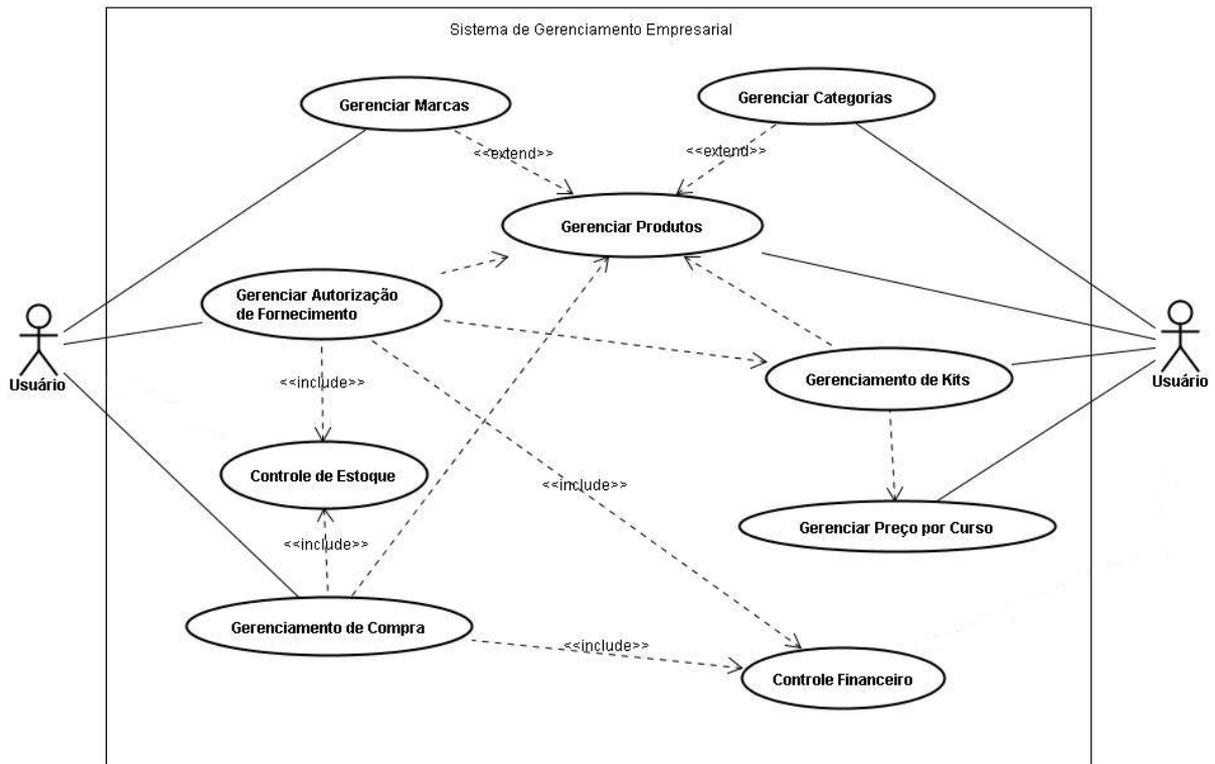


Figura 06: Diagrama de Caso de Uso - Produtos

Para melhor entendimento, será explicado o ator e seus componentes do diagrama do caso de uso exemplificado na Figura 06:

USUÁRIO: qualquer usuário que estiver no cadastrado e logado no sistema, podendo exercer as seguintes tarefas:

- **Gerenciar Marcas:** manipulação dos dados referentes às marcas dos produtos, ou seja, alterar, excluir, cadastrar e buscar. Podendo ser usado para o cadastro de Produtos.
- **Gerenciar Categorias:** manipulação dos dados referentes às categorias dos produtos, ou seja, alterar, excluir, cadastrar e buscar. Podendo ser usado para o cadastro de Produtos.

- **Gerenciar Produtos:** manipulação dos dados referentes aos produtos existentes na empresa. Além de operações básicas como cadastramento, alteração, exclusão e busca. Também pode auxiliar na conferência de produtos em estoque.
- **Gerenciar Preço por Curso:** manipulação dos preços referidos aos produtos considerados “kits” de acordo com o diâmetro deste.
- **Gerenciamento de Kits:** manipulação de produtos considerados com “kits”, ou seja, um produto que é formado por vários produtos. Além também de conter todas as operações básicas.
- **Gerenciar Autorização de Fornecimento:** trata-se da parte de venda do sistema, para quais clientes serão distribuídos os produtos, qual será a forma de pagamento, sendo assim quando concluída uma autorização de fornecimento, implica-se a atualização de estoque e da parte financeira do sistema.
- **Gerenciamento de compra:** trata-se da parte de compra de produtos, de quais fornecedores, qual a previsão de chegada, qual serão as formas de pagamento, dentro outras. Depois de finalizada implica-se a atualização do estoque e da parte financeira do sistema.
- **Controle de Estoque:** dependendo a ação feita pelo usuário o controle de estoque é acionado para atualização dos produtos da empresa, sendo este tanto pra entrada e saído dos mesmos.
- **Controle Financeiro:** dependendo a ação feita pelo usuário o controle financeiro é acionado para atualização da entrada e saída de capital dentro da empresa.

Na Figura 07 o diagrama que envolve toda a parte de contato com cliente e cadastramento financeiro:

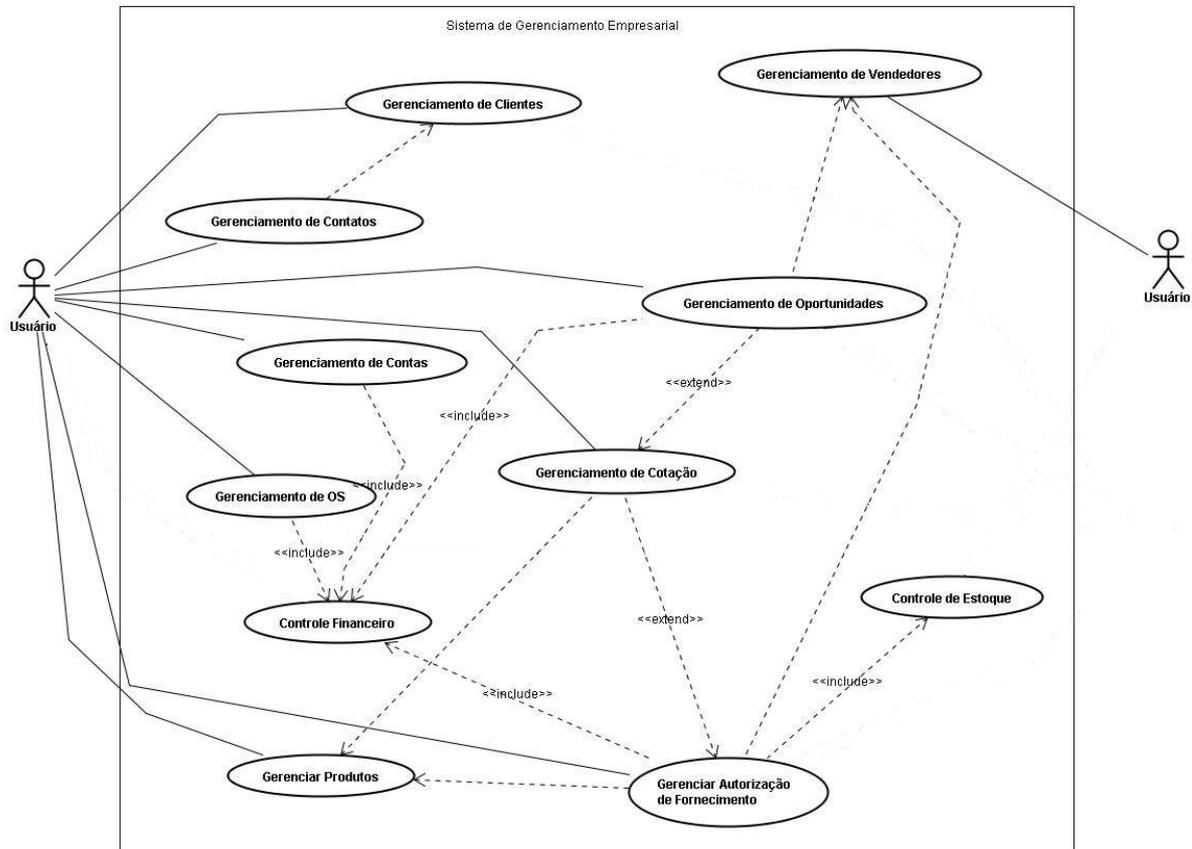


Figura 07: Diagrama de Caso de Uso – Contato Cliente e Controle Financeiro

Além das tarefas anteriores o **USUÁRIO** ainda exerce tarefas como:

- **Gerenciamento de Clientes:** manipulação dos dados referentes aos clientes existentes na empresa. Efetuando operações básicas como cadastramento, alteração, exclusão e busca.
- **Gerenciamento de Contatos:** manipulação de contatos referente aos clientes cadastrados na empresa. Além das operações básicas como cadastramento, alteração, exclusão e busca. O gerenciamento também conta com uma operação de exportação de e-mails para um arquivo separado por vírgulas (.csv), para envio de mala direta, que pode ser utilizado por alguns gerenciadores de e-mail.

- **Gerenciamento de Vendedores:** manipulação dos dados referentes aos vendedores existentes na empresa. Efetuando operações básicas como cadastramento, alteração, exclusão e busca. Também se inclui aqui operações que envolvem a parte do valor do quilometro rodado, em caso de despesas geradas a partir de oportunidades que foram feitas através de visitas.
- **Gerenciamento de Contas:** toda a manipulação da partes de contas (receitas e despesas) que não se encaixam em ordem de serviço, compra, autorização de fornecimento e despesas de visita. Exemplos: conta de água, conta de luz, etc.
- **Gerenciamento de OS:** trata-se da parte de operação que tomam conta da parte de prestação de serviços ao cliente. Quando lançada uma OS, implica-se a atualização da parte financeira do sistema.
- **Gerenciamento de Cotação:** trata-se da parte de cotação de produtos, para quais clientes, e quais produtos. Sendo assim gerado um documento que é encaminhado ao cliente pela própria empresa, em caso de aprovação pode-se torna uma autorização de fornecimento.
- **Gerenciamento da Oportunidade:** trata-se da parte que coordena o contato ou a oportunidade gerada para o cliente. Usufruindo de resultado esta se pode tornar uma cotação, até chegar a uma autorização de fornecimento.

