UNIVERSIDADE REGIONAL DE BLUMENAU

CENTRO DE CIÊNCIAS EXATAS E NATURAIS

CURSO DE CIÊNCIAS DA COMPUTAÇÃO – BACHARELADO

FERRAMENTA CASE PARA GERAÇÃO DE PÁGINAS ASP

CRISTIANO DE CASTILHOS

BLUMENAU
2004

2004/1-05

CRISTIANO DE CASTILHOS

FERRAMENTA CASE PARA GERAÇÃO DE PÁGINAS ASP

Trabalho de Conclusão de Curso submetido à Universidade Regional de Blumenau para a obtenção dos créditos na disciplina Trabalho de Conclusão de Curso II do curso de Ciência da Computação — Bacharelado.

Prof. Wilson Pedro Carli - Orientador

BLUMENAU
2004

2004/1-05

FERRAMENTA CASE GERADORA DE PÁGINAS EM ASP

Por

CRISTIANO DE CASTILHOS

Trabalho aprovado para obtenção dos créditos na disciplina de Trabalho de Conclusão de Curso II, pela banca examinadora formada por:

Presidente:	Prof. Wilson Pedro Carli, FURB
Membro:	Prof. Everaldo Arthur Grahl, FURB
Membro:	Prof. Fabiane B. Vavassori Benitti, FURB

Blumenau, 02 de julho de 2004

Dedico este trabalho primeiramente a Deus e aos meus pais: Marino de Castilhos e Sônia M. Silva de Castilhos por sempre me apoiarem nos estudos. Também dedico aos meus irmãos, a minha namorada e a todos os amigos que me ajudaram diretamente na realização deste.

Quem fica esperando que o vento mude e que o tempo fique bom nunca plantará nem colherá nada.

Bíblia Sagrada, Eclesiastes, Cap. 11:4.

AGRADECIMENTOS

À Deus, que tem me dado sabedoria, saúde e muita força de vontade para superar quaisquer desafios que encontram-se em minha vida.

Aos meus pais, que sempre me apoiaram nos estudos e financeiramente para a realização deles.

Aos meus irmãos e à minha namorada, pela paciência e por me entender em momentos difíceis durante esta etapa da minha vida.

Aos meus amigos Jonathan Kraemer e Thomas Sens, que serviram de suporte e apoio para o desenvolvimento e conclusão deste trabalho.

Ao meu orientador, Wilson Pedro Carli, por ter acreditado em minhas idéias e na conclusão deste trabalho.

RESUMO

Este trabalho apresenta o desenvolvimento de uma Ferramenta CASE, para criação, definição, documentação e geração de páginas em ASP de um projeto. Através da linguagem de *Script* ASP (*Active Server Pages*) e com a estrutura do projeto definida no banco de dados SQL Server será possível a geração de qualquer sistema de formulários com inclusão, alteração, exclusão, consulta, localizar e paginação dos registros.

Palavras chaves: Monografia; Resumo; Ferramenta CASE, Gerador, ASP.

ABSTRACT

This work presents the development of a Tool CASE, for creation, definition, documentation and generation of pages in ASP of a project. Through the language of Script ASP (Active Server Pages) and with the structure of the project defined in the data base SQL Server it will be possible the generation of any form system with inclusion, alteration, exclusion, consultation, locate and paging of the registers.

Words keys: Monograph; Abstract; Tool CASE; Generator; ASP.

LISTA DE FIGURAS

LISTA DE QUADROS

LISTA DE TABELAS

LISTA DE SIGLAS

ADO – Activex Data Object

ASP – Active Server Pages

B2B – Business-to-Business

B2C – Business-to-Consumer

BI – Business Intellingence

CASE – Computer Aided Software Engineering

CRM – Customer Relationship Management

DBA – Data Base Administrator

DBQ – Data Base Query

DER– Dia grama de Entidades e Relacionamentos

ERP – Enterprise Resource Planning

IIS – Internet Information Service

MER – Modelo de Entidades e Relacionamentos

ODBC – Open Database Connectivity

OLE DB – Object Linking and Embedding DataBase

SCM – Supply Chain Management

SQL – Structuded Query Language

T-SQL – Transact-SQL

XML – Extensible Markup Language

LISTA DE SÍMBOLOS

@ - arroba

% - por cento

- sustenido

SUMÁRIO

1 INTRODUÇÃO

O software agora ultrapassou o hardware como a chave para o sucesso de muitos sistemas baseados em computador. O software é um fator que diferencia para dirigir um negócio, controlar um produto ou capacitar um sistema (PRESSMAN, 1995, p.3). Conforme Silveira (2003, p.1), à medida que o tempo passa, os softwares tornam-se cada vez mais complexos, possuindo grande quantidade de funcionalidades.

Várias empresas de desenvolvimento estão aos poucos sendo induzidas pelo mercado a migrarem seus softwares para *web*. Alguns aplicativos independentes já foram criados por empresas pioneiras neste ramo. Estes aplicativos são independentes de hardware pois são gerados através de linguagem de alto nível e qualquer *browser* com acesso a internet pode acessá-lo. Estes aplicativos *web* são hospedados por Provedores de Serviços de Aplicações que são chamados de ASP's. Estes ASP's são servidores disponibilizados em grandes centros de dados, chamados de *Data Centers*. Estes *Data Centers* possuem infra-estrutura para manter a segurança de todas as informações armazenadas.

Os *Data Centers* disponibilizam aplicativos que vem tomando conta do mercado: *Customer Relationship Management* (CRM), *Supply Chain Management* (SCM), *Enterprise Resource Planning* (ERP), *Business-to-Business* (B2B), *Business to-Consumer* (B2C) e *Business-Intelligence* (BI). A partir do crescimento deste mercado de aplicativos independentes da *web*, a engenharia de software sofre várias modificações.

As linguagens de alto nível permitem que o desenvolvedor de software e o programa sejam independentes da máquina (PRESSMAN, 1995, p. 32), por este motivo pode-se dizer que é um aplicativo independente.

A área de engenharia de software tem atualizado constantemente seus conceitos devido às necessidades do mercado *web*. Os aplicativos independentes da *web* também têm utilizado métodos para modelagem conceitual e para mostrar a semântica do domínio da aplicação. A tecnologia da informação tem sido fortemente empregada para suportar modelos empresariais, incluindo aspectos importantes como recursos físicos e lógicos, regras de negócio, objetivos e processos (FURLAN, 1998). Para agilizar o processo de engenharia do software estão disponíveis no mercado as ferramentas *Computer Aided Software Engineering* (CASE).

Uma ferramenta CASE é a forma de apoio para acelerar e padronizar a documentação do desenvolvimento de software, onde quando bem utilizada, traz produtividade e qualidade a

este desenvolvimento. Para Martin (1991), é necessária uma revolução industrial do software, que provavelmente virá das técnicas orientadas a objeto combinadas com ferramentas CASE, geradores de código, programação visual e desenvolvimento baseado em repositórios. O objetivo deste conjunto de ferramentas é maximizar a reusabilidade de código, construindo e armazenando objetos complexos para posterior utilização, tornando o desenvolvimento de software mais rápido (SILVEIRA, 2003, p. 1).

Os geradores de código são ferramentas que produzem código sem nenhum erro de sintaxe a partir de projetos, gráficos e especificações de alto nível (MARTIN, 1991). O código deve ser gerado a partir de tabelas de decisão, regras, diagramas de ação, diagramas de eventos, diagramas de transição de estado, representação de objetos e suas propriedades e relacionamentos e assim por diante.

A necessidade das empresas em disponibilizar informações *on-line* é grande. Esta necessidade já vem sendo atendida com aplicativos independentes na *web*. Estes aplicativos são integrados com banco de dados e interagem com conteúdo dinâmico.

Segundo Buczek (2000), o *Active Server Pages* (ASP), oferece aos desenvolvedores da *web* meios de ativar seus *sites* com um conteúdo dinâmico, movido por uma base de dados. Os processos de desenvolvimento de *sites* com conteúdo dinâmico tais como: ligações das tabelas, representação do fluxo de dados, dicionário de dados, consistência na validação dos campos, definição de informações em grandes formulários de cadastros entre outros processos acarretam deficiências quanto ao tempo de desenvolvimento de páginas em ASP, aumentando o custo de desenvolvimento para as empresas *web*.

Estas deficiências podem ser atendidas com a ferramenta CASE desenvolvida que poderá melhorar estes processos e através dos mesmos, gerar aplicações completas de cadastros. Estas aplicações de cadastro possuem códigos fontes em ASP que contem a inclusão, exclusão, alteração, consulta e paginação de registros que será a base para a criação automática dos aplicativos de cadastros.

Estes aplicativos poderão sofrer alterações pelos seus administradores de qualquer lugar onde exista um computador com acesso a internet. Estas alterações poderão ser efetuadas na ferramenta CASE desenvolvida, que estará disponível *on-line* para a criação de novos projetos, tabelas, campos e novos relacionamentos para a geração do banco de dados relacional e da aplicação.

Através de alguns desenvolvedores *web* e estudantes da nossa universidade, verificou-se a necessidade de uma ferramenta que gerasse códigos fontes automaticamente a partir de um banco de dados previamente criado e modelado.

Na ferramenta CASE desenvolvida, o administrador definirá as informações necessárias para a criação dos projetos, tabelas, campos e relacionamentos para a geração da aplicação. Os cadastros possuem opções de inclusão, exclusão, alteração, localizar/consulta e paginação de registros, utilizando a linguagem de *script* ASP, baseado na estrutura dos campos relacionados na própria ferramenta CASE.

Esse tipo de ferramenta já é encontrado hoje no mercado (TEGNHER, 2001) ou em trabalhos de conclusões de cursos (DIAS, 2002 e SILVEIRA, 2003). Essas ferramentas permitem ao administrador através de várias etapas indicarem todas as tabelas e campos que serão utilizados para a geração de códigos. O diferencial da ferramenta implementada está na capacidade do usuário fazer a criação do projeto, das tabelas, dos campos, dos relacionamentos, da documentação, podendo automaticamente gerar toda a aplicação de cadastros.

1.1 OBJETIVOS DO TRABALHO

O objetivo deste trabalho foi o desenvolvimento de uma ferramenta CASE direcionada para o auxílio de programadores e analistas de sistemas para definição e especificações do projeto, na criação de aplicativos *web*, na geração de código, tendo assim por finalidade principal agilizar a criação de aplicativos de cadastros reduzindo custos as empresas e ganhando tempo no desenvolvimento.

Os objetivos específicos do trabalho são:

a) configurar conexão com o banco de dados:
   - escolha do *driver* (Access ou SQL Server);
   - IP da máquina onde encontra-se o banco de dados;
   - autenticação (*login* e *senha*) do banco de dados;

b) criar tabelas e uma estrutura de campos com seus respectivos tipos, tamanhos, se é requerido ou não e com validação. Como exemplo pode-se citar: *E-mail*, CPF e CNPJ;

c) relacionar os campos das tabelas;

d) criação de um banco de dados relacional;

e) gerar automaticamente as páginas de cadastro em ASP com inclusão, exclusão, alteração, consulta, paginação de registros;

f) documentação do projeto.

1.2 JUSTIFICATIVA

Justifica-se este trabalho pela criação automática de aplicativos de cadastros completos com banco de dados relacional, permitindo alto ganho de produtividade no desenvolvimento páginas em *ASP*. Estas páginas serão geradas de forma parametrizada e sem intervenção do programador no código gerado. O administrador somente precisará criar, definir e documentar o projeto através da ferramenta CASE.

Esta ferramenta CASE, além de fazer toda a criação, definição, documentação do projeto, a ferramenta cria o banco de dados, as tabelas, os campos, os relacionamentos no banco de dados, geração do código da "*Administração*" e do "*Front-end*" do *site*, sendo estes os diferenciais das ferramentas já encontradas como o DataForm (Tegnher, 2001) e os trabalhos de conclusão de curso apresentados por Dias (2002) e Silveira (2003).

1.3 ESTRUTURA DO TRABALHO

O primeiro capítulo apresenta uma introdução sobre o assunto e objetivos do trabalho, a fim de dar ao leitor as informações necessárias ao entendimento do assunto abordado.

O segundo capítulo contextualiza alguns conceitos gerais sobre Internet, banco de dados e a linguagem de *script* ASP, para que o leitor possa entender melhor o processo de criação, definição, documentação e geração dos códigos.

O terceiro capítulo apresenta a especificação e implementação da ferramenta, destacando a metodologia utilizada, a modelagem do sistema e as considerações finais sobre a codificação.

O quarto capítulo conclui o trabalho e apresenta sugestões para trabalhos futuros.

2 FUNDAMENTAÇÃO TEÓRICA

Esse capítulo contextualiza alguns conceitos gerais sobre a geração de aplicativos web, linguagem de *script* ASP e ferramentas CASE.

2.1 GERAÇÃO DE APLICATIVOS WEB

Atrás das telas da maioria dos aplicativos ASP, está um banco de dados; o que está armazenado naquele banco de dados é o que torna o ambiente dinâmico. A sua habilidade em se comunicar com aquele banco de dados de modo eficiente irá delinear a complexidade e as possibilidades de uso de sua aplicação ASP (BUCZEC, 2000, p. 512).

Justamente para agregar valor aos seus serviços de internet, empresas e organizações buscam disponibilizar aplicativos *web*, programas acessados pelo navegador de internet com a possibilidade de interação entre os usuários e a lógica de negócio. Exemplos óbvios de aplicativos *web* são as lojas de comércio eletrônico, mecanismos de busca que agregam informações, sistemas de internet banking, entre outros aplicativos para controle de *sites*. Aplicativos *web* também são viabilizados através de *intranets* ou *extranets*, redes restritas a uma empresa, organização ou grupo, que possibilitam o trabalho colaborativo e estruturado entre seus atores (UNITO, 2004).

2.2 ASP (ACTIVE SERVER PAGES)

Conforme Buczek (2000, p. IX), *Active Server Pages* (ASP) oferece aos desenvolvedores da *web* meios de ativar seus sites com um conteúdo vivo e dinâmico, movido por uma base de dados. O código que produz este rico conteúdo é todo do servidor, o que significa que ele é executado no servidor.

O ASP fornece um ambiente baseado em script de servidor, que conduz a aplicações dinâmicas, interativas, que não consomem um enorme tempo de processamento. Isto é um alívio, tanto para o desenvolvedor experiente da *web* como para o *webmaster* novato. Agora, o desenvolvedor pode construir lógica em sua página da *web* estática e transformar páginas somente informativas em páginas interativas com conteúdo vivo (WILLIE, 1999, p. 31).

2.2.1 OBJETOS DO ASP

Segundo Dias (2002, p. 7), o ASP possui objetos básicos para criar suas aplicações. Através destes objetos podem-se executar diversas funções que vão desde variáveis até acesso a banco de dados.

A disposição encontra-se a Tabela 1 abaixo para detalhes:

Tabela 1 - Detalhes dos Objetos.

Objetos	**Finalidade**
Request	Fornece informações sobre seu visitante.
Response	Métodos e propriedades para construção da sua resposta para o visitante.
Application	Trata as propriedades que governam um agrupamento de páginas *web* e são referenciadas como uma aplicação.
Server	Trata a criação de componentes e configurações do servidor.

Fonte: Buczek (2000).

2.2.1.1 Objeto Request

O Objeto Request retorna os valores das requisições feitas pelo *browser* do cliente ao servidor durante uma requisição HTTP. Exemplo:

<% Request.Form(“Nome”) %>

2.2.1.2 Objeto Response

O Objeto Response envia informações do servidor para o cliente. Exemplo:

<% Response.Write(“Nome”) %>

2.2.1.3 Objeto Application e Session

Os objetos *Application* e *Session* juntos gerenciam o armazenamento de informação em nível de sessão e de aplicação. O nível da aplicação é o tempo que o servidor fica ativo, e o nível de sessão é o tempo que o usuário acessa uma página. Exemplo abaixo, o *Application* e o *Session* recebem a data atual:

<% Application(“data”) = Now() %>

<% Session(“data”) = Now() %>

2.2.1.4 Objeto Server

O Objeto *Server* é usado para gerenciar e criar objetos, permitindo o processamento de *scripts* e de acesso a base de dados. Exemplo de criação de um objeto *Server*:

<% Server.CreateObject(TCC) %>

2.2.2 ASP E HTML

Conforme Wille (1999, p. 13), um arquivo ASP é um arquivo de texto com a extensão (.asp) que contém uma combinação de texto, *tags* de HTML e comandos de *script* ASP, ele pode ser criado ou alterado com o uso de todos os editores de texto.