5.3.4 Diagrama de Entidade Relacionamento

Na Figura 08 o diagrama que envolve o gerenciamento dos produtos.

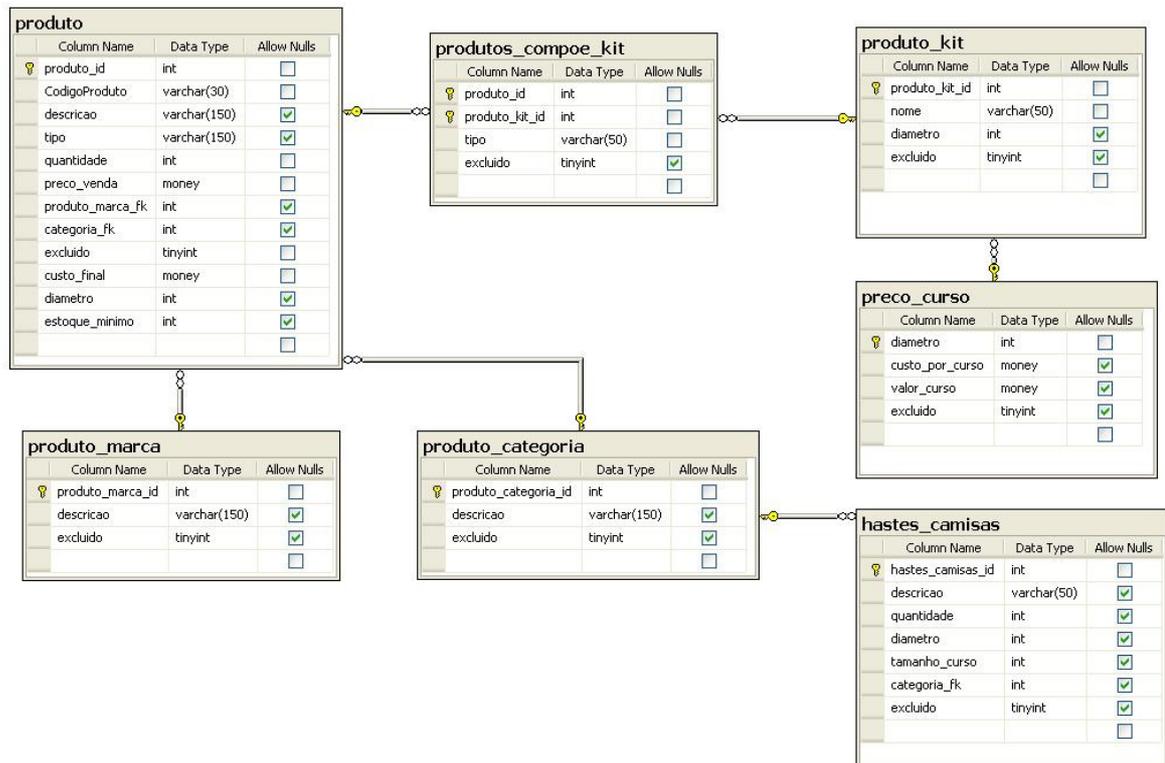


Figura 08: DER - Produtos

Na Figura 09 o diagrama que envolve o gerenciamento da negociação com o cliente.

Os relacionamentos circulares representam adaptações para necessidades exigidas pela empresa.

No caso de “AF, Clientes, cotação”, uma AF pode ser gerada através de uma cotação ou diretamente a um cliente, por isso, o campo “cotação_fk” da tabela AF pode ser nulo.

Em “lista_de_oportunidades” pode ser gerada a qualquer momento, podendo ou não gerar um cotação, assim o campo “oportunidade_fk” da tabela cotação também pode ser nulo.

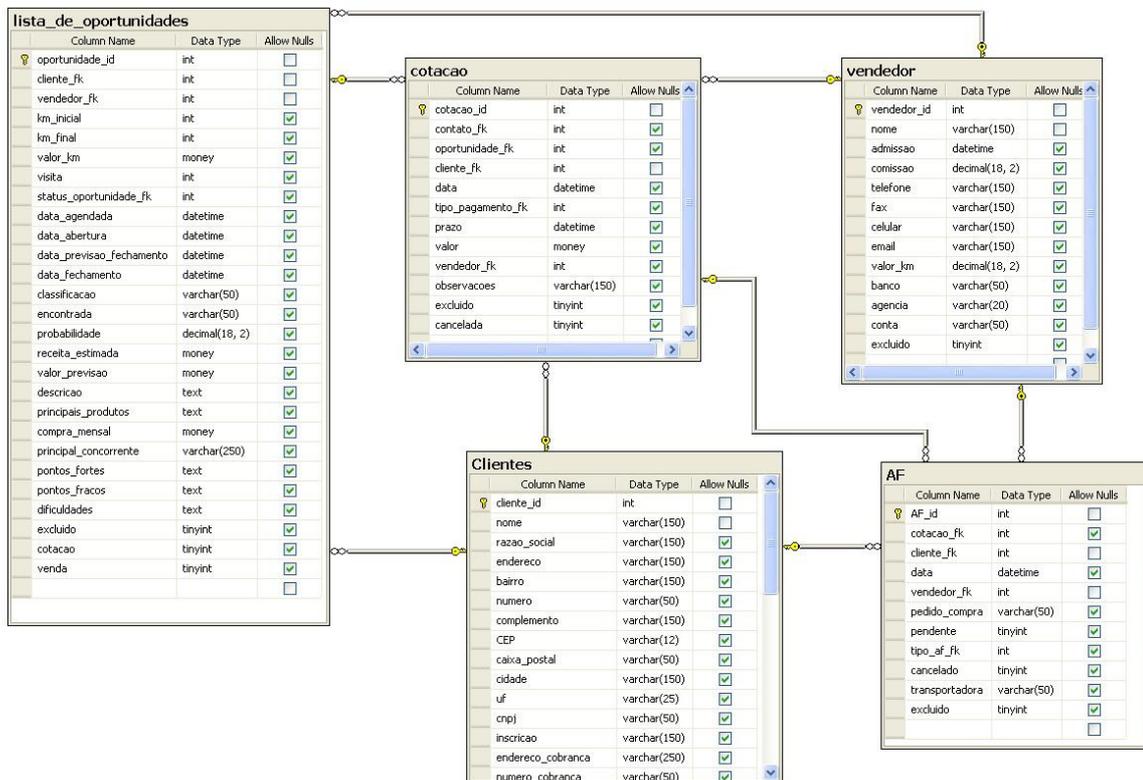


Figura 09: DER – Negociação Cliente

Na Figura 10 o diagrama que envolve o gerenciamento do controle financeiro (entrada e saída) que se utilizam das tabelas: “visita_despesas”, “AF_parcelas”, “compra_pagamento_parcelas” e “ordem_de_serviço_parcelas”.O diagrama também representa gerenciamento do estoque com a entrada e saída de produtos através da tabelas: “detalhes_AF”, “detalhes_AF_kits” e “compra_detalhes”.

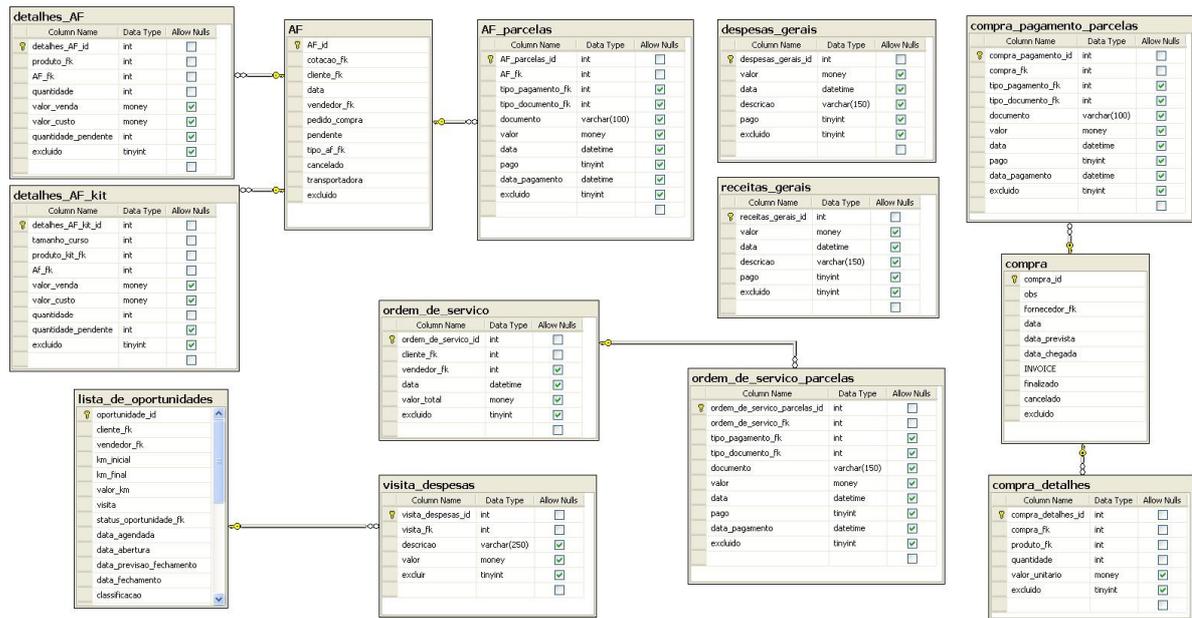


Figura 10: DER – Controle Financeiro e Gerenciamento de Estoque

5.4 Implementação do Sistema

Para justificar o estudo desenvolvido durante a fase de projeto, esta seção foi criada para representar as aplicações das metodologias descritas ao longo deste, de forma a deixar claro o processo de desenvolvimento e as funcionalidades da ferramenta proposta, atingindo o objetivo definido para este trabalho.

A implementação do sistema segue padrões da POO fazendo com que o projeto seja de fácil entendimento e manutenção. Durante a finalização de cada processo foram aplicados testes para averiguação de possíveis erros.

5.4.1 Interface Gráfica

Para proporcionar uma melhor utilização da ferramenta, optou-se por fazer a interação entre esta e o usuário através de uma interface gráfica. A Figura 11 refere-se à tela de inicial do sistema, contendo um controle de acesso para inicialização do sistema.

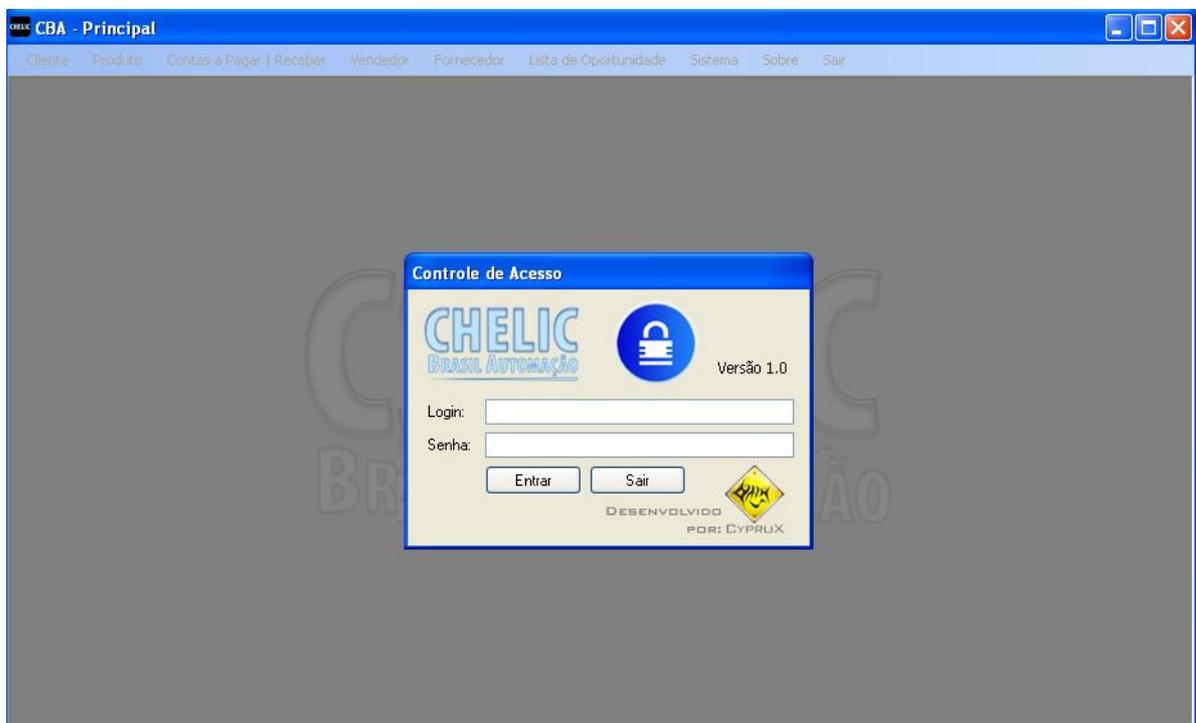


Figura 11: Tela de Login

Após o login efetuado com sucesso, o sistema apresentará a tela principal que apresenta todos os menus de funcionalidades, representado pela Figura 12.

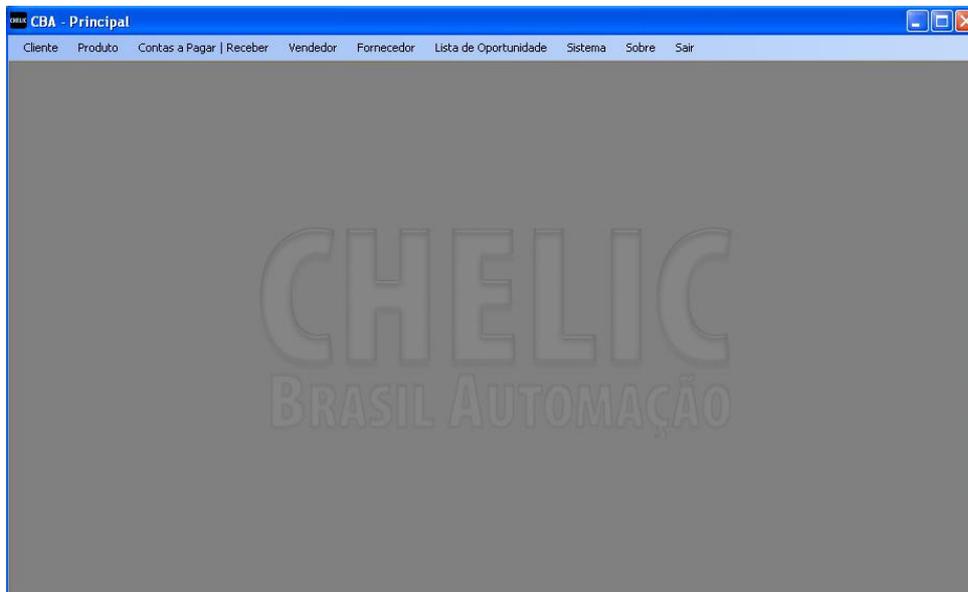


Figura 12: Tela principal

Com uma barra menu de fácil entendimento, a navegação do sistema se torna mais objetiva e clara, conforme Figura 13.

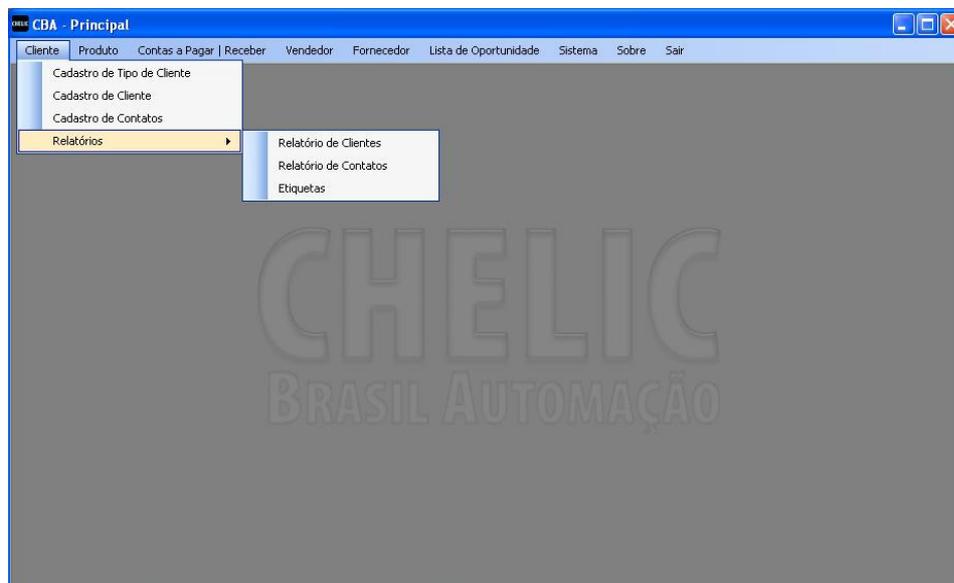


Figura 13: Tela Menu Cliente

Ao clicar em qualquer item da barra de menu, a tela correspondente será aberta dentro da tela principal. Isso proporciona uma maior organização para com o usuário, podendo ser sobrepostas várias telas.

A Figura 14 representa a tela de cadastro de clientes da empresa.

Figura 14: Tela de Cadastro do Cliente – Aba Registro

Todas as telas de cadastro do sistema possuem pelo menos duas abas. Sendo uma delas para busca de dados já cadastrados no sistema, podendo conter filtros requeridos e algumas operações relacionadas. As demais abas serão utilizadas para operações de inserção e alteração de dados.

Nome Cliente	Tipo do Cliente	Cidade	Estado	Telefone
ABOBRAS BLAS	Fabricante	Londrina	PR	(32) 1321-3212
Auto Posto Joao de Barro	Fabricante	Taquarubá	SP	(14) 3762-1982
Empresa Viricicus	Fornecedor	Taquarubá	SP	(14) 3762-1427
Oláia Joao de Barro	Revenda	Taquarubá	SP	(14) 3762-1381

Figura 15: Tela de Cadastro do Cliente – Aba Busca

A Figura 16 representa o menu “Produto” que oferece todas as funcionalidades para cadastro e manipulação do estoque.

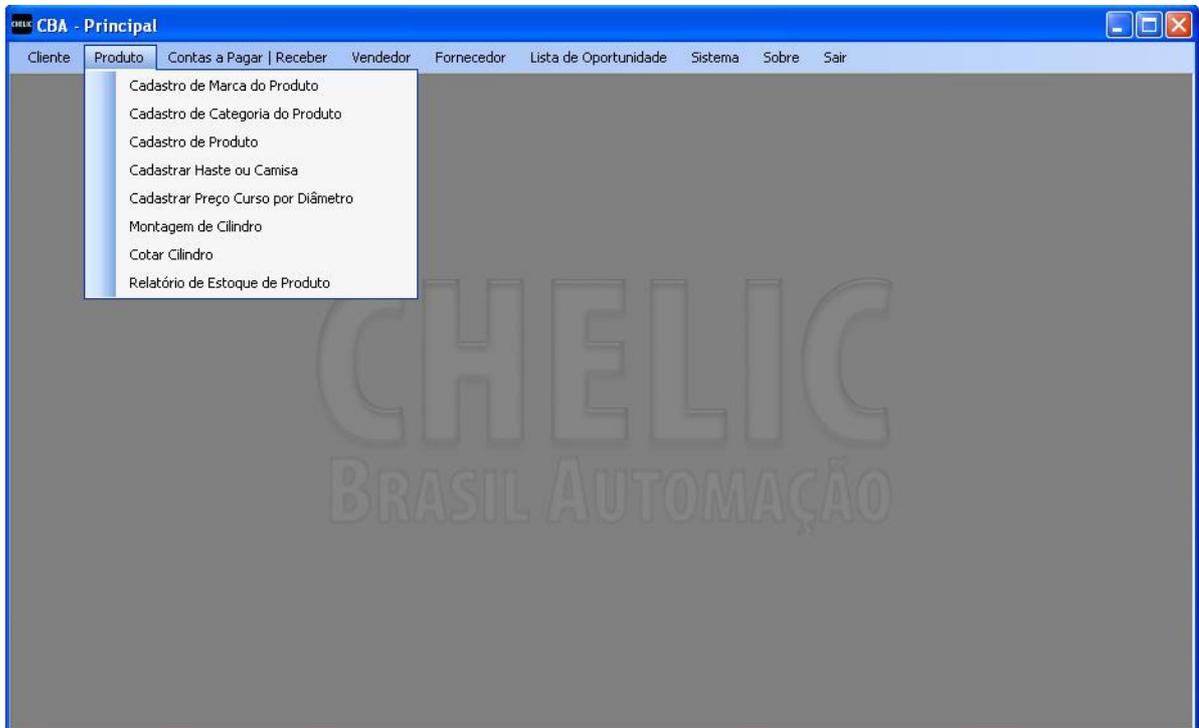


Figura 16: Tela Menu Produto

A Figura 17 representa as telas referente a manipulação dos produtos da empresa. A montagem do cilindro (kits) depende do cadastro de produtos, pois futuramente estes compõem o cilindro. O cadastro de hastes e de camisa são separados, pois o controle de estoque é feito a partir do tamanho do curso informado, no momento de uma autorização de fornecimento ou em uma atualização de estoque. Os preços de cada curso e armazenado em um cadastro separado de acordo com o diâmetro desejado.