O código ASP delimita-se com os caracteres <% e %> e incluem comandos de *scripts* definidos e padronizados pelo servidor. O ASP também pode controlar aparências de textos e *tags* HTML fora dos blocos de *script*. No Quadro 2 está um exemplo de interação entre o ASP e o HTML.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<title>TCC - Exemplo de uma página em HTML</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
<body>
<div align="center"><img src="img/logo_furb.jpg" width="293"
height="182"><br><br>
<Font color="#FF0000"><b>Trabalho de Conclusão de Curso</b></Font><br>
Autor: <b>Cristiano de Castilhos</b></div>
<center>
   <%
   session("data") = date()
   Response.write("Data atual:"&session("data"))
   %>
</center>
</body>
</html>
```

Quadro 1 - Exemplo de interação entre o ASP e o HTML.

2.2.3 ASP E BANCO DE DADOS

Segundo Wille (1999, p. 193), o banco de dados é usado para gerenciar as informações sendo que o *ActiveX Data Objects* (ADO) é o fornecedor da interface entre o banco de dados e o ASP. Os bancos de dados podem ser acessados com o *Object Linking and Embedding Database* (OLE DB) ou o *Open Database Connectivity* (ODBC).

O OLE DB fornece uma maneira de acessar todos os tipos de dados, tanto relacionais quanto não relacionais. O OLE DB pode ser acessado diretamente usando o ADO. O ODBC, por outro lado, fornece uma maneira de acessar os dados relacionais. Ele também é uma interface padrão da indústria e, portanto, está disponível para quase todos os bancos de dados (WILLE, 1999, p. 194).

Para acessar o banco de dados através do ASP, primeiramente precisamos conectá-lo que se faz pelo código descrito no Quadro 3 abaixo:

```
Set Conn = Server.CreateObject("ADODB.Connection")
```

Quadro 2 - Objeto *Connection* para conexão com um banco de dados.

2.2.4 TAXONOMIA DE FERRAMENTAS CASE

Uma série de riscos é inerente quando quer que tentemos categorizar as ferramentas CASE. Existe a sutil implicação de que, para criarmos um ambiente CASE efetivo, deve-se implementar todas as categorias de uma ferramenta CASE, mas isso não é verdade. Confusão pode ser criada ao colocar uma ferramenta específica dentro de uma categoria quando outros poderiam achar que ela pertence a outra categoria (PRESSMAN, 1995, p. 950).

Mesmo assim é necessária a criação de uma taxonomia de ferramentas CASE para melhor entender a amplitude do CASE e melhor apreciar onde tais ferramentas podem ser aplicadas no processo de engenharia de software. Segundo Pressman (1995, p.951), a taxonomia de ferramentas CASE foram categorizadas conforme mostra a Figura 1.

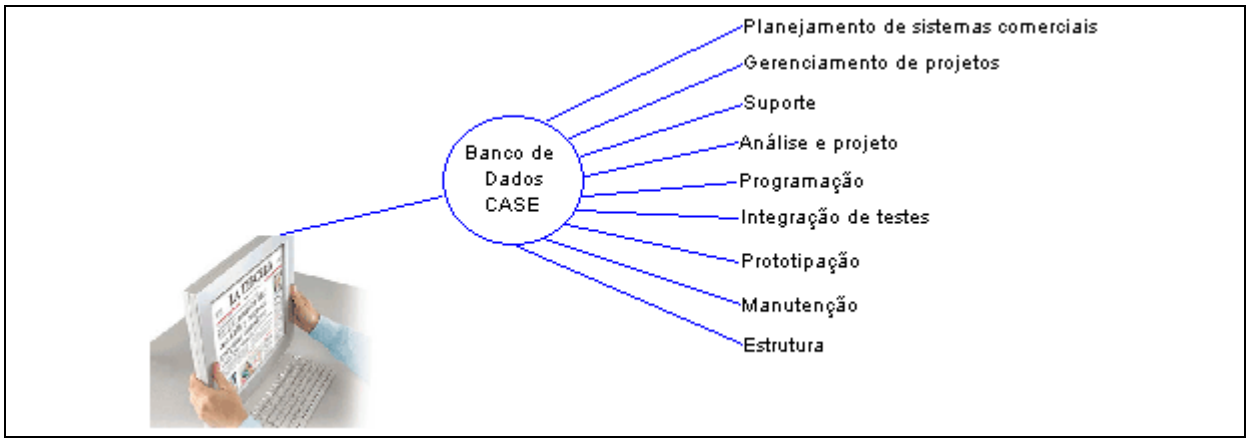


Figura 1 - Taxonomia CASE.

Dentre estas, encontram maior importância ao presente estudo as ferramentas de programação, prototipação e manutenção.

2.2.4.1 Ferramentas de Planejamento de Sistemas Comerciais

As ferramentas de planejamento de sistemas comerciais modelam as informações que fluem entre os vários setores e entidades dentro de uma organização ou empresa. A importância deste procedimento encontra-se no auxílio à construção de um novo sistema de informações, guiando para o gerenciamento mais eficiente destas informações.

2.2.4.2 Ferramentas de Gerenciamento de Projetos

Estas ferramentas auxiliam na construção de softwares. Segundo Pressmann (1995, p. 952), o gerente de projetos pode gerar úteis estimativas de esforço, custo e duração de um projeto de software, definir uma estrutura de divisão de trabalho, planejar uma programação viável de projeto e acompanhar projetos em base contínua. Este autor lembra, ainda, que estas ferramentas CASE podem rastrear os requisitos desde o requerimento original da proposta do cliente até o desenvolvimento do software que implantará esses requisitos.

2.2.4.3 Ferramentas de Suporte

Esta categoria abrange as ferramentas que complementam o processo de engenharia de software. Segundo Pressmann (1995, p. 955) nesta categoria incluem-se as ferramentas de documentação, as ferramentas de rede e de software básico, ferramentas de garantia de

qualidade, as ferramentas de gerenciamento de banco de dados e as ferramentas de gerenciamento de configuração.

2.2.4.4 Ferramentas de Análise de Projeto

As ferramentas de análise e projeto auxiliam na avaliação da qualidade do projeto. O modelo criado poderá representar o fluxo de controle e de dados, conteúdo de dados, representações de processo, especificações de controles e uma variedade e de outras representações de modelagem (PRESSMANN, 1995, p. 958).

2.2.4.5 Ferramentas de Programação

Nas ferramentas de programação encontram-se compiladores, editores e depuradores que se encontram à disposição para apoiar a maioria das linguagens de programação convencionais. Além destes, encontram-se nesta categoria os ambientes de programação orientados a objeto, as linguagens de quarta geração, os geradores de aplicações e as linguagens de consulta a banco de dados (PRESMANN, 1995, p. 961).

Segundo Pressmann (1995, p. 962), não há dúvida de que o objetivo final do CASE é a geração automática de código, a representação de sistemas em um nível mais elevado de abstração do que as linguagens de programação convencionais. Para este autor, estas ferramentas de geração de código também servem para ajudar o engenheiro de software a verificar a corretitude da especificação do sistema, de forma que a saída resultante esteja de acordo com os requisitos do usuário.

Os ambientes de programação orientados a objeto são aqueles ligados a uma linguagem de programação específica. Um exemplo típico de um ambiente orientado a objeto é aquele que incorpora características de *interface* de terceira geração (*mouse*, janelas, menus suspensos, operações contextuais e multitarefas), com funções especializadas como o *browser*. Segundo Pressmann (1995, p. 964) esta função auxilia o engenheiro de software a examinar os objetos contidos numa biblioteca de objetos verificando se um deles poderá ser recusado na aplicação.

2.2.4.6 Ferramentas de Integração de Testes

Quanto a esta modalidade de ferramentas CASE, Pressman (1995, p. 964) apresenta as categorias apresentadas pela Software Quality Engineering:

a) Aquisição de dados: ferramentas que adquirem dados a serem usados durante os testes.

b) Medição estática: ferramentas que analisam o código-fonte sem executar os casos de teste.

c) Medição dinâmica: ferramentas que simulam a função do hardware ou outros equipamentos externos.

d) Gerenciamento de teste: ferramentas que auxiliam no planejamento, desenvolvimento e controle dos testes.

e) Ferramentas transfuncionais: ferramentas que cruzam as fronteiras das categorias citadas.

As mais utilizadas são as ferramentas de análise estática, as ferramentas de análise dinâmica e as ferramentas de gerenciamento de teste.

2.2.4.7 Ferramentas de Prototipação

Segundo Pressman (1995, p. 968), as ferramentas CASE de prototipação mais sofisticadas possibilitam a criação de um projeto de dados, acoplados tanto com os *layouts* de relatório como com os de tela. Muitas ferramentas de análise e projeto têm extensões que fornecem uma opção de prototipação. Ainda, segundo este autor, as ferramentas PRO/SIM geram um arcabouço Ada e código-fonte C para aplicações (de tempo real) de engenharia. Finalmente, uma variedade de ferramentas de quarta geração tem características de prototipação.

As ferramentas de prototipação gradativamente se tornarão específicas quanto ao domínio, restringindo a sua área de aplicação e concedendo maior segurança à operação realizada.

2.2.4.8 Ferramentas de Manutenção

Aproximadamente 70% de todos os esforços relacionados a software dizem respeito às ferramentas CASE para manutenção, segundo Pressmann (1995, p. 969), que assim as classifica:

a) Ferramentas de engenharia reversa para especificação – capta o código-fonte como entrada e gera modelos gráficos de análise e projetos estruturados e também lista outras informações de projeto.

b) Ferramentas de análise e reestruturação de código – analisa a sintaxe do programa, gera um gráfico do fluxo de controle e gera automaticamente um programa estruturado.

c) Ferramentas de reengenharia de sistemas *on-line* – usadas para modificar sistemas de banco de dados *on-line*.

Este autor releva a importância das técnicas de inteligência artificial às ferramentas de reengenharia e de engenharia reversa da próxima geração, aplicando uma base de conhecimentos que é específica quanto ao domínio de aplicação. A vantagem do componente de inteligência artificial encontra-se no auxílio à decomposição e reconstrução do sistema, o que, ainda assim, não afastará o engenheiro de software ao longo do ciclo da reengenharia.

Para Pressman, as ferramentas de engenharia reversa dinâmicas monitoram o software quanto à sua execução e usam as informações obtidas durante a monitoração para construir um modelo comportamental do programa. Estas aplicações são raras, no entanto oferecem importantes informações àqueles que trabalham com sistemas embutidos ou software de tempo real.

Poucas ferramentas de qualidade industrial encontram-se em uso atualmente, apesar das vantagens que elas trazem, lembra Pressman. Este autor divide em duas subcategorias as ferramentas de reestruturação – as ferramentas de reestruturação de código e as ferramentas de reengenharia. E assim define as define: as ferramentas de reestruturação de código aceitam código-fonte não-estruturado como entrada, realizam a análise de engenharia reversa e depois reestruturam o código para que ele se conforme aos conceitos da moderna programação

estruturada. Ainda que tais ferramentas possam ser úteis, elas se concentram somente no desenho procedimental de um programa (PRESSMANN, 1995, p. 971).

As ferramentas de reengenharia de dados, por sua vez, avaliam as definições de dados, um banco de dados descrito numa linguagem de programação ou uma linguagem de descrição de banco de dados, sendo que, depois, as traduzem numa notação gráfica que poderá, então, ser analisada por um engenheiro de software.

2.2.4.9 Ferramentas de Estrutura

As ferramentas de estrutura fornecem gerenciamento de banco de dados, de configuração e capacidades de integração de ferramentas CASE. Segundo Pressmann (1995, p. 971), a maioria destas ferramentas vincula um banco de dados orientado a objeto com um conjunto de ferramentas internas para estabelecer *interfaces* harmoniosas com ferramentas de outros fornecedores CASE.

2.2.5 AMBIENTES CASE INTEGRADOS

Pressmann (1995, p. 979) assimila alguns benefícios no uso de CASE integrado (I-CASE):

a) a transferência harmoniosa de informações (modelos, programas, documentos, dados) de uma ferramenta para outra e de uma etapa da engenharia de software para a seguinte;

b) uma redução do esforço exigido para realizar atividades “guarda-chuva”, tais como gerenciamento de configuração de software, garantia de qualidade e produção de documentação;

c) um aumento no controle do projeto, que é obtido por meio de um melhor planejamento, monitoração e comunicação;

d) coordenação melhorada entre os membros de uma equipe que esteja trabalhando num grande projeto de software.

Segundo este autor, a definição de “integração”, no contexto de engenharia de software, necessita de um conjunto de requisitos. Portanto, um ambiente CASE integrado deve:

a) Oferecer um mecanismo para compartilhar as informações de engenharia de software entre todas as ferramentas contidas no ambiente.

b) Permitir que uma mudança efetuada num item de informação seja rastreada até outros itens de informação relacionados.

c) Oferecer controle de versão e gerenciamento de configuração global para todas as informações de engenharia de software.

d) Permitir o acesso direto, não seqüencial, a qualquer ferramenta contida no ambiente.

e) Estabelecer suporte automatizado para um contexto procedimental de trabalho de engenharia de software que integre as ferramentas e os dados num a estrutura de divisão de trabalho padrão.

f) Possibilitar que os usuários de cada ferramenta experimentem uma visão e uma percepção consistentes em nível de interface ser humano/computador.

g) Apoiar a comunicação entre os engenheiros de software.

h) Coletar tanto métricas administrativas com técnicas que possam ser utilizadas para melhorar o processo e o produto. (PRESSMANN, 1995, p. 981)

2.2.6 ARQUITETURA DE INTEGRAÇÃO EM AMBIENTE CASE

Para Pressmann (1995, p. 987), usando-se ferramentas CASE, métodos correspondentes e um *framework*, um *pool* de informações de engenharia de software é criado. O *framework* de integração auxilia na transferência para dentro e para fora do *pool*. Serão necessários os seguintes componentes de estruturação: um banco de dados, um sistema de gerenciamento de objetos, um mecanismo de controle de ferramentas, uma interface com o usuário com um *pathway* consistente entre as ações praticadas pelo usuário e as ferramentas contidas no ambiente.

2.2.7 ENGENHARIA NA WEB

Na elaboração de *Webs* e aplicativos relacionados, é imprescindível seguir uma disciplina, organizando a *Web* a ser visitada com coerência e consistência. Para Pressmann (2002, p. 754), as seguintes características guiam o referido processo:

1 - Imediatismo. Aplicações baseadas na *web* têm um imediatismo que não é encontrado em nenhum outro tipo de software. Isto é, o prazo de colocação no mercado de um site completo da *Web* pode ser questão de alguns dias ou semanas. Os desenvolvedores precisam usar métodos para planejamento, análise, projeto, implementação e teste, que tenham sido adaptados aos cronogramas comprimidos em tempo, requeridos para o desenvolvimento de *WebApp*.

2 - Segurança. Como as *WebApps* estão disponíveis através de acesso à rede, é difícil, se não impossível, limitar a população de usuários finais que podem ter acesso à aplicação. A fim de proteger o conteúdo reservado e fornecer modos seguros de transmissão de dados, fortes medidas de segurança precisam ser implementadas em toda a infra-estrutura que apóia uma *WebApp* e na aplicação propriamente dita.

3 - Estética. Uma inegável parte da atração de uma *WebApp* é o seu aspecto. Quando uma aplicação é projetada para o mercado ou para vender produtos ou idéias, a estética pode ter tanto a ver com o sucesso quanto o projeto técnico.

2.2.7.1 Estrutura da *WebApp*

São quatro os modelo de estrutura arquitetural de um WebApp, conforme a mesma se dispõe perante o usuário e liga um página à outra. Existem as estruturas lineares, em malha, hierárquicas e em rede (PRESSMANN, 2002, p. 762).

Estruturas lineares: são aquelas em que há uma seqüência de interações previsíveis, com páginas de informações em seqüência linear de apresentação.

Estruturas em malha: estas estruturas ocorrem quando além de uma estrutura linear, existe num grau de apresentação uma estrutura específica. Graficamente, pode-se dizer que há ramos horizontais numa estrutura linear vertical.

Estrutura hierárquica: ocorre quando há uma estrutura em pirâmide, podendo ocorrer ligações entre um ramo inferior e outro, permitindo assim uma navegação mais rápida. Esta estrutura é a mais comumente utilizada.

Estrutura em rede: esta estrutura firma-se numa verdadeira teia de ligações entre as diversas páginas, podendo numa página encontra-se hipertextos para as diversas páginas do conjunto sem qualquer cunho de hierarquia e seqüência entre as mesmas.

2.2.7.2 Projeto de Interface

O projeto que guiará a constituição da interface em aplicativos da *Web* deve respeitar algumas regras básicas, sob pena de afastar o usuário. Pressmann (2002, p. 766) lembra as diretrizes sugeridas por Nielsen e Wagner:

- Erros do servidor, mesmo de menor importância, provavelmente farão o usuário abandonar o site da *web* e procurar outro lugar de informação ou serviços.

- A velocidade de leitura no monitor é aproximadamente 25% mais lenta que a velocidade de leitura para material impresso. Por isso, não obrigue o usuário a ler quantidades volumosas de texto, particularmente quando o texto explica a operação da *WebApp* ou o auxilia na navegação.

- Evite sinais de “em construção” – eles provocam expectativas e causam uma ligação desnecessária que certamente irá desapontar.

- O usuário prefere não rolar a imagem. Uma informação importante deve ser colocada dentro das dimensões de uma janela de busca típica.

- Menus de navegação e barras de título devem ser projetados consistentemente e estar disponíveis em todas as páginas disponibilizadas ao usuário. O projeto não deve se basear em funções de busca para ajudar na navegação.

- A estética nunca deve sobrepujar a funcionalidade.

- As opções de navegação ou ícone devem ser óbvias, mesmo para um usuário ocasional. O usuário não deve ser obrigado a procurar na tela para determinar como se ligar com outro conteúdo ou serviço.

3 DESENVOLVIMENTO DO TRABALHO

Neste capítulo é apresentada a ferramenta CASE para definição, documentação e geração de páginas de cadastros em linguagem de script ASP, assim como sua implementação e etapas de desenvolvimento.

3.1 REQUISITOS PRINCIPAIS DO PROBLEMA A SER TRABALHADO

A ferramenta CASE tem a finalidade de definir, documentar e armazenar informações necessárias para a geração do banco de dados com os relacionamentos entre as tabelas, assim garantindo a integridade referencial. Após a geração do banco de dados, será possível a geração de código da parte para a administração das informações do *site* com cadastros completos incluindo opções de inclusão, alteração, exclusão, localizar e consultar por campo, paginação dos registros e hierarquia de campos.

3.2 ESPECIFICAÇÃO

Essa ferramenta foi desenvolvida com o auxílio do *Macromedia Dreamweaver MX 2004* e com as definições e documentações armazenadas em banco de dados *SQL Server* para a geração do código ASP.

Para validar e testar soluções para problemas reais, foi realizada a implementação de um sistema, exercitando conhecimentos de banco de dados relacionais e desenvolvimento *web* através do levantamento de requisitos e da narrativa do estudo de caso. Para especificação desse sistema utilizou-se dos Diagramas de Caso de Uso, Diagrama de Contexto, Modelo de Entidade Relacional (MER), Diagrama de Atividades e Diagrama de Implantação.

3.2.1 DESCRIÇÃO DO CASO

Pretende-se criar uma ferramenta para auxílio dos programadores *web*, para geração de sistemas completos de cadastros, podendo gerar aplicações completas como a administração e para o *front-end* do site, baseando-se em informações predefinidas e armazenadas em um banco de dados físico e relacional.

Um administrador poderá definir o nome do banco de dados, tabelas, campos e os relacionamentos que serão gerados. A ferramenta vai gerar os códigos fontes integrados com os bancos de dados *SQL Server* ou *Access* conforme o administrador predefinir.

Na administração do site, serão geradas telas de formas simplórias para facilitar a manipulação dos dados que estão armazenados no banco de dados. Esta manipulação de informações do banco está incluída a opção: localizar por campo, paginação dos registros, inserção, alteração, exclusão de registros.

3.2.2 LEVANTAMENTO DE REQUISITOS

A atividade de levantamento de requisitos corresponde à etapa de compreensão do problema aplicada ao desenvolvimento de software. O principal objetivo do levantamento de requisitos é que usuários e desenvolvedores tenham a mesma visão do problema a ser resolvido (BEZERRA, 2002, p. 20). O levantamento de requisitos divide-se em três tipos: funcionais, não-funcionais e de restrições.

Os requisitos funcionais definem as funcionalidades do sistema. Esta ferramenta deve utilizar dos seguintes requisitos funcionais:

a) a ferramenta deve permitir que qualquer administrador crie um projeto;

b) a ferramenta deve permitir que o administrador defina o nome do banco de dados, tabelas, campos e usuários do banco de dados;

c) a ferramenta deve permitir que o administrador configure todo o projeto, tabelas e também personalize todos os campos dos formulários de cadastros;

d) a ferramenta deve criar um banco de dados relacional, gerar páginas para a administração e *front-end* do site.

Os requisitos não-funcionais declaram as características de qualidade que o sistema deve possuir e que estão relacionadas às suas funcionalidades. Esta ferramenta deve utilizar dos seguintes requisitos não-funcionais:

a) confiabilidade: deve ser configurado um arquivo no *IIS* do servidor para que os administradores sejam comunicados por e-mail de erros que possam ocorrer nos scripts e assim resolvermos rapidamente o ocorrido;

b) desempenho: a ferramenta CASE deve ter alto desempenho em sua navegação e em outras funcionalidades. Na geração do código, que é a funcionalidade mais trabalhosa na execução deste processo no servidor onde está hospedado a ferramenta CASE, o tempo normal para a geração é de até cinco minutos. No aplicativo gerado, a navegação deve ser bastante rápida, devido ao código fonte gerado possuir funções simples e enxutas;

c) portabilidade: a ferramenta deve ser hospedada em um servidor *web* com o sistema operacional *Microsoft Windows 2000 Server* ou *Microsoft Windows 2003 Server* e com o *Internet Information Service* (IIS) instalado e devidamente configurado. O servidor *web* também deve ter disponibilidade para o uso dos bancos de dados *SQL Server* e *Access*, assim como os seus *drivers* do ODBC instalados;

d) segurança: a ferramenta deve possuir acesso restrito para usuários não autorizados com a utilização de uma tela de autenticação com *login* e *senha*. A ferramenta também deve permitir na geração da administração do site a opção para restrição de usuários não autorizados. Em todas as páginas da ferramenta CASE e de suas páginas geradas deverá ser incluído um arquivo: seguranca.asp, que restringe o acesso direto de usuários não autorizados a qualquer link;

e) usabilidade: O sistema deve interagir com o usuário mostrando mensagens de instruções para navegação, tanto na ferramenta CASE, como no sistema gerado.

Os requisitos de restrições impostas sobre o desenvolvimento do sistema. Esta ferramenta utiliza-se dos seguintes requisitos de restrição:

a) deve possuir um servidor *web* com sistema operacional *Microsoft Windows 2000 Server* ou *Microsoft Windows 2003 Server* e com o *IIS* instalado e devidamente configurado;

b) servidor *web* também deve ter a disponibilidade para o uso dos bancos de dados *SQL Server* ou *Access*, assim como os seus *drivers* do ODBC instalados;

c) a ferramenta deve ser testada no *browser* da *Microsoft*, o *Internet Explorer 6.0*., podendo o sistema não funcionar corretamente em qualquer outro *browser*.

3.2.3 DIAGRAMA DE CASO DE USO

A Figura 2 mostra o digrama de caso de uso da ferramenta do ponto de vista do administrador que irá gerar os sistemas.

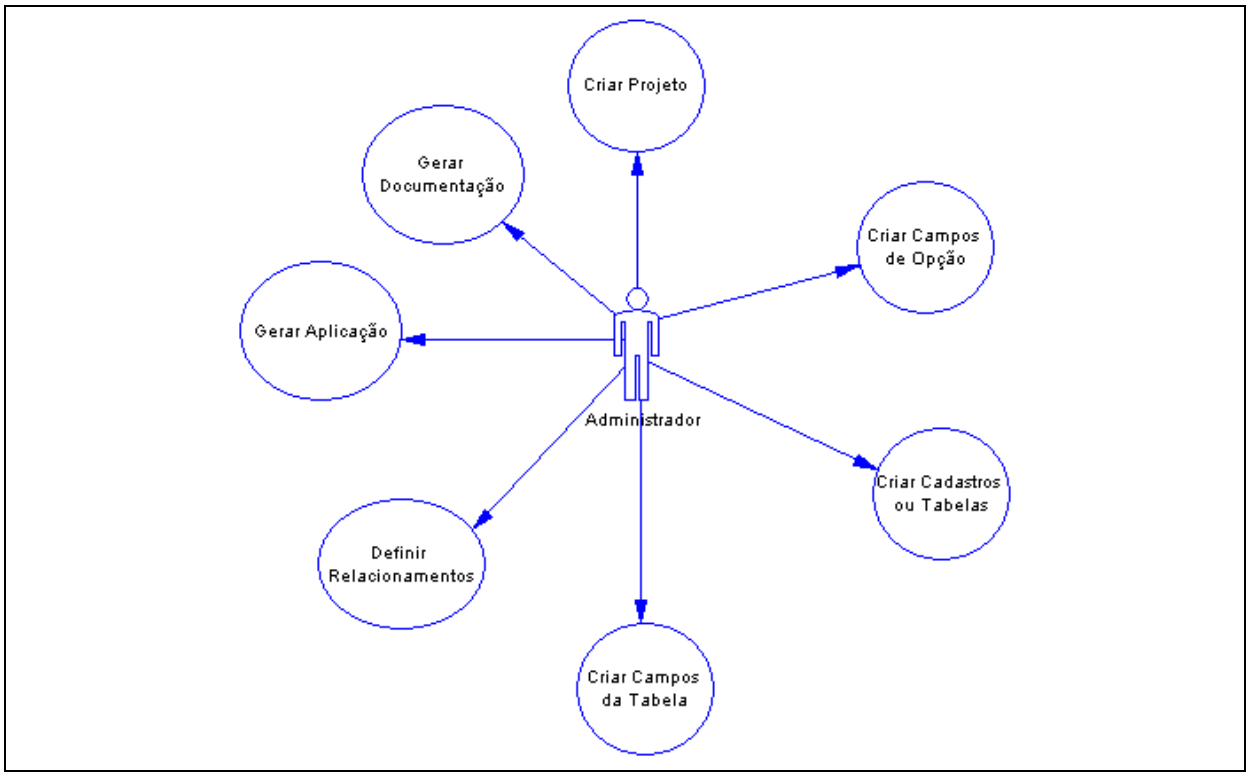


Figura 2 - Diagrama de Caso de Uso.

Na Tabela 2 encontra-se a descrição dos Casos de Uso.

Tabela 2 - Descrição dos Casos de Uso

Caso de Uso: **CRIAR PROJETO** **Sumário:** O administrador usa a ferramenta para criar o projeto dando início ao um novo aplicativo e também é utilizada das informações do projeto para a geração de código. **Ator Principal:** Administrador **Precondições:** Nenhuma. **Fluxo Principal:**

a) O administrador cadastra o projeto definindo o nome e o banco de dados que será utilizado.

b) O administrador também define no cadastro do projeto, informações relevantes à conexão da futura aplicação com o banco de dados já existente ou que ainda será gerado.

c) As informações do projeto são armazenadas na base de dados da ferramenta.

Caso de Uso: **CRIAR CAMPOS DE OPÇÃO**

Sumário: O administrador usa a ferramenta para definir campos de opção que são campos utilizados para criar as opções como ***combobox*** e ***checkbox*** na geração dos formulários de cadastro.

Ator Principal: Administrador

Precondições: Nenhuma.

Fluxo Principal:

a) O administrador cria campos de opção para serem utilizadas na criação dos campos das tabelas.

b) São armazenadas as informações dos campos de opção para a geração de código.

Caso de Uso: **CRIAR CADASTROS/TABELAS**

Sumário: O administrador usa a ferramenta para criar cadastros/tabelas onde serão utilizadas das informações para a geração de código.

Ator Principal: Administrador

Precondições: Deve existir um projeto criado.

Fluxo Principal:

a) O administrador cria o cadastros/tabelas informando o nome do cadastro e nome da

tabela que será criada.

b) O administrador também define na criação dos cadastros/tabelas, a descrição do cadastro e da tabela para a documentação e o número de registros listados por página na aplicação que será gerada.

c) O administrador pode definir na criação do cadastro, se o cadastro é utilizado como um formulário de envio, e se isto ocorrer, será informado o título do e-mail e o e-mail para envio das informações do formulário.

d) As informações dos cadastros/tabelas são gravadas no banco de dados se o administrador selecionar a opção **Utiliza banco de dados** na criação do cadastro/tabela.

Caso de Uso: **CRIAR CAMPOS DA TABELA**

Sumário: O administrador usa a ferramenta para criar campos da tabela onde serão utilizadas das informações para a geração de código.

Ator Principal: Administrador