The screenshot displays the 'CBA - Principal' software interface with four overlapping windows for product management:

- Produtos:** A window for general product registration with fields for 'Codigo', 'Categoria' (set to 'Anel'), 'Diametro Ø mm', 'Tipo', 'Descrição', 'Marca' (set to 'Chelic'), 'Quantidade', 'Estoque M ínimo', 'Custo Final', 'Preço de Venda', and 'Porcentagem de Lucro'. It includes 'Limpar' and 'Salvar' buttons.
- Montagem de Cilindro:** A window for cylinder kit assembly with fields for 'Código Cilindro', 'Diametro Cilindro(mm)' (set to 10), 'Valor por Curso' (R\$ 10,00), 'Código Kit Montagem' (set to 'tyler_10'), 'Categoria Kit Montagem', 'Tipo Kit Montagem' (set to 'Kit de Cilindro'), 'Descrição Kit Montagem', 'Valor Kit Montagem' (R\$ 15,00), 'Anel' (checked), 'Código do Anel' (set to 'Anel_Tyler'), 'Categoria do Anel' (set to 'Anel'), 'Tipo do Anel', 'Descrição do Anel', and 'Valor do Anel' (R\$ 15,00). It includes 'Limpar' and 'Salvar' buttons and a red asterisk indicating a mandatory field.
- Hastes e Camisas:** A window for registering shafts and sleeves with fields for 'Descrição' (set to 'Camisa'), 'Categoria' (radio buttons for 'Camisa' and 'Haste'), 'Quantidade', 'Diametro Ø mm', and 'Tamanho Curso (mm)'. It includes 'Limpar' and 'Salvar' buttons and a red asterisk indicating a mandatory field.
- Preço de Curso:** A window for course pricing with fields for 'Diametro Ø mm', 'Preço Custo por Curso', and 'Preço de Venda por Curso'. It includes 'Limpar' and 'Salvar' buttons and a red asterisk indicating a mandatory field.

Figura 17: Telas para manipular os Produtos

Para gerenciar a parte financeira, entrada e saída de capital na empresa, e a parte que envolve toda movimentação de estoque o menu foi denominado de “Contas a Pagar | Receber”, conforme Figura 18.

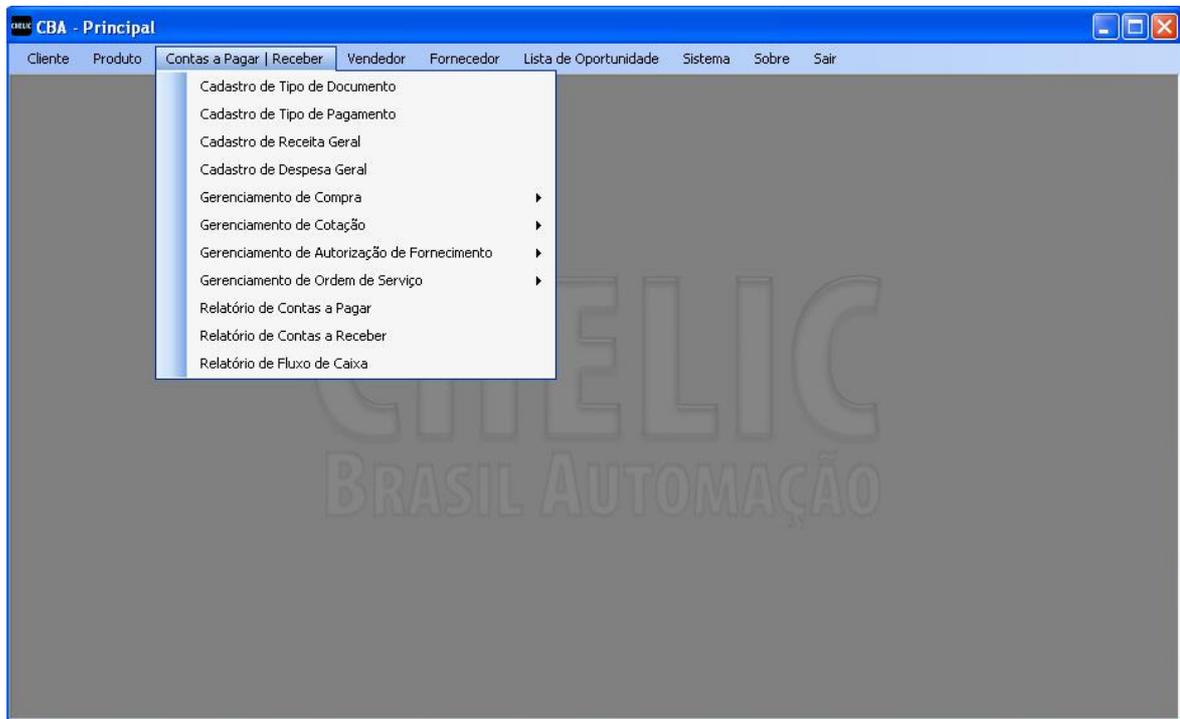


Figura 18: Tela Menu Contas a Pagar | Receber

A Figura 19 corresponde ao gerenciamento de compra de produtos. Todos os dados da compra são cadastrados, mas seu controle de estoque é efetivado ao checar o campo finalizado, está por sua vez não poderá ser alterada.

Toda a parte de gerenciamento de compra também armazena informações para o módulo financeiro (entrada e saída).

The image shows a screenshot of a software application window titled "CBA - Principal". The window contains two overlapping "Compra" (Purchase) forms. The left form is in "Passo 1" (Step 1) and includes fields for "Fornecedor" (Cyprux), "Data Prevista" (quarta-feira, 27 de maio de 2009), "Data Chegada" (quarta-feira, 27 de maio de 2009), "INVOICE", "Finalizado" (checkbox), "Cancelado" (checkbox), "Codigo Produto" (AC-150), "Tipo Produto" (F.R.L.), "Categoria Produto" (1/8"), "Descrição Produto" (Unidade de Tratamento de Ar), "Quantidade", and "Vl. Unitário" (R\$ 285,00). The right form is in "Passo 2" (Step 2) and includes fields for "Tipo de Pagamento" (1 entrada mais 1_tylar), "Tipo de Documento" (Boleto), "Documento", "Valor", "Pago" (checkbox), "Data Pagamento" (quarta-feira, 27 de maio de 2009), and a "Total Restante" (R\$ 0,00) and "Total da Compra" (R\$ 0,00) summary. Both forms have "Salvar Compra", "Excluir Produto", "Editar Produto", and "Passo 2" buttons.

Figura 19: Tela Gerenciar Compra

A Figura 20 corresponde ao gerenciamento de cotação. São cadastrados todos os dados para uma cotação ao cliente.

Figura 20: Tela Gerenciar Cotação

Quando a cotação é aceita pelo cliente, ela pode se tornar uma autorização de fornecimento, apresentada na Figura 21. Posteriormente, isto poderá ser visto através de relatórios.

Código	Cliente	Vendedor	Data Cotação	Prazo de Entrega	Valor	Cancelada	Gerou Venda
23	ABOBRAS. BLAS	vendedor	20/2/2008 ...	20/2/2008 ...	R\$ 496,00	<input type="checkbox"/>	<input checked="" type="checkbox"/>
800	Olaria Joao de B...	vendedor	21/2/2008 ...	21/2/2008 ...	R\$ 646,00	<input type="checkbox"/>	<input checked="" type="checkbox"/>
808	ABOBRAS. BLAS	vendedor	22/2/2008 ...	22/2/2008 ...	R\$ 100,00	<input type="checkbox"/>	<input type="checkbox"/>
816	Olaria Joao de B...	vendedor	3/3/2008 1...	3/3/2008 1...	R\$ 250,00	<input type="checkbox"/>	<input checked="" type="checkbox"/>
824	ABOBRAS. BLAS	vendedor	11/3/2008 ...	11/3/2008 ...	R\$ 2.000,00	<input type="checkbox"/>	<input checked="" type="checkbox"/>
832	Auto Posto Joa d...	porco	24/3/2008 ...	24/3/2008 ...	R\$ 1.254,00	<input type="checkbox"/>	<input checked="" type="checkbox"/>
840	ABOBRAS. BLAS	vendedor	7/4/2008 1...	7/4/2008 1...	R\$ 934,00	<input type="checkbox"/>	<input type="checkbox"/>
848	ABOBRAS. BLAS	vendedor	7/4/2008 1...	7/4/2008 1...	R\$ 4.304,00	<input type="checkbox"/>	<input type="checkbox"/>
856	ABOBRAS. BLAS	vendedor	7/4/2008 1...	7/4/2008 1...	R\$ 638,00	<input type="checkbox"/>	<input type="checkbox"/>
864	Olaria Joao de B...	porco	7/4/2008 1...	7/4/2008 1...	R\$ 1.238,00	<input type="checkbox"/>	<input checked="" type="checkbox"/>
872	ABOBRAS. BLAS	vendedor	7/4/2008 1...	7/4/2008 1...	R\$ 938,00	<input type="checkbox"/>	<input type="checkbox"/>
880	ABOBRAS. BLAS	vendedor	24/6/2008	8/4/2008	R\$ 838,00	<input type="checkbox"/>	<input type="checkbox"/>
888	ABOBRAS. BLAS	vendedor	8/4/2008 1...	8/4/2008 1...	R\$ 1.184,00	<input type="checkbox"/>	<input type="checkbox"/>

Figura 21: Tela Gerenciar Compra – Gera AF

A Figura 22 corresponde ao gerenciamento de autorização de fornecimento. Se todos os produtos tiverem disponíveis em estoque nenhuma AF de pendência é gerada, assim ao salvar o controle de estoque é efetivado. Em caso de pendência o controle só é efetivado após o campo pendente ser desabilitado. Ao concluir uma AF, esta não sendo pendente, sua alteração é bloqueada.

Como no gerenciamento de compra, o gerenciamento de autorização de fornecimento também armazena informações para o módulo financeiro (entrada e saída).

Todas estas informações podem ser vista através de relatórios para melhor controle do usuário.

The screenshot shows a software interface for managing supply authorization (AF). The main window is titled "CBA - Principal" and has a menu bar with options: Cliente, Produto, Contas a Pagar / Receber, Vendedor, Fornecedor, Lista de Oportunidade, Sistema, Sobre, and Sair.

The primary window is "Autorização de Fornecimento" (AF Management). It is divided into two tabs: "Registo" (selected) and "Busca".

Registo Tab:

- Header:** Número Cotação: [dropdown], Vendedor: Emerson
- Client Info:** Nome do Cliente: ABOBRAS BLAS, Pedido de Compra: [dropdown]
- AF Type:** Tipo de AF: Normal
- Logistics:** Transportadora: [text], Pendente:
- Product Details (Detalhes AF):**
 - Então Produto: Normal
 - Código do Produto: AC-150
 - Categoria Produto: 128*
 - Tipo Produto: F.R.L.
 - Descrição Produto: Unidade de Tratamento de Ar
 - Tamanho do Curo: [text]
 - Quantidade: [text], Qtd. Pendente: [text]
 - Valor Venda: R\$ 285,00, Valor Custo: R\$ 85,00
- Table:** A table with columns: Produto, Quantidade, Quantidade Pendente, Valor de Custo, Valor de Venda, T. Custo, Kt. The table is currently empty.
- Summary:** Total: R\$ 0,00
- Buttons:** Nova AF, Excluir Produto, Editar Produto, Salvar AF

Gerar Parcelas Tab:

- Header:** Selecionar AF, Gerar Parcelas
- Payment Info:**
 - Tipo de Pagamento: 1 entrada mais 1Tyler
 - Tipo de Documento: Boleto
 - Documento: [text]
 - Valor: [text]
 - Pago:
 - Data Pagamento: quarta-feira, 27 de maio de 2009
- Table:** A table with columns: Tipo Pagamento, Tipo Documento, Documento, Valor, Pago, Data Pagamento. The table is currently empty.
- Summary:** Total Restante: R\$ 0,00, Total da AF: R\$ 0,00
- Buttons:** Adicionar Parcela, Excluir Parcela, Editar Parcela

Figura 22: Tela Gerenciar Autorização Fornecimento

A Figura 23 corresponde ao gerenciamento de ordem de serviço, neste todos os serviços prestados são descritos e armazenados, incluindo a parte financeira.

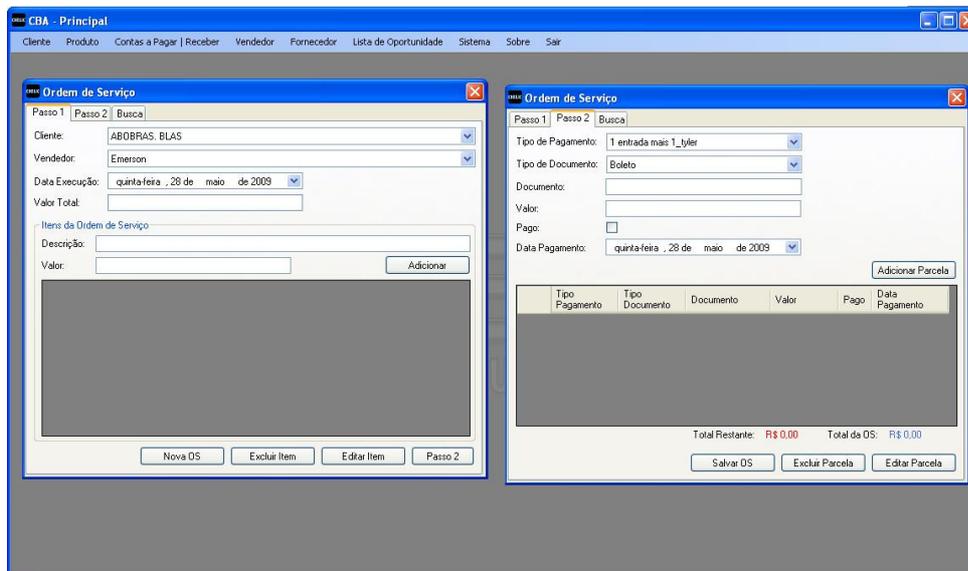


Figura 23: Tela Gerenciar Ordem de Serviço

Para gerenciar oportunidades e contatos com o cliente o menu correspondente foi denominado de “Lista de Oportunidades”, representada pela Figura 24.

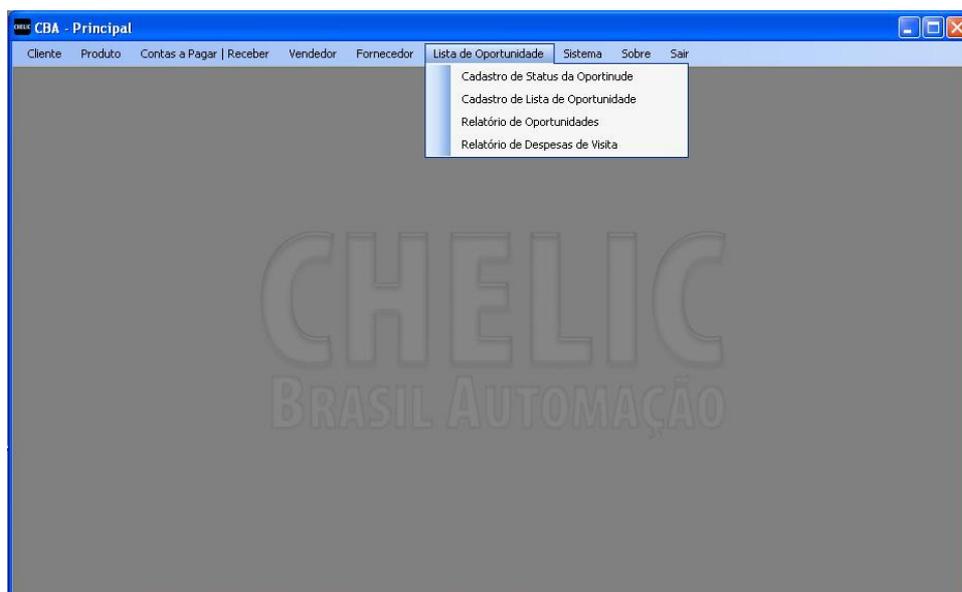


Figura 24: Tela Menu Lista de Oportunidades

A Figura 25 corresponde ao gerenciamento da oportunidade. São cadastradas todas as informações de um contato ou de uma oportunidade gerada por um cliente. Em caso de visita pode-se ocorrer despesa, decorrentes pelo vendedor, que serão armazenadas para o módulo financeiro (entrada e saída).

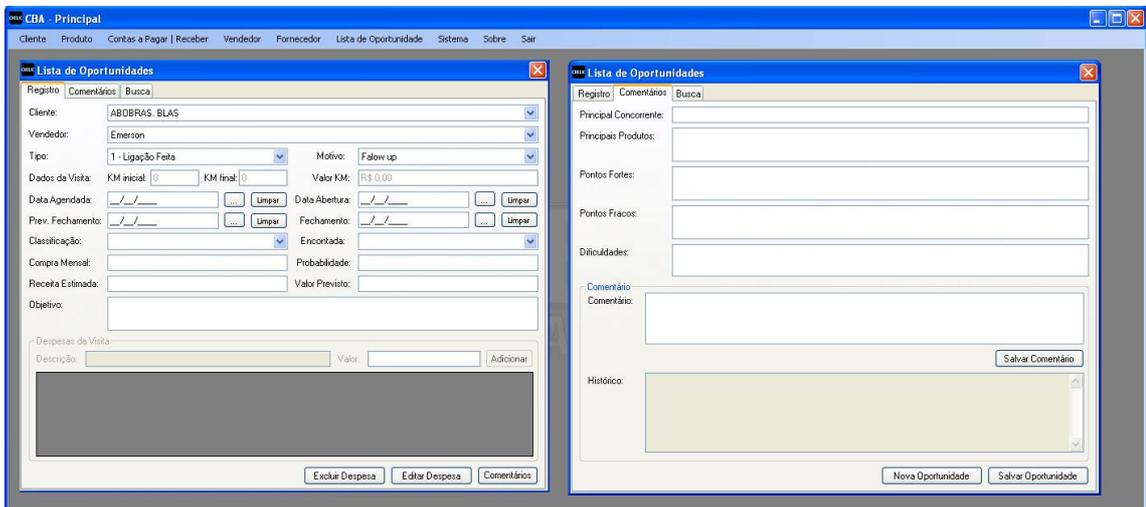


Figura 25: Tela Gerenciar Oportunidade

Toda oportunidade ou contato pode gerar uma cotação que por ventura poderá se torna em uma autorização de fornecimento, armazenando assim todas as informações do primeiro ao último contato com o cliente, conforme Figura 26. Estas informações são informadas através de relatórios que facilitam o controle e a tomada de decisões da empresa.