Precondições: Deve existir um projeto criado e pelo menos um cadastro/tabela com opção de utilizar banco de dados.

Fluxo Principal:

a) O administrador cria os campos da tabela informando dados essenciais.

b) O administrador também define se o campo é do tipo seleciona (***CheckBox***) ou escolha (***Radio Button***), onde tem acesso e opção de escolher os campos de opção.

c) As informações dos campos das tabelas são gravadas no banco de dados da ferramenta.

Caso de Uso: **DEFINIR RELACIONAMENTOS**

Sumário: O administrador usa a ferramenta para definir os relacionamentos sendo utilizado na geração da aplicação na criação do banco de dados relacional.

Ator Principal: Administrador

Precondições: Deve existir um projeto criado e pelo menos dois cadastros/tabelas com opção de utilizar banco de dados.

Fluxo Principal:

a) O administrador define os relacionamentos entre as tabelas definindo campo principal e secundário.

b) O administrador também define qual campo que será o identificador do cadastro secundário no cadastro principal.

c) As informações das definições dos relacionamentos são gravadas no banco de dados da ferramenta.

Caso de Uso: **GERAR APLICAÇÃO**

Sumário: O administrador usa a ferramenta para gerar a aplicação, tanto a "Administração do Site", como o seu "*Front-end*".

Ator Principal: Administrador

Precondições: Deve existir um projeto criado e pelo menos dois cadastros/tabelas com opção de utilizar banco de dados.

Fluxo Principal:

a) O administrador solicita a geração do banco de dados, da administração e do *front-end* do *site*.

b) O sistema gera o banco de dados com toda a estrutura relacional.

c) O sistema gera a administração do site com escolha para autenticação com *login* e *senha*, gerando o menu da aplicação baseado nos nomes dos cadastros/tabelas.

d) O sistema gera o *front-end* do *site* listando as informações cadastradas na "Administração do Site" com paginação e um sistema de localizar por campo.

Caso de Uso: **GERAR DOCUMENTAÇÃO**

Sumário: O administrador usa a ferramenta para gerar a documentação completa da aplicação. Sendo utilizado pelo administrador para ter uma visão ampla do sistema e poder melhorá-lo cada vez mais.

Ator Principal: Administrador

Precondições: Deve existir um projeto com o máximo de informações cadastradas e informadas a ferramenta.

Fluxo Principal:

a) O administrador solicita a geração da documentação da aplicação.

b) O programa gera a documentação completa da aplicação que foi ou será gerada.

c) As informações da documentação são consultas efetuadas em todas as tabelas existentes do projeto, trazendo informações relevantes para a decisão do administrador do sistema.

3.2.4 DIAGRAMA DE CONTEXTO

A Figura 3 mostra o diagrama de contexto, visto do ponto de vista do desenvolvedor, que irá gerar o sistema.

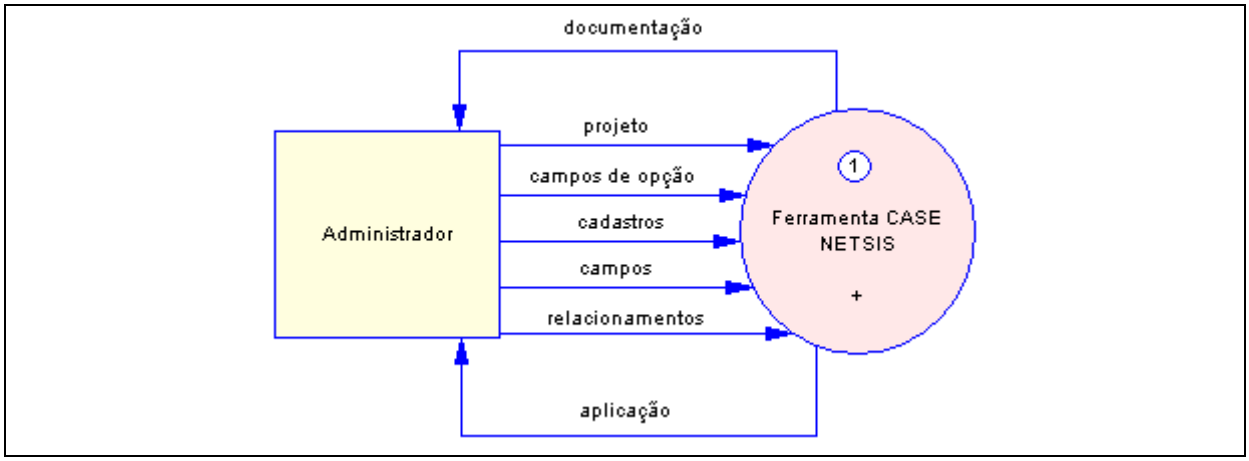


Figura 3 - Diagrama de Contexto.

3.2.5 DIAGRAMA – MODELO CONCEITUAL

A Figura 4 mostra o Diagrama Entidade Relacionamento (DER) conceitual, sendo que todas as informações necessárias para a geração de código estarão armazenadas nestas tabelas.

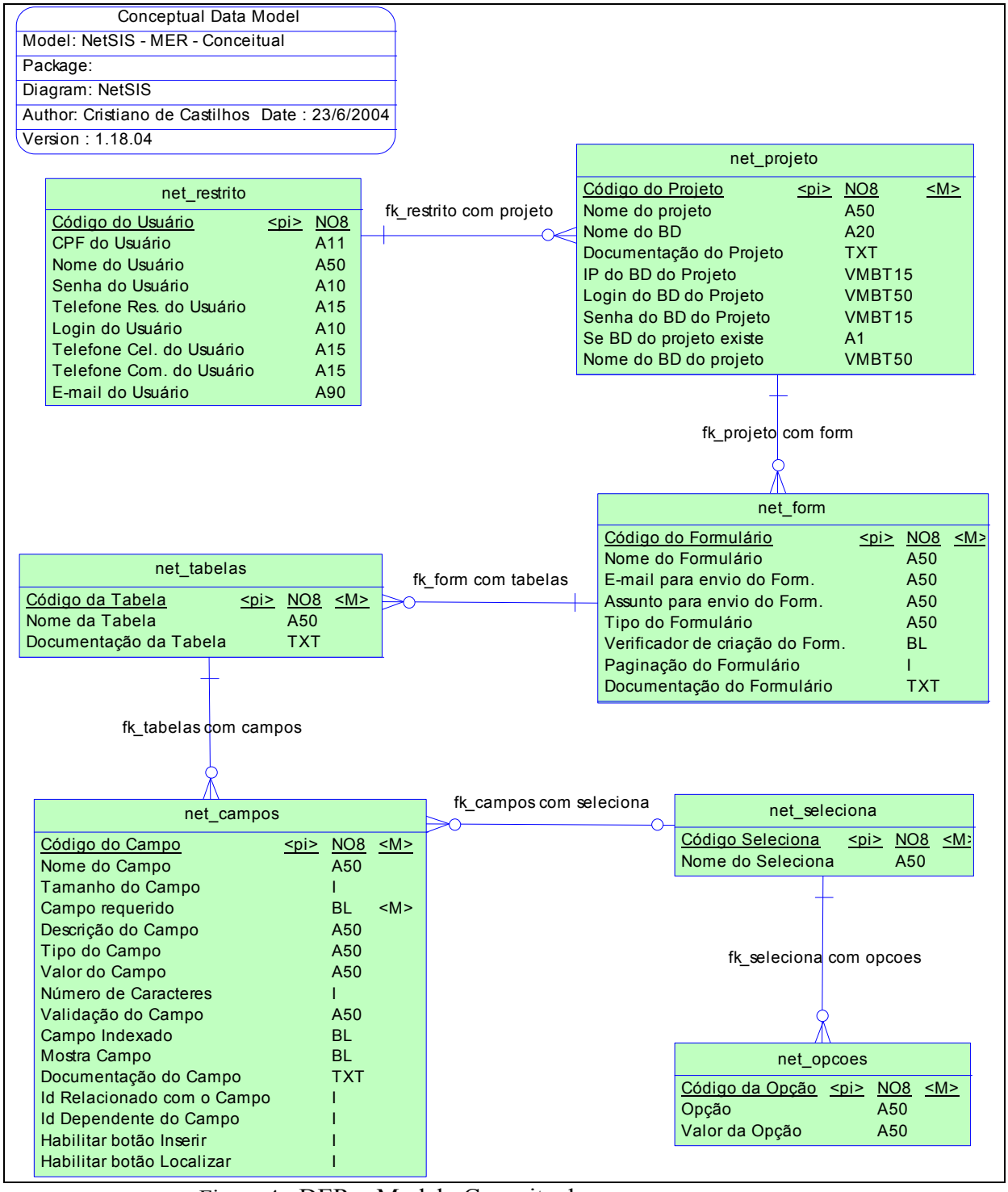


Figura 4 - DER – Modelo Conceitual.

3.2.6 DIAGRAMA – MODELO FÍSICO

A Figura 5 mostra o Diagrama Entidade Relacionamento (DER) físico.

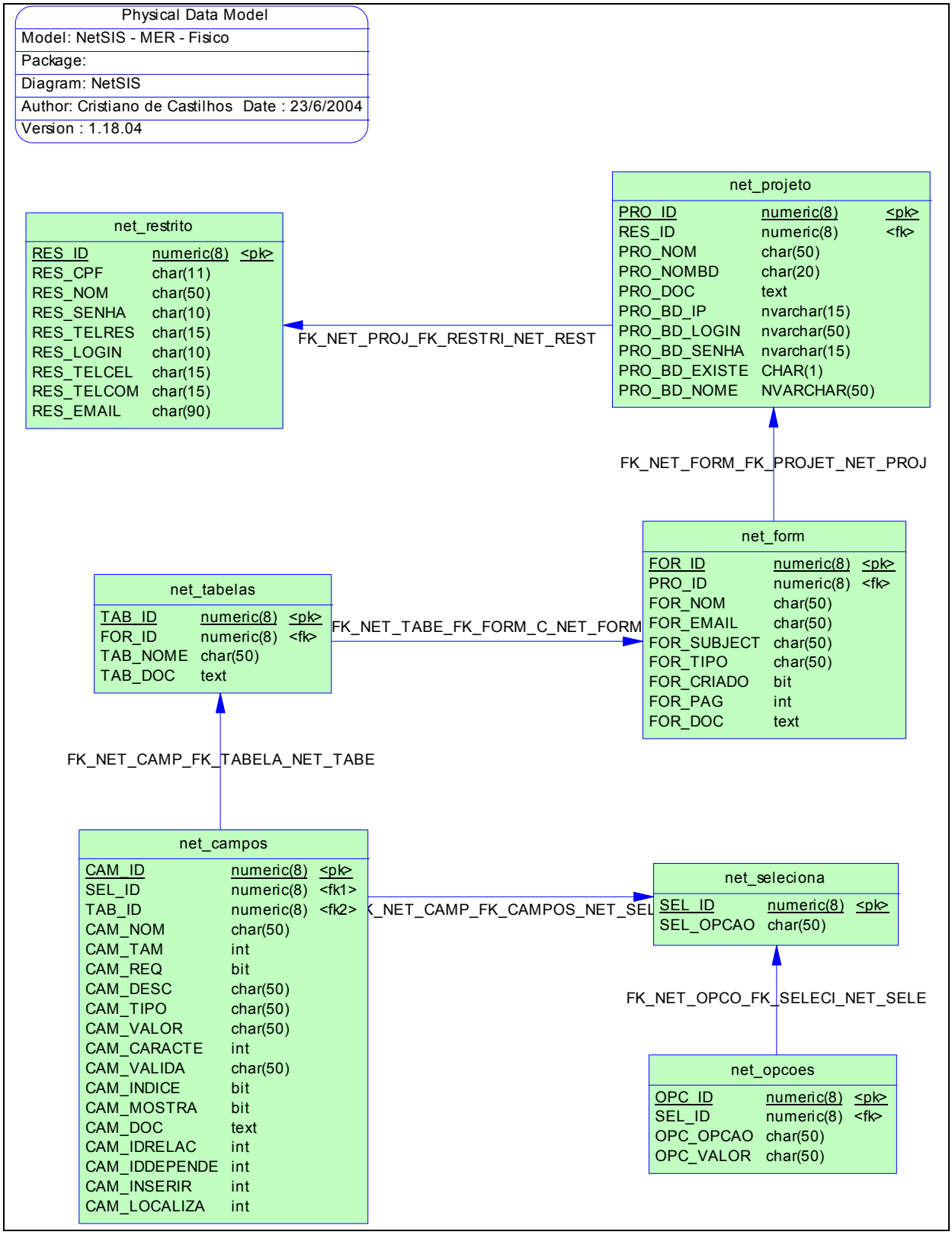


Figura 5 - DER – Modelo Físico.

3.2.7 DICIONÁRIO DE DADOS

A seguir encontram-se os dicionários de dados classificados em tabelas, onde são listadasas informações dos campos como: o código, nome, tipo, chave primária (PK), chave secundário (FK) e se o campo é requerido/obrigatório.

Na Tabela 3 está listada informações dos atributos da tabela **net_campos.** Esta tabela serve para armazernar informações dos campos das tabelas do projeto.

Tabela 3 - Dicionário de dados – Tabela: net_campos

Código	*Nome*	*Tipo*	*PK*	*FK*	*Req.*
CAM_ID	Código do Campo	NO8	X		X
TAB_ID	Código da Tabela	NO8		X	X
SEL_ID	Código Seleciona	NO8		X	
CAM_NOM	Nome do Campo	A50			
CAM_TAM	Tamanho do Campo	I			
CAM_REQ	Campo requerido	BL			X
CAM_DESC	Descrição do Campo	A50			
CAM_TIPO	Tipo do Campo	A50			
CAM_VALOR	Valor do Campo	A50			
CAM_CARACTE	Número de Caracteres	I			
CAM_VALIDA	Validação do Campo	A50			
CAM_INDICE	Campo Indexado	BL			
CAM_MOSTRA	Mostra Campo	BL			
CAM_DOC	Documentação do Campo	TXT			
CAM_IDRELAC	Id Relacionado com o Campo	I			
CAM_IDDEPENDE	Id Dependente do Campo	I			
CAM_INSERIR	Habilitar botão Inserir	I			
CAM_LOCALIZA	Habilitar botão Localizar	I			

Na Tabela 4 está listada informações dos atributos da tabela **net_form**Esta tabela serve para armazernar informações dos cadastros/tabelas do projeto.

Tabela 4 - Dicionário de dados – Tabela: net_form

Código	*Nome*	*Tipo*	*PK*	*FK*	*Req.*
FOR_ID	Código do Formulário	NO8	X		X
PRO_ID	Código do Projeto	NO8		X	X
FOR_NOM	Nome do Formulário	A50			
FOR_EMAIL	E-mail para envio do Form.	A50			
FOR_SUBJECT	Assunto para envio do Form.	A50			
FOR_TIPO	Tipo do Formulário	A50			
FOR_CRIADO	Se foi criado o Form.	BL			

FOR_PAG	Paginação do Formulário	I			
FOR_DOC	Documentação do Formulário	TXT			

Na Tabela 5 está listada informações dos atributos da tabela **net_opcoes.** Esta tabela serve para armazernar as **opções** dos campos de opção do projeto.

Tabela 5 - Dicionário de dados – Tabela: net_opcoes

Código	***Nome***	***Tipo***	***PK***	***FK***	***Req.***
OPC_ID	Código da Opção	NO8	X		X
SEL_ID	Código Seleciona	NO8		X	X
OPC_OPCAO	Nome da Opção	A50			
OPC_VALOR	Valor da Opção	A50			

Na Tabela 6 está listada informações dos atributos da tabela **net_projeto .** Esta tabela serve para armazernar informações do projeto.

Tabela 6 - Dicionário de dados – Tabela: net_projeto

Código	***Nome***	***Tipo***	***PK***	***FK***	***Req.***
PRO_ID	Código do Projeto	NO8	X		X
RES_ID	Código do Usuário	NO8		X	X
PRO_NOM	Nome do Projeto	A50			
PRO_NOMBD	Nome do BD utilizado	A20			
PRO_DOC	Documentação do Projeto	TXT			
PRO_BD_IP	IP do BD do Projeto	VMBT15			
PRO_BD_LOGIN	Login do BD do Projeto	VMBT50			
PRO_BD_SENHA	Senha do BD do Projeto	VMBT15			
PRO_BD_EXISTE	Se o BD existe no Projeto	A1			
PRO_DB_NOME	Nome do banco de dados	VMBT50			

Na Tabela 7 está listada informações dos atributos da tabela **net_restrito.** Esta tabela serve para armazernar informações dos usuários da ferramenta.

Tabela 7 - Dicionário de dados – Tabela: net_restrito

Código	***Nome***	***Tipo***	***PK***	***FK***	***Req.***
RES_ID	Código do Usuário	NO8	X		X
RES_CPF	CPF do Usuário	A11			X
RES_NOM	Nome do Usuário	A50			
RES_SENHA	Senha do Usuário	A10			
RES_TELRES	Telefone Res. do Usuário	A15			
RES_LOGIN	Login do Usuário	A10			
RES_TELCEL	Telefone Cel. do Usuário	A15			
RES_TELCOM	Telefone Com. do Usuário	A15			
RES_EMAIL	E-mail do Usuário	A90			

Na Tabela 8 está listada informações dos atributos da tabela **net_seleciona.** Esta tabela serve para armazernar os campos de opção.

Tabela 8 - Dicionário de dados – Tabela: net_seleciona

Código	*Nome*	*Tipo*	*PK*	*FK*	*Req.*
SEL_ID	Código Seleciona	NO8	X		X
SEL_OPCAO	Nome do Seleciona	A50			

Na Tabela 9 está listada informações dos atributos da tabela **net_tabelas.** Esta tabela serve para armazernar informações das tabelas do projeto.

Tabela 9 - Dicionário de dados – Tabela: net_tabelas

Código	*Nome*	*Tipo*	*PK*	*FK*	*Req.*
TAB_ID	Código da Tabela	NO8	X		X
FOR_ID	Código do Formulário	NO8		X	X
TAB_NOME	Nome da Tabela	A50			
TAB_DOC	Documentação da Tabela	TXT			

3.2.8 DIAGRAMA DE ATIVIDADES

A Figura 6 mostra o Diagrama de Atividades para exemplificar o Diagrama de Caso de Uso da Figura 2.

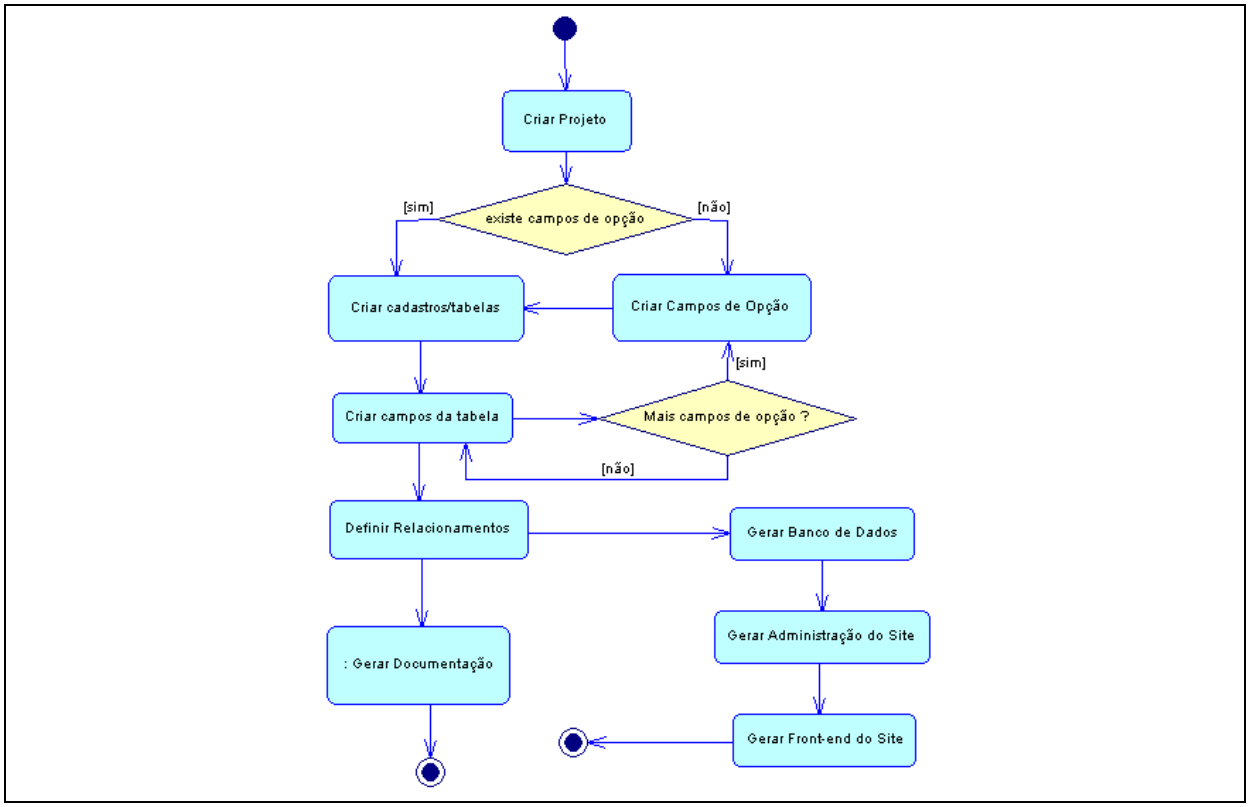


Figura 6 - Diagrama de Atividades.

3.2.9 DIAGRAMA DE IMPLANTAÇÃO

A Figura 7 é mostrado o Diagrama de Implantação.

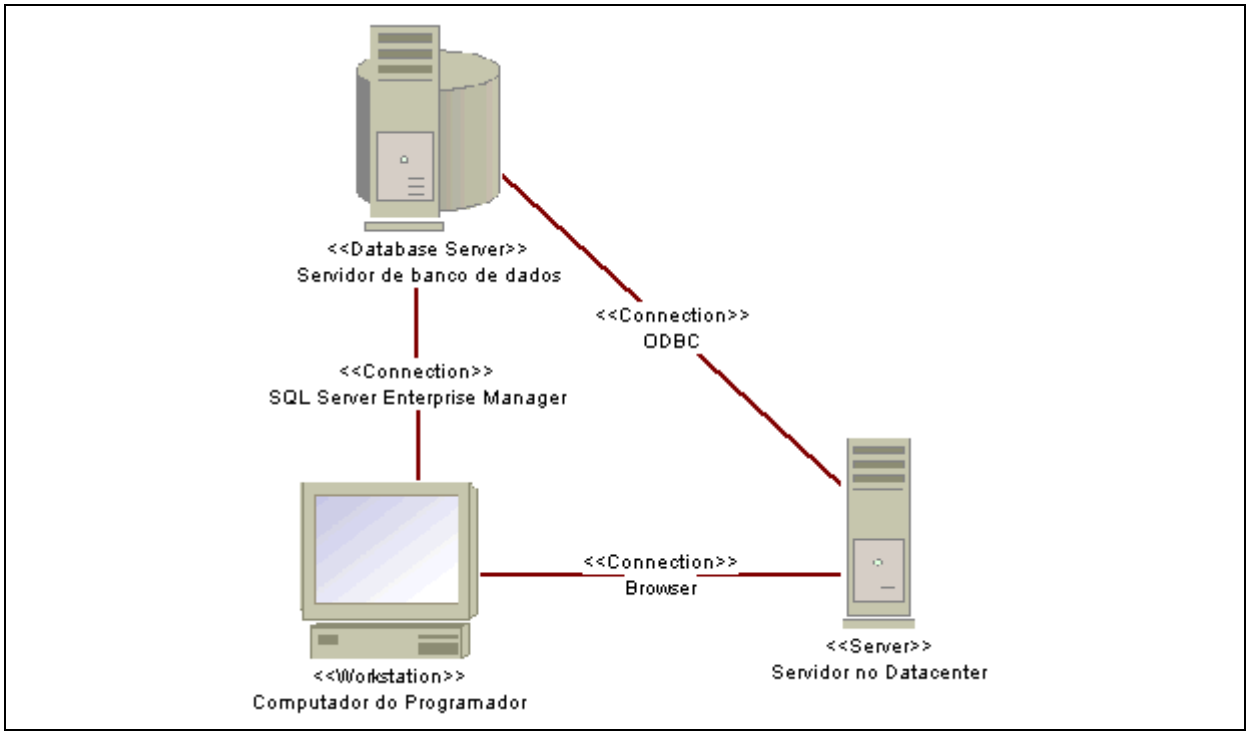


Figura 7 - Diagrama de Implantação.

3.3 IMPLEMENTAÇÃO

Para exemplificar o funcionamento da ferramenta, foi implementado um sistema de imóveis para a validação e testes da mesma.

3.3.1 TÉCNICAS E FERRAMENTAS UTILIZADAS

Esta ferramenta CASE para a geração de páginas em ASP foi desenvolvida no ambiente de desenvolvimento web *Macromedia Dreamweaver MX 2004* conforme é demonstrado na Figura 8.

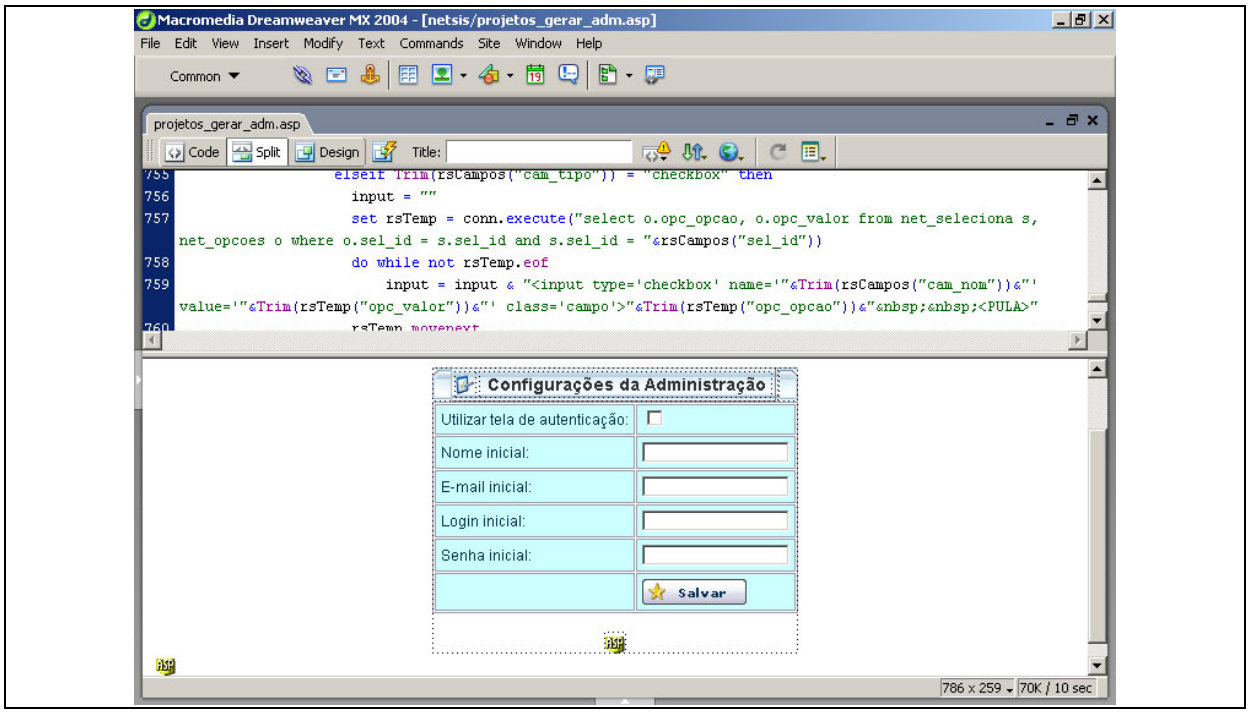


Figura 8 - Macromedia Dreamweaver MX 2004.

Na Figura 9 é mostrado o *SQL Server* que foi o principal banco de dados para o armazenamento das definições e informações para a geração de código.

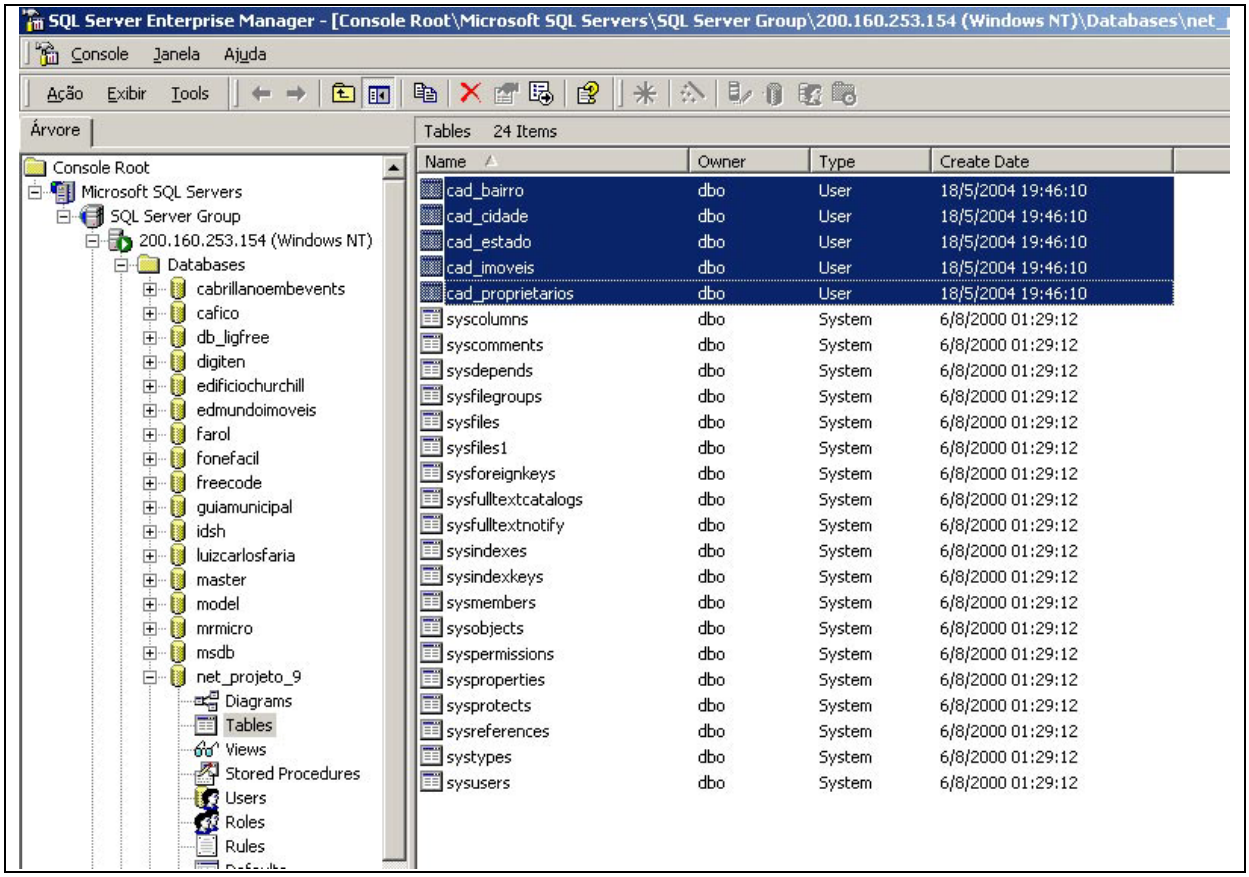


Figura 9 - SQL Server Enterprise Manager.

As linguagens utilizadas foram: *HTML*, *ASP*, *JavaScript* e *VBScript* e sendo que o *IIS* do *Windows 2000* foi utilizado como servidor *web* para executar os *scripts* ASP.

Foram utilizados recursos disponíveis na ferramenta *PowerDesigner Trial 10* que foi obtido no endereço <http://www.sybase.com> e permitiu a criação da estrutura e da integridade referencial do banco de dados.

Todos os testes de desenvolvimento do sistema foram efetuadas utilizando o *Microsoft Internet Explorer 6*.

3.3.2 OPERACIONALIDADE DA IMPLEMENTAÇÃO

Este item trata da operacionalidade da implementação e as telas demonstradas fazem parte da exemplificação da ferramenta em modo de tutorial. Passo a passo é mostrada a criação de um aplicativo de imóveis com cadastros de imóveis, tipos de imóveis, proprietários, corretores, interessados, registros de contatos, estados, cidades e bairros. Cada cadastro possui a opção para fazer a localização do registro por campo e possui paginação em sua listagem. O aplicativo de imóveis terá opções para incluir, alterar e excluir qualquer informação nos cadastros.

3.3.3 FUNCIONALIDADE DA FERRAMENTA

É uma ferramenta de auxílio aos programadores *web* e analistas de sistemas*,* para a geração de sistemas completos de cadastros para gerenciamento das informações armazenadas em banco de dados relacional. Neste capítulo é disposto de forma clara, passo em passo, na forma de um tutorial a funcionalidade da ferramenta.

O administrador cria um projeto e informa qual o nome do projeto, descrição, tipo do banco de dados, se o banco de dados já existe ou deve ser criado, IP do servidor do banco de dados, nome do banco de dados, o seu *login* e sua *senha,* conforme Figura 10.

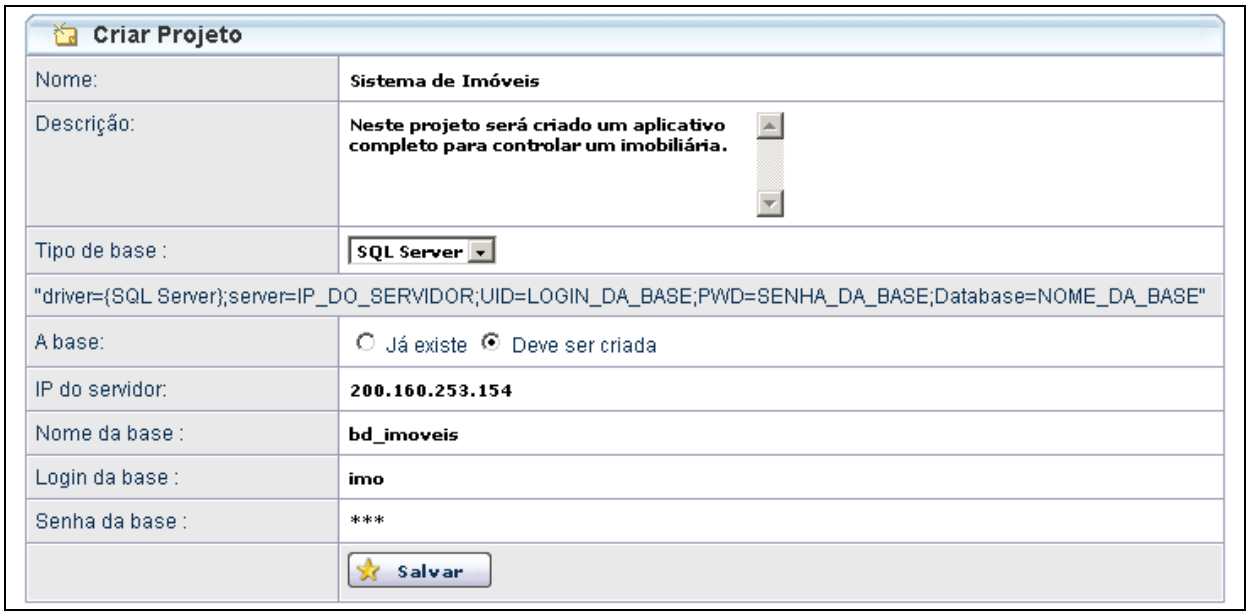


Figura 10 - Criação do projeto.

O administrador cria campos com escritas repetitivas que não serão indexadas em consultas e podem ser criadas no item **Campos de Opção** conforme Figura 11.

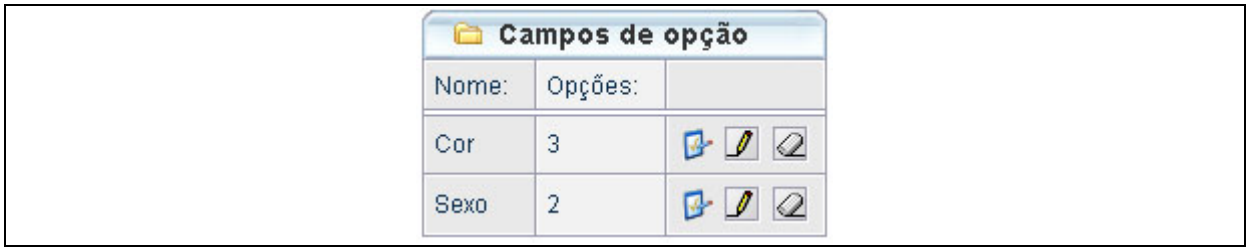


Figura 11 - Listagem dos Campos de Opção.

No item **Campos de Opção** da ferramenta encontra-se no nome do campo **Sexo** dois subitens cadastrados conforme a Figura 12, as opções: Feminino e o Masculino. Se o administrador criar um campo do tipo *bit,* poderá ser definido os valores dos campos Feminino e Masculino tanto como 1 (um) ou 0 (zero), que é o valor a ser armazenado no banco de dados por um campo tipo *bit*.

Figura 12 - Campos de Opção da ferramenta.

Após definir informações do projeto, o administrador começar a definir informações referentes ao cadastro a ser criado conforme Figura 13.

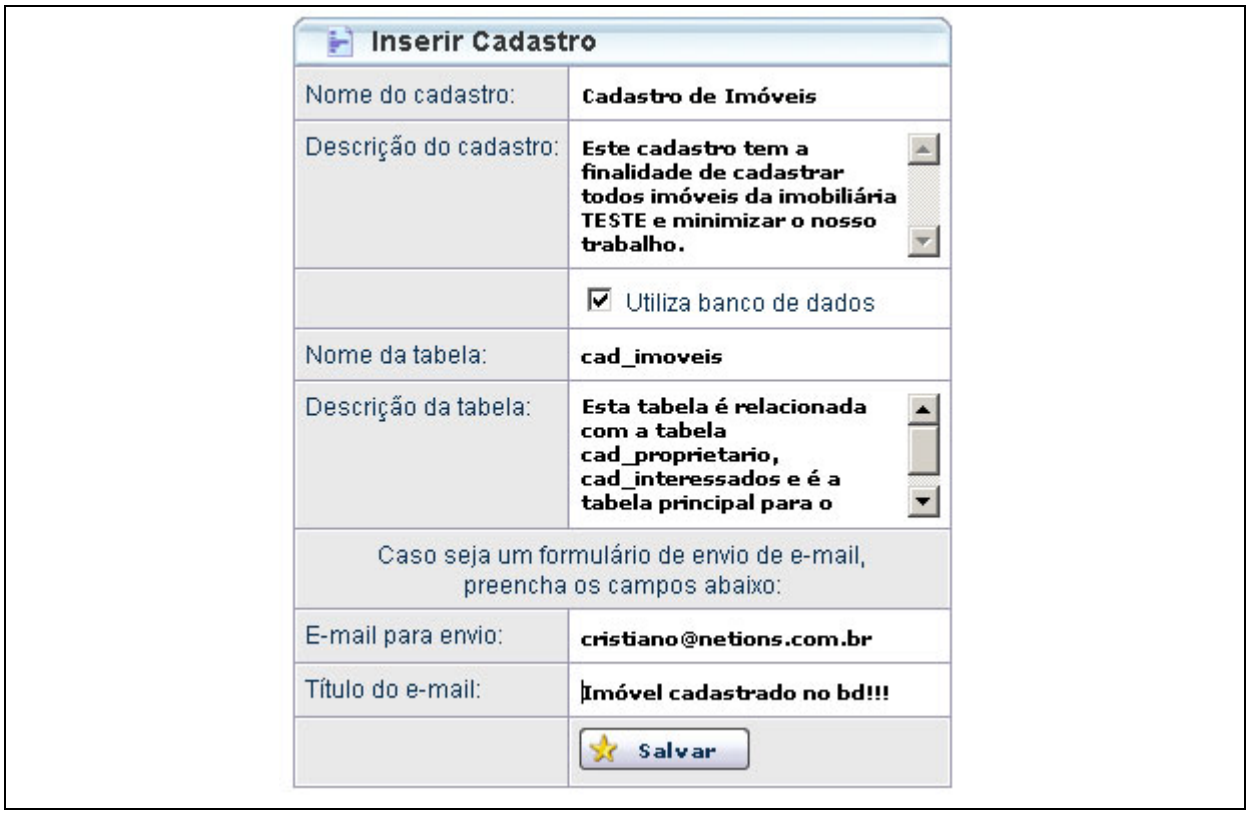


Figura 13 - Inserir cadastro.

Quando o administrador insere um novo cadastro, automaticamente é criado um campo auto-numérico definido como índice conforme Figura 14.

Figura 14 - Campo auto-numérico definido como índice.

Para a criação do campo do tipo **Texto**, o administrador define se o campo vai ser índice, se ele é requerido, o tamanho, o número máximo de caracteres e define a validação do campo. Na validação do campo texto pode-se optar pela opção **Número**, onde após a criação do formulário só será permitido a digitação de números neste campo. Caso seja escolhida a opção de **E-mail**, o formulário fará a consistência da digitação de uma arroba (@) no e-mail. Também é efetuada a validação da digitação do **CPF** e **CNPJ**. Demonstram-se detalhes na Figura 15.