Código	Data Abertura	Vendedor	Cliente	Previsão de Fechamento	Fechamento	Receita Estimada
10	21/2/2008	vendedor	Olaria Joao de Barro	21/2/2008	21/2/2008	R\$ 8.000,00
11	21/2/2008	vendedor	Olaria Joao de Barro	21/2/2008	21/2/2008	R\$ 5.000,00
12	21/2/2008	vendedor	Olaria Joao de Barro	21/2/2008	21/2/2008	R\$ 300,00
13	25/2/2008	vendedor	ABOBRAS. BLAS			R\$ 10,00
14	3/3/2008	vendedor	Olaria Joao de Barro	6/3/2008	16/3/2008	R\$ 7.000,00
15	10/3/2008	vendedor	ABOBRAS. BLAS	15/3/2008	16/3/2008	R\$ 5.000,00
16	24/3/2008	porco	Auto Posto Joa de B...	27/3/2008	28/3/2008	R\$ 8.000,00
18	29/4/2008	porco	ABOBRAS. BLAS	30/4/2008		R\$ 9.500,00
19	30/4/2008	vendedor	ABOBRAS. BLAS	1/5/2008		R\$ 4.500,00
20	24/4/2008	Rattu Cyprux	ABOBRAS. BLAS	30/4/2008		R\$ 5.000,00
21	6/5/2008	porco fera	ABOBRAS. BLAS	22/5/2008		R\$ 5.000,00
22	14/5/2008	Rattu Cyprux	ABOBRAS. BLAS	16/5/2008		R\$ 5.000,00
23	18/5/2008	porco fera	ABOBRAS. BLAS	22/5/2008		R\$ 10.000,00
24	21/5/2008	vendedor	ABOBRAS. BLAS	26/5/2008		R\$ 1.000,00
25	17/6/2008	Emerson	Cyprux	19/6/2008		R\$ 0,00

Figura 26: Tela Gerenciar Oportunidade – Gera Cotação

Os relatórios presentes no sistema sempre são antecedidos por filtros e informações, para melhor oferecer resultados ao usuário, conforme Figura 27.

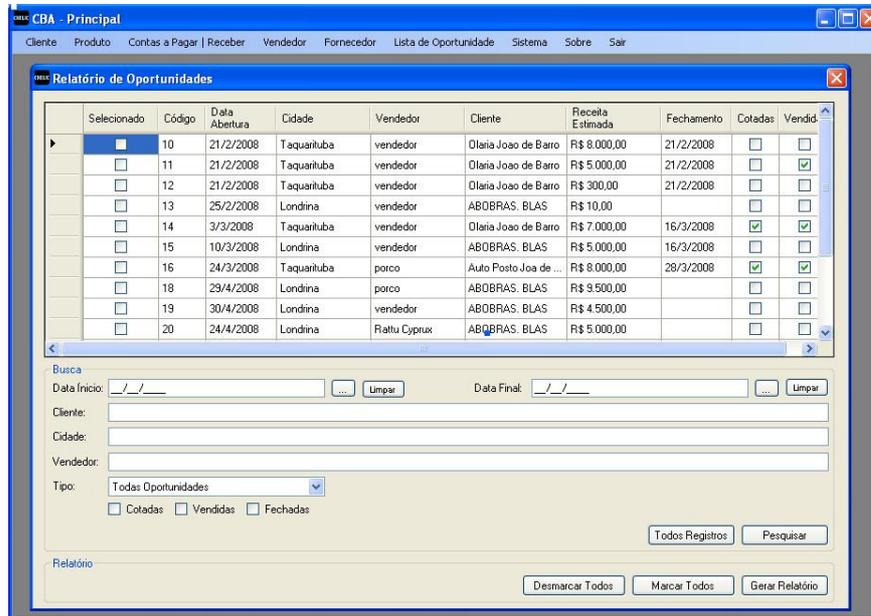


Figura 27: Tela para geração de relatório de Oportunidades

Os relatórios são de layouts simples e limpos, representado pela Figura 28, para fácil entendimento do usuário. Podendo também exportá-los para diversas extensões, como PDF, DOC, JPG, entre diversos.

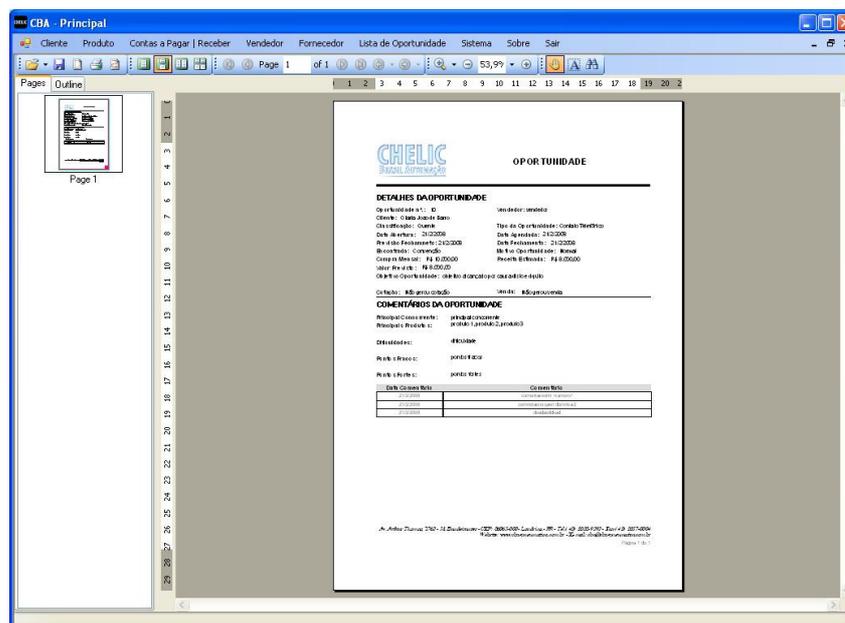


Figura 28: Tela de relatório de Oportunidades

5.5 Aplicação MyGeneration

A ferramenta MyGeneration para utilização de soluções ORM foi de fundamental importância no desenvolvimento do projeto, a seguir veremos como foi utilizado.

5.5.1 Instalação

É necessária fazer o download do MyGeneration primeiramente em :
<http://www.mygenerationsoftware.com/portal/default.aspx>

Junto com o programa, será instalado outros recursos para auxílio durante a programação, como referências DNP, dOODads (camada de persistência de dados) e Zeus (a linha de comando utilizada), um sistema completo com relações de metadados e arquivos de Help completos.

5.5.2 Configuração

Após a instalação quando e inicialização do MyGeneration pela primeira vez ele mostra a janela das configurações padrão - *Default Settings*, na aba *Connection* deve-se definir a string de conexão com o banco de dados para que o MyGeneration seja capaz de executar os templates com base nas tabelas do seu banco de dados.

A linguagem usada pode ser C#, VB .NET no caso da plataforma .NET, e, na opção *DbTarget* é definido o tipo usado no mapeamento entre o tipo de banco de dados e o *driver* escolhido, conforme Figura 29.

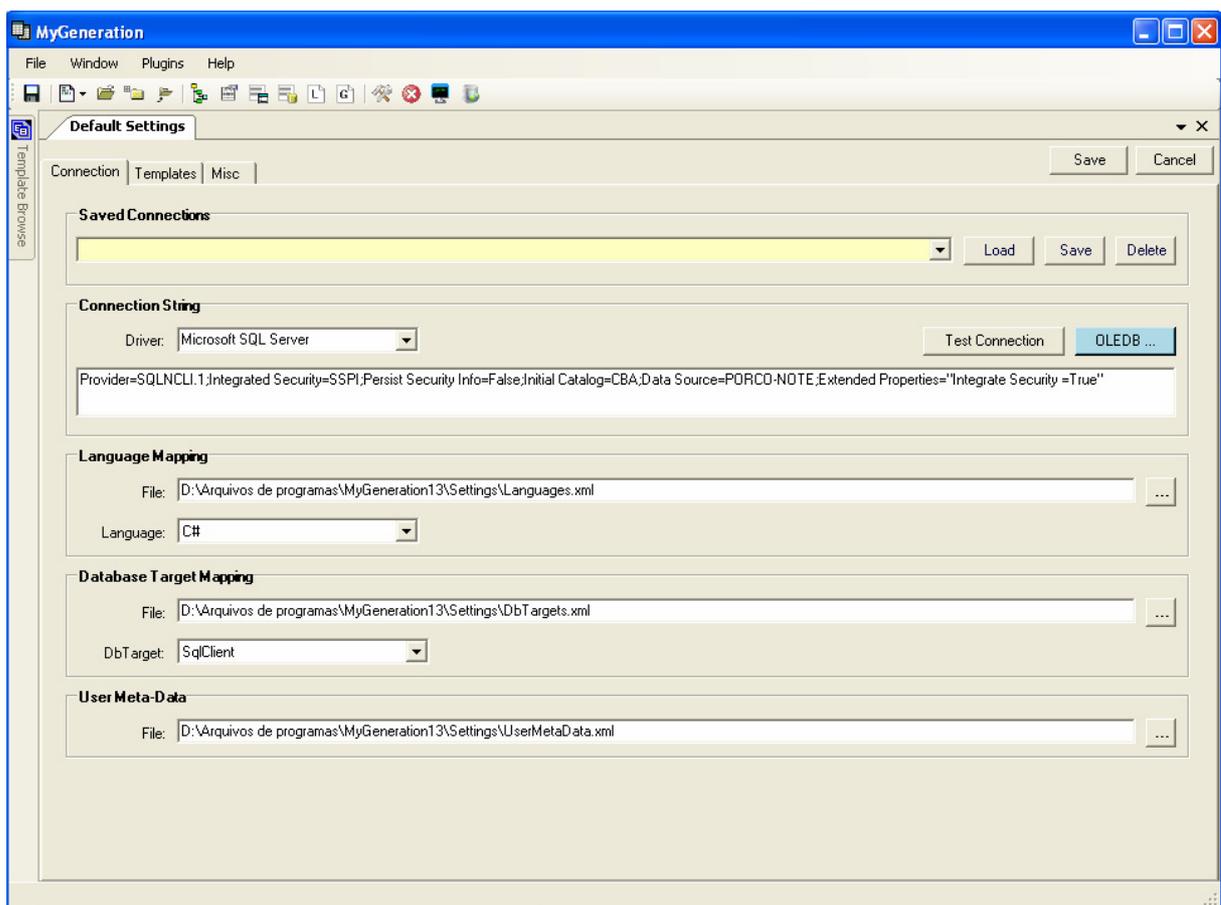


Figura 29: Tela de configuração do MyGeneration

5.5.3 Utilização

Após estas definições pode-se usar um template pronto ou definir o um template próprio.