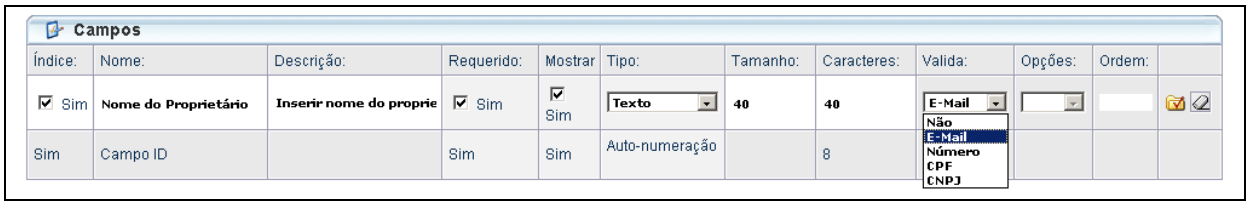


Figura 15 - Criação de campo do tipo Texto.

Na Figura 16 demonstra que a ferramenta também se pode criar campos de escolha e seleciona. No tipo de campo **Escolha**, o administrador poderá escolher uma das opções propostas e quanto no tipo de campo **Seleciona** o administrador poderá escolher mais de uma das opções propostas na definição dos campos de opção. Pode ser verificado na Figura 20 que também é disponível o tipo de campo **Memorando** sendo normalmente utilizado em formulários de dúvidas, perguntas, sugestões e descrição de produtos ou serviços.

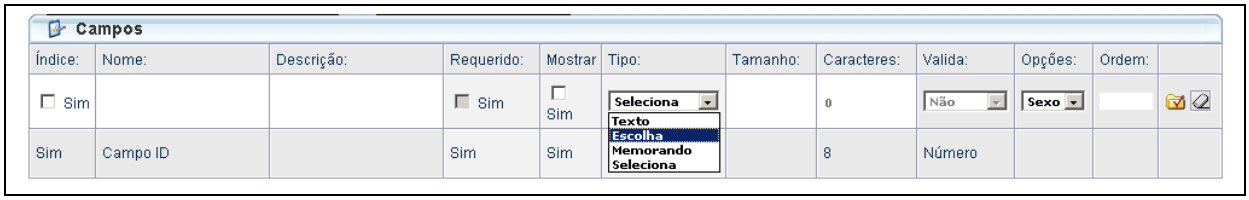


Figura 16 - Criação de um campo do tipo escolha ou seleciona.

Para o relacionamento entre os cadastros ou tabelas e assim garantir a integridade referencial, o administrador poderá definir o **Cadastro principal** como exemplo **Bairros**. Este **Cadastro principal** depende de algum outro cadastro, que é o **Campo secundário**. Por exemplo, o cadastro de bairros depende do cadastro de cidades, e conforme a Figura 21, o cadastro de bairros recebe o campo identificador do cadastro de cidades para definir a relação entre os dois cadastros.

Nas opções: **mostrar botão inserir** e **mostrar botão localizar** o administrador pode habilitar e desabilitar a geração destes botões. Será visualizado claramente mais a frente na Figura 26 os botões que foram gerados pela ferramenta. Mais detalhes é demonstrado na Figura 17.

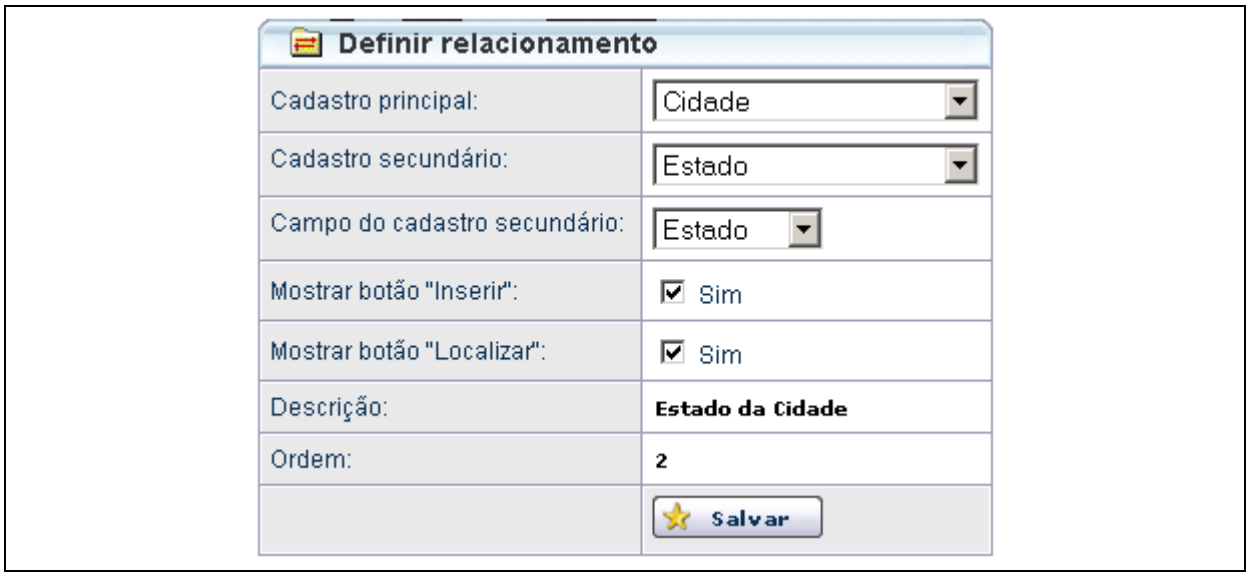


Figura 17 - Definição de um relacionamento entre os cadastros.

Após a criação de todos os relacionamentos, os campos são definidos automaticamente com o tipo definido **Relacionamento**. Na Figura 18, o campo **Cidade**, na coluna **Valida** está descrito em negrito o **Estado.** Quando é gerado o formulário para este cadastro, o campo **Estado** deverá preceder o campo **Cidade**.

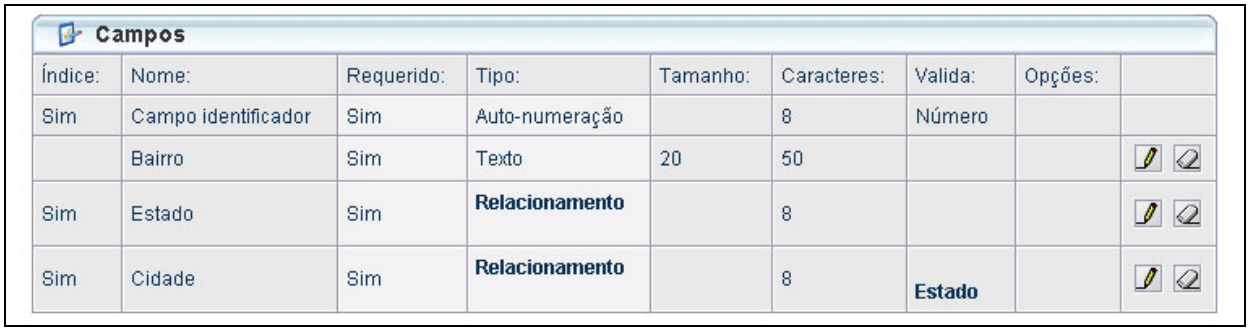


Figura 18 - Campos relacionados.

Pode se chamar de cadastros, formulários ou tabelas a listagem apresentada na Figura 19. A opção banco de dados tem o significado que tal formulário utiliza banco de dados.

Figura 19 - Cadastros/formulários criados.

Para chegar até a opção **Gerar BD** é utilizado o menu em largura e profundidade disponível na ferramenta CASE *on-line* como está na Figura 20. Quando é selecionada a opção **Gerar BD**, o banco de dados é criado, sendo processado um arquivo para geração do mesmo, onde parte do código fonte encontra-se no Apêndice A. Nos Apêndices B e C encontram-se os códigos fontes para a geração das tabelas e campos do banco de dados.

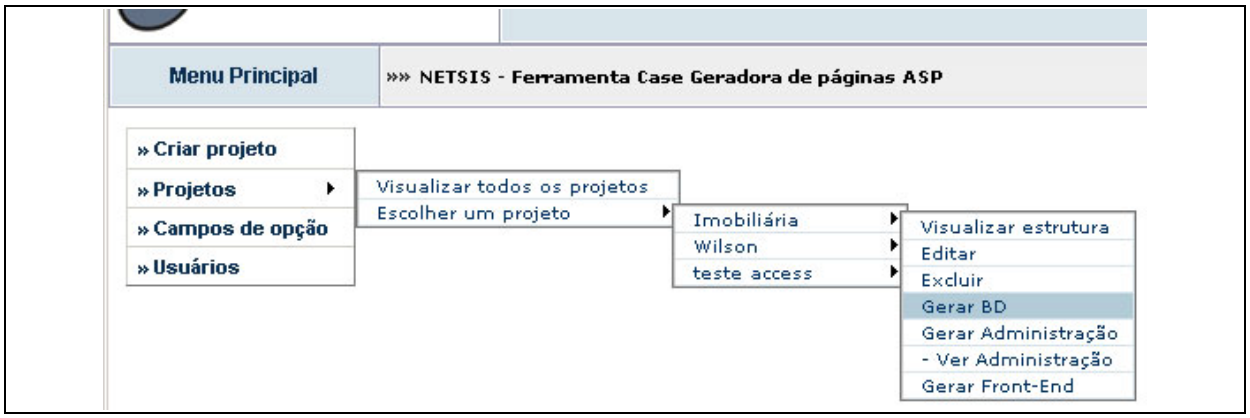


Figura 20 - Menu com sub-níveis.

Além de todos os campos criados, a ferramenta também cria os relacionamentos entre as tabelas conforme é mostrado na Figura 21, no *SQL Server Enterprise Manager* em *Diagrams*.

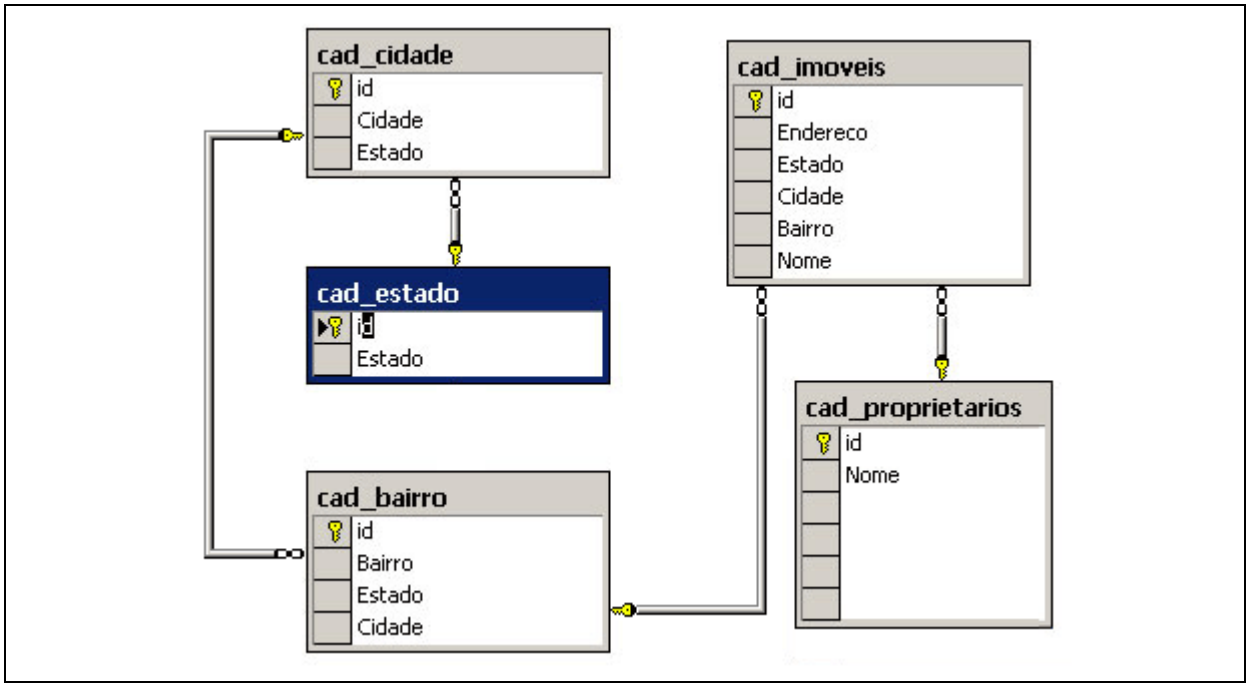


Figura 21 - Diagrama do SQL Server mostra os relacionamentos.

Depois que o banco de dados é criado, gera-se a administração do site. Para isto, deve-se ir ao menu e escolher a opção **Gerar Administração** que direciona para uma tela de configurações da administração, conforme a Figura 22 demonstra.

Figura 22 - Configuração da geração da administração do site.

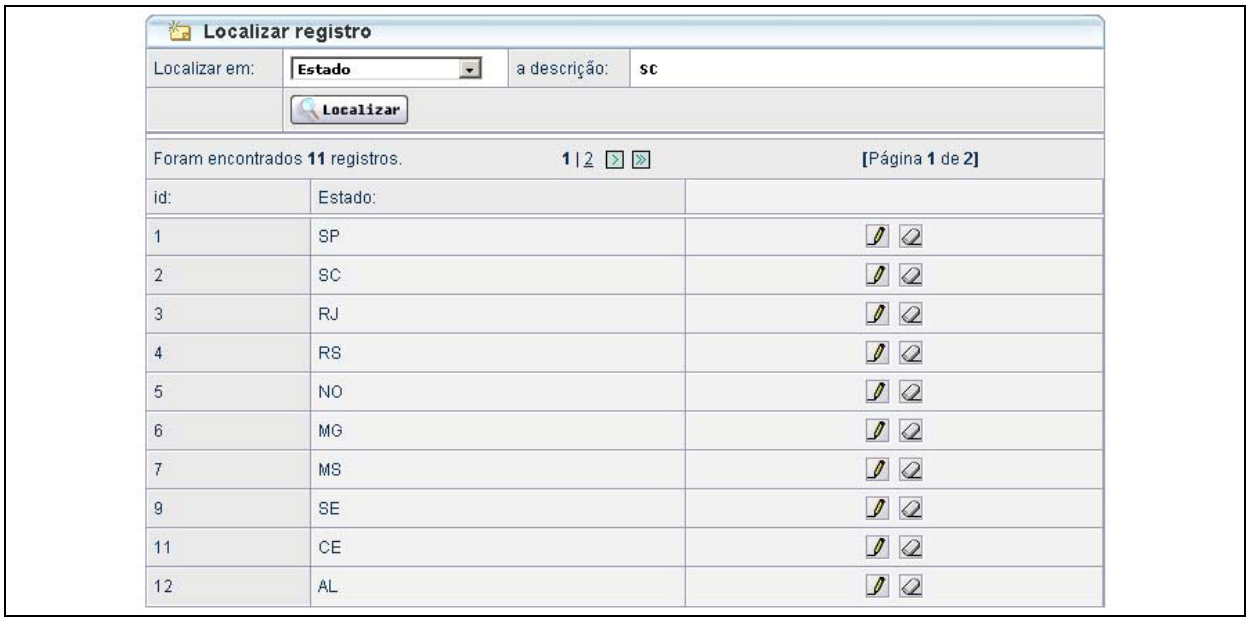
Nas **Configurações da Administração** é definido se é necessário uma tela inicial de autenticação com *login* e *senha* para restringir acessos a usuários não permitidos, mantendo a segurança em todas as páginas internas da administração. Esta segurança bloqueia qualquer usuário que tenta acessar diretamente um link da administração sem a autenticação. Na Figura 23, mostra-se a página de autenticação gerada para entrar na administração do site.

Figura 23 - Sistema gerado - Tela de autenticação.

A tela principal do sistema foi gerada colocando-se os próprios cadastros ou tabelas como menus conforme é verificado na Figura 24.

Figura 24 - Sistema gerado – Tela principal.

Na tela de localizar registros é encontrado os botões para alterar e excluir o registro e também pode ser verificada a paginação dos registros de dez em dez, conforme mostra a Figura 25.



id:	Estado:	
1	SP	
2	SC	
3	RJ	
4	RS	
5	NO	
6	MG	
7	MS	
9	SE	
11	CE	
12	AL	

Figura 25 - Sistema gerado – Localizar registros por campo.

Na Figura 26 está a tela de inserção de dados que foi gerada pela ferramenta onde uma parte do código fonte encontra-se no Apêndice D.

Nas telas de inserção de registros é encontrado os botões de inserir ou localizar registros de outras tabelas, conforme é demonstrado nas flechas destacadas na Figura 26. Esta opção do botão de inserção é importante para o usuário não precisar voltar até a tela de

inserção de estados, facilitando a inserção dos dados quando o cadastro possuir muitos relacionamentos com outras tabelas. A facilidade do botão localizar, também é importante, pois ele é disposto para cadastros com um número muito grande de registros armazenados, assim o usuário poderá localizar e selecionar o registro desejado mais rapidamente.

Figura 26 - Sistema gerado – Tela de inserção de registros.

A tela de edição de registros é muito semelhante a tela anterior da Figura 26, pois a mudança ocorre no código fonte no comando *SQL*, que ao invés de *Insert*, é alterado para *Updade*.

Figura 27 - Sistema gerado – Tela de edição de registros.

3.4 RESULTADOS E DISCUSSÃO

A ferramenta apresentou alguns diferenciais em relação a outras ferramentas encontradas e implementadas em trabalhos anteriores, como por exemplo, a geração completa de aplicativos na *web*, incluindo criação *on-line* do banco de dados, definição, documentação, integridade referencial e interfaces prontas para o usuário final começar a trabalhar com o aplicativo. Um grande diferencial também é o trabalho com hierarquias de campos em um formulário de cadastro baseados na integridade referencial do banco de dados como exemplo na Figura 28.

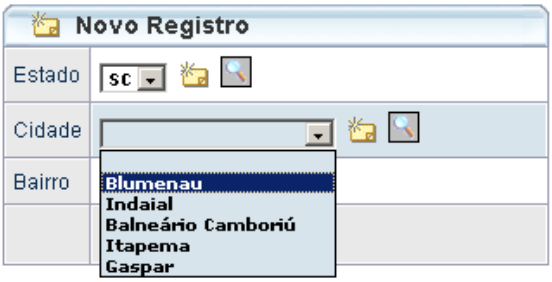


Figura 28 - Hierarquia de campos.

Na Figura 28, mostra que em um formulário de cadastro pode-se ter campos de outras tabelas relacionadas entre si. Os campos: Estado, Cidade e Bairro são tabelas e possuem relacionamento entre elas no banco de dados. No formulário quando é selecionado um Estado, é atualizado o campo Cidade com suas respectivas cidades.

Não são encontrados os diferenciais citados na ferramenta disponível no mercado por (TEGNHER, 2001) ou em trabalhos de conclusões de cursos (DIAS, 2002 e SILVEIRA, 2003).

Descrição – Geração	NETSIS	SILVEIRA	DATAFORM
Criação do banco de dados	**X**	**X**	
Flexibilidade da ferramenta	**X**	**X**	
Não intervenção do programador	**X**	**X**	
Opções de escolha e seleção	**X**	**X**	
Formulário de inclusão	**X**	**X**	**X**
Formulário de alteração	**X**	**X**	**X**
Exclusão de registros	**X**	**X**	
Localizar/Consulta	**X**	**X**	**X**
Localizar por campo	**X**		
Paginação	**X**		
Geração do Menu	**X**		
Banco de Dados Access	**X**	**X**	**X**
Banco de Dados SQL Server	**X**		
Cria os relacionamentos entre as tabelas	**X**	**X**	
Gera a aplicação completa	**X**		
Mostrar campos na consulta	**X**	**X**	
Campos requeridos	**X**	**X**	
Consistência de campos CNPJ, CPF, entre outros.	**X**		
Geração apartir da leitura do banco de dados		**X**	

Quadro 3 - Comparação das ferramentas.

Apesar das várias vantagens citadas, a ferramenta CASE que foi desenvolvida limita-se a geração de código somente na linguagem *ASP* e nos bancos de dados *Access* e *SQL Server*. Também se limita na geração apartir da leitura de um banco de dados previamente

criado e modelado por outra ferramenta, onde a ferramenta proposta por Silveira (2003) está disponível esta opção.

4 CONCLUSÕES

A implementação da ferramenta CASE gera automaticamente sistemas completos de cadastros com o banco de dados relacional possuindo opções de inclusão, alteração, exclusão, localização e paginação dos registros alcançando os objetivos propostos, porém é necessária a utilização da ferramenta para a geração de vários outros sistemas, para garantir a sua completa integridade. Na ferramenta são definidas informações para geração de um banco de dados relacional e a geração completa da aplicação.

A ferramenta alcançou o seu principal objetivo que é agilizar a criação de aplicativos de cadastros para programadores e analistas de sistemas, onde anteriormente o programador demorava meses para programar, agora é possível levar horas sem o conhecimento da linguagem de *script* ASP.

Além dos objetivos propostos de gerar a aplicação completa de cadastro "Administração do Site", foi desenvolvida a parte do "*front-end*" do *site*, onde é a parte que fica disponível as informações aos usuários do *site*. Também foi criada a documentação de toda a aplicação, onde o usuário poderá imprimir e apresentar primeiramente ao seu cliente, para em caso de manutenção da aplicação.

O resultado obtido pela ferramenta não apenas cria páginas isoladas de cadastros, mas cria uma aplicação completa e consistente, com menu ligados aos cadastros e ligados ao banco de dados relacional. Com o resultado deste trabalho será dada continuidade ao desenvolvimento desta ferramenta para melhoramento da mesma em novas versões, ocultando-se cada vez mais a complexidade para a criação de aplicações de manipulação de dados na internet.

4.1 EXTENSÕES

Como sugestões para desenvolvimentos de futuros trabalhos pode-se fazer a geração de páginas em outras linguagens de scripts ou linguagens de programação tais como: *PHP*, *Cold Fusion*, *ASP .NET* e *C#.* Também pode ser expandido o uso de outros bancos de dados tais como: *MySQL*, *Informix* entre outros.

Outra sugestão também é a geração de códigos, estruturas e esquemas pré-definidos em *XML*.

REFERÊNCIAS BIBLIOGRÁFICAS

BEZERRA, Eduardo. **Princípios de análise e projeto de sistemas com UML**. Rio de Janeiro: Campus, 2002.

BUCZEK, GREG. **ASP guia do programador.** São Paulo: Market Books, 2000.

DEITEL, H. M., et al. **Visual Basic .Net: como programar.** São Paulo: Pearson Education do Brasil, 2004.

DIAS, Adriano. **Aplicativo para atualização de banco de dados utilizando Active Server Pages (ASP)**. 2002. 42 f. Trabalho de Conclusão de Curso (Bacharelado em Ciências da Computação) – Centro de Ciências Exatas e Naturais, Universidade Regional de Blumenau, Blumenau.

FURLAN, José Davi. **Modelagem de objetos através da UML – the unified modeling language.** São Paulo: Makron Books, 1998.

ISC – Internet Systems Consortium. **Domain Survey Information**, Redwood City, CA, EUA. Disponível em: <http://www.isc.org/index.pl?/ops/ds/>. Acesso em: 29 abr. 2004.

MARTIN, James; MCCLURE, Carma. **Técnicas estruturadas e case**. São Paulo: Makron Books & McGraw-HILL, 1991.

PRESSMAN, Roger S. **Engenharia de software.** São Paulo: Makron Books, 1995.

_____**Engenharia de software.** Rio de Janeiro: MCGraw-Hill, 2002.

SILVEIRA, Claudionor. **Geração automática de cadastros e consultas para linguagem ASP baseado em banco de dados**. 2003. 77 f. Trabalho de Conclusão de Curso (Bacharelado em Ciências da Computação) – Centro de Ciências Exatas e Naturais, Universidade Regional de Blumenau, Blumenau.

TEGNHER, José Luís. **SUPERASP.COM**, o portal do desenvolvedor ASP, Florianópolis, 2001. Disponível em: <http://www.superasp.com.br/>. Acesso em: 29 mar. 2004.

UNITO – Unito Sistemas . **Domain Survey Information**, Rio de Janeiro, RJ, Brazil. Disponível em: <http://www.unito.com.br/materia/view/131>. Acesso em: 06 mai. 2004.

WILLIE, Christoph; KOLLER, Christian. **Aprenda em 24 horas Active Server Pages ASP.** Rio de Janeiro: Campus, 1999.

APÊNDICE A – Código fonte – Geração do banco de dados

```
...
' Apagando a base de dados atual e a conexão com a mesma
call ApagaPasta("projetos/"&id_projeto&"/database")

' Se não existir a base de dados, nesta linha ocorrerá o erro de código -2147217865
conn.execute("DROP DATABASE "&bd_nome)
'Apagando o usuário da base de dados

conn.execute("sp_droplogin '"&bd_login&"'")
Server.ScriptTimeOut = 9000

'Criando a pasta onde ficará armazenada a conexão (conn.asp) e o arquivo Access caso seja deste tipo a base de dados
call CriaPasta("projetos/"&id_projeto&"/database")

if tipo_bd = "Access" then ' Se a base de dados for Access

' Copia o arquivo do Access padrão que está no servidor para a pasta do projeto /database
call CopiaArquivo("bd.mdb","projetos/padrao/database","projetos/"&id_projeto&"/database")
' Em seguida renomeia o mesmo para bd_IDDOPROJETO

call RenomeiaArquivo("projetos/"&id_projeto&"/database","bd.mdb",bd_nome&".mdb")

'Se conecta a base de dados que acaba de ser criada
set ConnTemp=server.createobject("ADODB.Connection")

ConnTemp.open "Driver={Microsoft Access Driver
(*.mdb)};DBQ="&server.MapPath("projetos/"&id_projeto&"/database/bd_"&id_projeto&".mdb")

ConnTemp.open "Provider=Microsoft.Jet.OLEDB.4.0;"&"Data
Source="&server.MapPath("projetos/"&id_projeto&"/database/"&bd_nome&".mdb")