Para escolher o template é necessário abrir a aba chamada “Template Browser”, através do local indicado pela Figura 30.

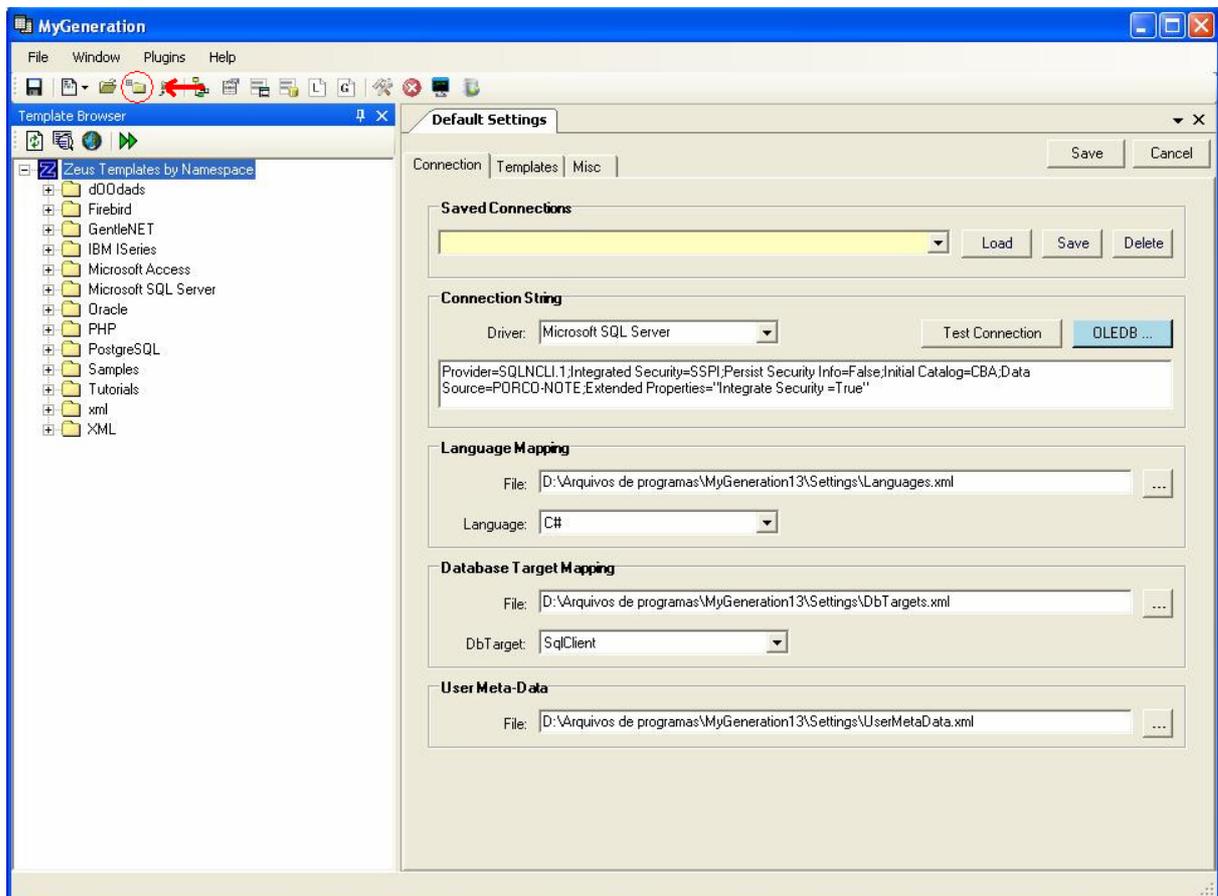


Figura 30: Tela de configuração do MyGeneration

A linha de comando utilizada para geração dos códigos é a chamada, “Zeus”, conforme Figura 30. Primeiramente é gerada uma interface para que as futuras classes, que irão ser geradas posteriormente, façam as operações de CRUD (*Create, Review, Update, Delete*) com o banco de dados.

O campo “Output file path” significa o local onde serão criadas as *stored procedures*. A operação esta exemplificada através da Figura 31 abaixo:

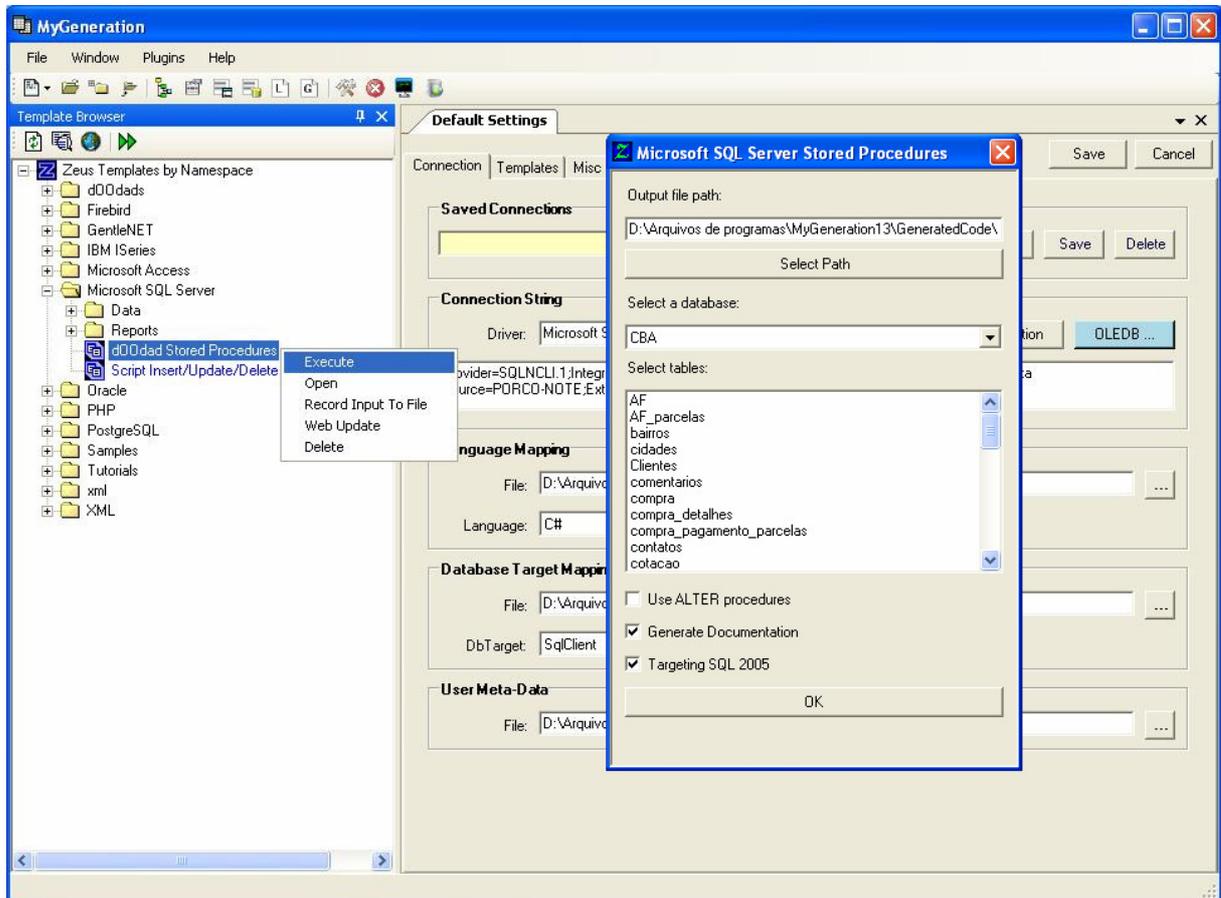


Figura 31: Tela de configuração do MyGeneration

Ao serem criadas são executadas no banco de dados desejado.

O MyGeneration fornece um camada de persistência de dados e objetos para o .Net chamada “dOODads”, que serão utilizadas para a geração das classes que irão se comunicar com o banco de dados.

O “dOODads” nos fornece a geração de três tipos de classe, representada na Figura 32:

- dOODads Business View: classes que irão se comunicar com as views criadas no banco de dados.
- dOODads Business Entity: classes que irão se comunicar com as tabelas criadas no banco de dados.
- dOODads Concrete Class: classes que herdam todos os métodos e propriedades das classes Entity (que fazem todas as operações com banco de dados) e ficam abertas para implementações próprias ou para sobrescrever algum método existente.

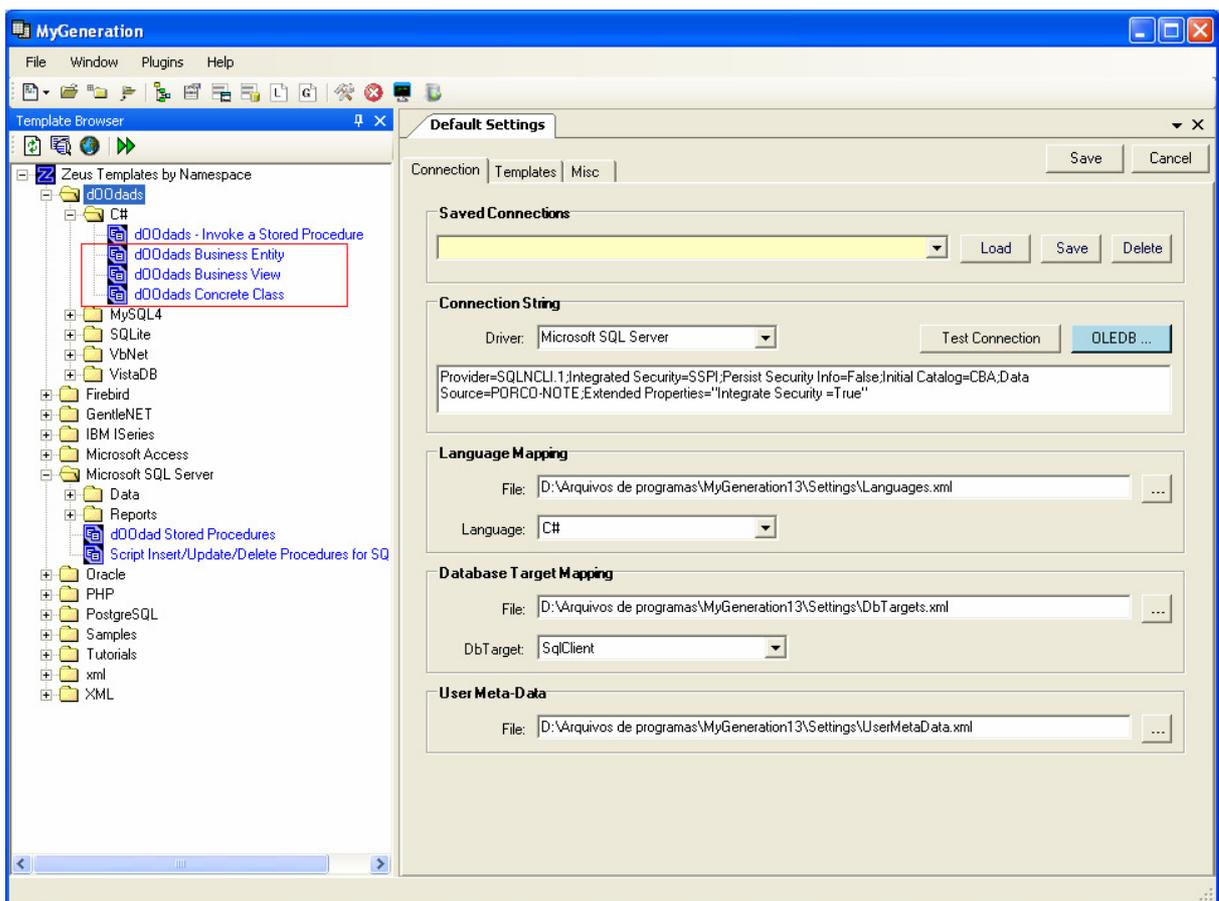


Figura 32: Tela de configuração do MyGeneration

As classes geradas estão representadas através da modelagem UML apresentada anteriormente.

5.5.4 Aplicação

Com o auxílio do MyGeneration a escrita do código fica mais produtiva e elegante. O projeto se torna mais manutenível e mais padronizado.

```
private void btnFiltrar_Click(object sender, EventArgs e)
{
    view_af_grid af = new view_af_grid();
    af.Where.Excluido.Value = 0;
    af.Where.Excluido.Operator =
MyGeneration.dOODads.WhereParameter.Operand.Equal;
    af.Where.Data.Conjunction = MyGeneration.dOODads.WhereParameter.Conj.And;
    af.Where.Data.BetweenBeginValue =
Convert.ToDateTime(tbDataInicio.Value).ToShortDateString();
    af.Where.Data.BetweenEndValue= Convert.ToDateTime
(Convert.ToDateTime(tbDataFinal.Value).ToShortDateString()).AddHours(23).AddMinutes(59);
    af.Where.Data.OperatorMyGeneration.dOODads.WhereParameter.Operand.Between;

    if (af.Query.Load())
    {
        dgvAF.DataSource = af.DefaultView;
        dgvAF.Refresh();
        DataGridViewCellStyle style = new DataGridViewCellStyle();
        style.Format = "C2";
        dgvAF.Columns[8].DefaultCellStyle = style;
        dgvAF.Refresh();
    }
    else
    {
        MessageBox.Show("Nenhum registro encontrado", "Filtrar",
MessageBoxButtons.OK, MessageBoxIcon.Information);
    }
}
}
```

Quadro 03: Exemplo de Aplicação MyGeneration
Fonte: Revista .Net Magazine, (2007)

No Quadro 03 apresentado é utilizada a classe “view_af_grid” que é instanciada através do objeto “af”. São atribuídos valores que simulam um query SQL, que filtram os dados da view e alimentam assim um grid denominado de “dgvAF”. Tudo isso ocorre dentro do evento do botão “btnFiltrar”.

6 CONSIDERAÇÕES FINAIS

Após o estudo feito sobre as tecnologias apresentadas, POO, sintaxe da linguagem de programação C#, ferramenta MyGeneration para utilização de soluções ORM e Microsoft SQL Server 2005 pôde-se chegar as conclusões abaixo descritas:

O conceito de POO proporciona um ganho de tempo nas implementações e manutenções, devido a conceitos, que permitem o reaproveitamento de código. Esse código reaproveitado também influencia na alteração do sistema, portanto, a alteração feita no código herdado, refletirá em todos os objetos que o herdem.

Unindo as técnicas de OO com as soluções ofertadas através da ferramenta MyGeneration, o desenvolvimento se torna mais produtivo, deixando o código de uma maneira mais elegante e simples, além de deixá-lo mais padronizado.

Existem muitas ferramentas que oferecem soluções como a do MyGeneration. Pretende – se assim estudo de novas técnicas que otimizem ainda mais o processo de desenvolvimento.

Em relação a softwares de gerenciamento empresarial, todos se apresentam de uma maneira muito generalizada e muitas vezes não atendem as necessidades reais da empresa. Dessa forma destacam aqueles que aceitam implementações complementares, assim sendo, a forma na qual as tecnologias foram empregadas, deixou o sistema pronto para receber implementações futuras.

REFERÊNCIAS

- ALUR D., CRUPI & J., MALKS, D.. **Core J2EE Patterns: Best Practices and Design Strategies**. 2001. Publisher: Prentice Hall / Sun Microsystems Press, 2001.
- ARAÚJO, A. V. Conceitos Iniciais. In: _____. **Treinamento Avançado em .Net**. [S.L.]: Universo dos livros, 2009. p. 5-22. Disponível em : <http://books.google.com.br/books?id=Wrrw_mJ6Z64C&pg=PA17&dq=ENCAPSULAMENTO&lr=#PPA18,M1>. Acesso em : 17/05/2009
- CAMARA, Fabio. **C# Introdução e Visão Geral**. Disponível em: <http://www.csharpbr.com.br/mostra_artigo.asp?id=0003> Acessado em: 15/02/2009
- COMPONENTONE. **Reports for WinForms**. Disponível em: < <http://www.componentone.com/SuperProducts/ReportsWinForms/>> Acessado em: 21/05/2009
- DEITEL, H. M ., DEITEL, P. J., LLISTIFIELD, J., et al. Introdução ao IED do Visual studio .Net. In: _____. **C# Como Programar**. São Paulo: Pearson Makron Books, 2003. p.27-36.
- FURLAN, J. D.. **Modelagem de objetos através da UML**. São Paulo. Ed.: Makron Books, 1998. p. 23 – 30.
- FUSSEL, M. L.. **Foundations of Object Relational Mapping**. ChiMu Cooperation. 1997. Disponível em <www.chimu.com/publications/objectRelational/>. Última visita em 05/05/2009.
- GUEDES, G. T. A. Orientação a objetos. In: _____. **UML Uma abordagem prática**. 3.Ed. São Paulo: Novatec, 2007, p. 39 – 45.
- JUDE COMMUNITY SITE. Disponível em: <<http://jude-users.com/en/>> Acesso em: 20/05/2009.
- MACORATI, J. C. **.NET 2008 - Gerando código com MyGeneration**. Disponível em: http://www.macoratti.net/08/10/net_myg.htm. Acesso em: 14/02/2009
- MEYER, B.. **Object - Oriented Software Construction**. Prentice Hall, 2000. 2 edition
- MICROSOFT ®. **Sql Server. O que é sql Server ?**. Disponível em: <<http://www.microsoft.com/brasil/servidores/sql/prodinfo/overview/what-is-sql-server.aspx>>. Acesso em: 25/01/2009.
- MICROSOFT ® DEVELOPER NETWORK. **Introdução à linguagem C# e ao Framework .NET**. Disponível em: <<http://msdn.microsoft.com/pt-br/library/z1zx9t92.aspx>>. Acesso em: 22/01/2009.

MICROSOFT® DEVELOPER NETWORK. **Visão geral conceitual do .NET Framework.** Disponível em: <<http://msdn.microsoft.com/pt-br/library/zw4w595w.aspx>>. Acesso em: 20/01/2009b.

MIRANDA, M. A. **Introdução ao .NET Framework.** 2008. Disponível em : <<http://www.concatenar.com/index.php?acao=artigo&cdtutorial=90&clique=1>>. Acesso em: 18/02/2009

REVISTA. **Pequenas Empresas. Grandes Negócios Informática.** Rio de Janeiro. Editora Globo S.A. 2006 P. 62.

REVISTA .NET MAGAZINE - **Análise dos melhores ORM.** Rio de Janeiro. DevMedia. 2007. p. 26 - 35

SHORT, Scott. **Building XML Web Services for the .NET Platform.** Washington, EUA: Microsoft Corporation, ISDN:[0-735-61406-7], 2002. p. 1-5

SILVIO, R. O. **SQL Server 2005.** 2007. Apostila (SQL Server 2005).UNIBAN – São Paulo.

STOCK SOFT. **ComponentOne Studio Enterprise.** 2009. Disponível em : <http://www.allcominfo.com.br/produtos_det.asp?ID=46>. Acesso em: 21/05/2009

TERRY, Q.. **Modelagem visual com Rational Rose 2000 e UML.** Rio de Janeiro. Editora Ciência Moderna Ltda, 2001. P 30 - 43

THAI, Thuan. LAM, Hoang. **NET Framework Essentials.** Washington, EUA: O'Reilly, ISDN:[0-596-00165-7], 2001. p 1-47

UCHÔA, J. P. **.NET Framework – Introdução.** 2006. Disponível em: <http://www.linhadecodigo.com.br/ARTIGO.ASPX?ID=1149>. Acesso em: 17/02.2008

VINCENZI, A. M. R.. **Orientação a objeto: definição, implementação e análise de recursos de teste e validação.** Instituto de Ciências Matemáticas e de Computação - ICMC – USP, 2004. p 28–29.

WIKPEDIA, A ENCICLOPÉDIA LIVRE. **Common Language Runtime.** Disponível em: < http://pt.wikipedia.org/wiki/Common_Language_Runtime>. Acesso em: 20/02/2009a.

WIKPEDIA, A ENCICLOPÉDIA LIVRE. **Microsoft Visual Studio.** Disponível em: < http://pt.wikipedia.org/wiki/Visual_studio>. Acesso em: 20/02/2009.

YOURDON, E. . **Análise e projeto orientados a objetos.** São Paulo. Ed.: Makron Books, 1999. p. 23 – 30.