'ConnTemp.open "PROVIDER=Microsoft.Jet.OLEDB.4.0;Data
Source="&server.MapPath("projetos/"&id_projeto&"/database/bd_"&id_projeto&".mdb")

' Por último cria o arquivo de conexão conn.asp na pasta "projetos/IDDOPROJETO/database"
call CriaArquivo("projetos/"&id_projeto&"/database","conn.asp",asp("set
conn=server.createobject('ADODB.Connection')<PULA>conn.open 'Driver={Microsoft Access Driver
(*.mdb)};DBQ='&server.MapPath('../database/"&bd_nome&".mdb')"))

' Caso a base seja SQL Server
else if tipo_bd = "SQL Server" then

Set ConnTemp = Server.CreateObject("ADODB.Connection")
' Cria no SQL Server a base de dados de nome "net_projeto_IDDOPROJETO"
if bd_existe = "N" then
Conn.execute("CREATE DATABASE "&bd_nome)
ConnTemp.Open "database="&bd_nome&";DSN=netsistema;UID=netsistema;PWD=g8a6k7c9;"

' Em seguida cria o usuário para a base de dados
ConnTemp.execute("sp_addlogin '"&bd_login&"', '"&bd_senha&"', '"&bd_nome&"'")
ConnTemp.execute("sp_changedbowner '"&bd_login&"'")

' E por último cria o arquivo de conexão conn.asp na pasta (projetos/IDDOPROJETO/database)

call CriaArquivo("projetos/"&id_projeto&"/database","conn.asp",asp("set
conn=server.createobject('ADODB.Connection')<PULA>Conn.Open

'database="&bd_nome&";DSN=netsistema;UID="&bd_login&";PWD="&bd_senha&"'))
else

'Se conecta a base de dados que acaba de ser criada
ConnTemp.Open "driver={SQL Server};server="&bd_ip&";UID="&bd_login&";PWD="&bd_senha&";Database="&bd_nome&""

ConnTemp.execute("CREATE DATABASE "&bd_nome)
' E por último cria o arquivo de conexão conn.asp na pasta (projetos/IDDOPROJETO/database)
call CriaArquivo("projetos/"&id_projeto&"/database","conn.asp",asp("set
conn=server.createobject('ADODB.Connection')<PULA>Conn.Open 'driver={SQL
Server};database="&bd_nome&";UID="&bd_login&";PWD="&bd_senha))
end if
end if
...
```

APÊNDICE B – Código fonte – Geração das tabelas e campos do SQL Server

```
...
' Começando a criação das tabelas

' SE A BASE FOR SQL SERVER
'================================
if tipo_bd = "SQL Server" then
'================================

set rsTabelas = conn.execute("select t.* from net_tabelas t, net_form f where t.for_id = f.for_id and f.for_tipo <> 0 and f.pro_id =
"&id_projeto)
do while not rsTabelas.eof
        sql = "CREATE TABLE "&Trim(rsTabelas("tab_nome"))&" ("
        set rsCampos = conn.execute("select * from net_campos where tab_id="&rsTabelas("tab_id")&"")
        do while not rsCampos.eof
                nome          = Trim(rsCampos("cam_nom"))
                descricao     = Trim(rsCampos("cam_desc"))
                if rsCampos("cam_req") <> 0 then requerido = "NOT NULL" else requerido = "" end if
                tipo          = Trim(rsCampos("cam_tipo"))
                if rsCampos("cam_tam") = "0" then tamanho = "" else tamanho = rsCampos("cam_tam") end if
                if rsCampos("cam_caracte") = "0" then caracteres = "50" else caracteres = rsCampos("cam_caracte") end if
                indice        = rsCampos("cam_indice")
                relacionamento = rsCampos("cam_idrelac")
                if (indice <> 0) and (tipo <> "Inteiro") and (tipo <> "Auto") and (tipo <> "text") and (tipo <> "textarea") then
                        chave = chave & nome & " ,"
                        chave2 = chave2 & nome & " ,"
                end if
                if tipo = "Auto" then
                        sql = sql & ""&nome&" int IDENTITY(1,1) NOT NULL ,"
                        chave_primaria = nome
                 elseif tipo = "Inteiro" then
                        sql = sql & ""&descricao&" int NOT NULL ,"
                        chave = chave & nome & " ,"
                        chave2 = chave2 & descricao & " ,"
                 elseif tipo = "text" then
                        sql = sql & ""&nome&" nvarchar("&caracteres&") COLLATE Latin1_General_CI_AS
"&requerido&","
                 elseif tipo = "textarea" then
                        sql = sql & ""&nome&" nText ,"
                 else
                        sql = sql & ""&nome&" nvarchar(100) COLLATE Latin1_General_CI_AS "&requerido&","
                end if
        rsCampos.movenext
        loop
        sql = sql & "CONSTRAINT PK_"&Trim(rsTabelas("tab_nome"))&" PRIMARY KEY ("&chave_primaria&")ON
[PRIMARY]) ON [PRIMARY]"
        ConnTemp.execute(sql)

        if chave <> "" then
                'chave = left(chave,len(chave)-2)
                for i = 1 to Len(chave)
                        if InStr(chave," ,") <> 0 then
                                ConnTemp.execute("create index "&left(chave,InStr(chave," ,")-1)&" on
"&Trim(rsTabelas("tab_nome"))&" ("&left(chave2,InStr(chave2," ,")-1)&")")
                                chave = Mid(chave,InStr(chave," ,")+2,Len(chave))
                                chave2 = Mid(chave2,InStr(chave2," ,")+2,Len(chave2))
                        end if
                next
        end if
rsTabelas.movenext
loop


'GERANDO OS RELACIONAMENTOS
rsTabelas.moveFirst
do while not rsTabelas.eof
        set rsCampos = conn.execute("select * from net_campos where tab_id = "&rsTabelas("tab_id")&" and cam_tipo =
'Inteiro'")
        do while not rsCampos.eof
```

```
                        tab =
Mid(Trim(rsCampos("cam_nom")),InStr(Trim(rsCampos("cam_nom")),"_")+1,len(Trim(rsCampos("cam_nom"))))
                        set rsTabela2 = conn.execute("select * from net_tabelas where tab_id="&left(tab,InStr(tab,"_")-1))
                        ConnTemp.execute("alter table "&Trim(rsTabelas("tab_nome"))&" add constraint
FK_"&Trim(rsTabelas("tab_nome"))&"_REFERENCE_"&Trim(rsTabela2("tab_nome"))&" foreign key
("&rsCampos("cam_desc")&") references "&Trim(rsTabela2("tab_nome"))&" (id) ON DELETE CASCADE")
            rsCampos.movenext
            loop
rsTabelas.movenext
loop
...
```

APÊNDICE C – Código fonte – Geração das tabelas e campos do ACCESS

```
...
' PORÉM SE A BASE DE DADOS FOR ACCESS
'================================
elseif tipo_bd = "Access" then
'================================


set rsTabelas = conn.execute("select t.* from net_tabelas t, net_form f where t.for_id = f.for_id and f.for_tipo <> 0 and f.pro_id =
"&id_projeto)
do while not rsTabelas.eof
        sql = "CREATE TABLE "&Trim(rsTabelas("tab_nome"))&" ("
        set rsCampos = conn.execute("select * from net_campos where tab_id="&rsTabelas("tab_id")&"")
        do while not rsCampos.eof
                nome          = Trim(rsCampos("cam_nom"))
                descricao     = Trim(rsCampos("cam_desc"))
                if rsCampos("cam_req") <> 0 then requerido = " NOT NULL " else requerido = "" end if
                tipo          = Trim(rsCampos("cam_tipo"))
                if rsCampos("cam_tam") = "0" then tamanho = "" else tamanho = rsCampos("cam_tam") end if
                if rsCampos("cam_caracte") = "0" then caracteres = "50" else caracteres = rsCampos("cam_caracte") end if
                indice        = rsCampos("cam_indice")
                relacionamento = rsCampos("cam_idrelac")
                if (indice <> 0) and (tipo <> "Inteiro") and (tipo <> "Auto") then
                        chave = chave & nome & " ,"
                end if
                if tipo = "Auto" then
'                       sql = sql & ""&nome&" counter ,"
                        sql = sql & ""&nome&" number ,"
                        chave_primaria = nome
                 elseif tipo = "Inteiro" then
                        sql = sql & ""&descricao&" number ,"
                        chave = chave & nome & " ,"
                        chave2 = chave2 & descricao & " ,"
                 elseif tipo = "text" then
                        sql = sql & ""&nome&" char("&caracteres&") ,"
                 elseif tipo = "textarea" then
                        sql = sql & ""&nome&" memo ,"
                 else
                        sql = sql & ""&nome&" char(100),"
                end if
        rsCampos.movenext
        loop
        sql = sql & "CONSTRAINT PK_"&Trim(rsTabelas("tab_nome"))&" PRIMARY KEY ("&chave_primaria&"))"
        ConnTemp.execute(sql)
        if chave <> "" then
                chave = left(chave,len(chave)-2)
                for i = 1 to Len(chave)
                        if InStr(chave,",") <> 0 then
                                ConnTemp.execute("create index "&left(chave,InStr(chave," ,")-1)&" on
"&Trim(rsTabelas("tab_nome"))&" ("&left(chave2,InStr(chave2," ,")-1)&")")
                                chave = Mid(chave,InStr(chave," ,")+2,Len(chave))
                                chave2 = Mid(chave2,InStr(chave2," ,")+2,Len(chave2))
                        end if
                next
        end if
rsTabelas.movenext
loop

err.number = 0
'GERANDO OS RELACIONAMENTOS
rsTabelas.moveFirst
do while not rsTabelas.eof
        set rsCampos = conn.execute("select * from net_campos where tab_id = "&rsTabelas("tab_id")&" and cam_tipo =
'Inteiro'")
        do while not rsCampos.eof
                tab =
Mid(Trim(rsCampos("cam_nom")),InStr(Trim(rsCampos("cam_nom")),"_")+1,len(Trim(rsCampos("cam_nom"))))
                set rsTabela2 = conn.execute("select * from net_tabelas where tab_id="&left(tab,InStr(tab,"_")-1))
'               set ConnTemp2=server.createobject("ADODB.Connection")
'               ConnTemp2.open "Provider=Microsoft.Jet.OLEDB.4.0;"&"Data
Source="&server.MapPath("projetos/"&id_projeto&"/database/bd_"&id_projeto&".mdb")
if err.number <> 0 then
```

```
            response.write "<BR>ERRO: " & err.description & " ||| "& "create index "&left(chave,InStr(chave," ,")-1)&" on
"&Trim(rsTabelas("tab_nome"))&" ("&left(chave2,InStr(chave2," ,")-1)&")" &"<br>"
            err.number = 0
end if
                        rel = "alter table "&Trim(rsTabelas("tab_nome"))&" add constraint
FK_"&Trim(rsTabelas("tab_nome"))&"_REFERENCE_"&Trim(rsTabela2("tab_nome"))&" foreign key
("&rsCampos("cam_desc")&") references "&Trim(rsTabela2("tab_nome"))&" (id) ON DELETE CASCADE"
                        ConnTemp.execute(rel)
if err.number <> 0 then
            response.write "<BR>ERRO: " & err.description & " ||| "& "create index "&left(chave,InStr(chave," ,")-1)&" on
"&Trim(rsTabelas("tab_nome"))&" ("&left(chave2,InStr(chave2," ,")-1)&")" &"<br>"
            err.number = 0
end if
            rsCampos.movenext
            loop
rsTabelas.movenext
loop

' FIM
'=====================
end if
'=====================

call CriaArquivo("projetos/"&id_projeto,"config.asp",asp("database = '"&tipo_bd&"'"))

Response.Redirect("home.asp?bd=gerado&teste="&server.mappath("projetos/"&id_projeto&"/database/"&bd_nome&".mdb"))
...
```

APÊNDICE D – Código fonte – Geração das telas de inserção dos dados

```
...
                    ' CÓDIGO DA TELA DE INSERÇÃO
                    codigo = ""
                    if request.form("autenticacao")  <> "" then
                              codigo = codigo & "<inclui>seguranca.asp</inclui><PULA>"
                    end if
                    codigo = codigo & "<inclui>../database/conn.asp</inclui><PULA>"
                    codigo = codigo & "<inclui>../config.asp</inclui><PULA>"
                    codigo = codigo & "<inclui>includes/funcoes.asp</inclui><PULA>"
                    codigo = codigo & "<script language='JavaScript' src='java/scripts.js'></script><PULA>"
                    codigo = codigo & "<script language='JavaScript'><PULA>"
                    codigo = codigo & "window.name=j'"&tabela&"'j';<PULA>"
                    codigo = codigo & "<asp>if request('tipo') <> '' then</asp><PULA>"
                    codigo = codigo & "if (window.name==opener.window.name) {window.close();}<PULA>"
                    codigo = codigo & "<asp>end if</asp><PULA>"
                    codigo = codigo & "</script><PULA>"
                    action = ""
                    rsCampos.moveFirst
                    do while not rsCampos.eof
                              set rsTemp = conn.execute("select * from net_campos where cam_idrelac =
"&rsCampos("cam_id")&" and cam_inserir <> 0")
                              if not rsTemp.eof then
                                        codigo = codigo & "<script language='javascript'><PULA>"
                                        codigo = codigo & "function salvar(texto, valor){<PULA>"
                                        codigo = codigo & "campo =
opener.document.getElementById(j'<asp>=request.QueryString('cam')</asp>j');<PULA>"
                                        codigo = codigo & "option = opener.document.createElement(j'OPTIONj');<PULA>"
                                        codigo = codigo & "tamanho = campo.length;<PULA>"
                                        codigo = codigo & "option.text = texto;<PULA>"
                                        codigo = codigo & "option.value = valor;<PULA>"
                                        codigo = codigo & "campo.options.add(option);<PULA>"
                                        codigo = codigo & "campo.selectedIndex = tamanho;<PULA>"
                                        ok = 0
                                        do while not rsTemp.eof
                                                  set rsTemp2 = conn.execute("select cam_nom, cam_desc from net_campos
where cam_iddepende = "&rsTemp("cam_id"))
                                                  if (not rsTemp2.eof) and (ok <> 1) then
                                                            set rsTemp3 = conn.execute("select tab_nome from
net_tabelas where tab_id = "&Mid(Trim(rsTemp2("cam_nom")),4,2))
                                                            codigo = codigo &
"opener.MM_openBrWindow(j'"&Trim(rsTemp3("tab_nome"))&"_inserir.asp?tipo=inserir&cam="&Trim(rsTemp2("cam_desc"))&"j
',j'pop"&Trim(rsTemp3("tab_nome"))&"j');<PULA>"
                                                            ok = 1
                                                  end if
                                        rsTemp.movenext
                                        loop
                                        codigo = codigo & " window.close();<PULA>"
                                        codigo = codigo & "}<PULA>"
                                        codigo = codigo & "</script><PULA>"
                                        action = "response.write
'<script>salvar(j'"&request.form("'&Trim(rsCampos("cam_nom"))&"')&'j',j'"&ult('ultimo')&'j')</script>'<PULA>"
                              end if
                    rsCampos.movenext
                    loop
                    codigo = codigo & "<link href='css/estilos.css' rel='stylesheet' type='text/css'><PULA>"
                    codigo = codigo & "<asp><PULA>"
                    codigo = codigo & "if request.querystring('salvar') <> '' then<PULA>"
                    rsCampos.moveFirst
                    var  = ""
                    var2 = ""
                    var3 = ""
                    do while not rsCampos.eof
                    if Trim(rsCampos("cam_nom")) <> "id" then
                              if rsCampos("cam_idrelac") <> 0 then
                                        var  = var & Trim(rsCampos("cam_desc"))&","
                                        var2 = var2 & "j'"&request.form("'&Trim(rsCampos("cam_desc"))&"')&'j',"
                                        var3  = var3 & Trim(rsCampos("cam_desc"))&": " &
"'&request.form("'&Trim(rsCampos("cam_desc"))&"')&'<BR>"
                               else
                                        var = var & Trim(rsCampos("cam_nom"))&","
                                        var2 = var2 & "j'"&request.form("'&Trim(rsCampos("cam_nom"))&"')&'j',"
```

```
                                        var3  = var3 & Trim(rsCampos("cam_nom"))&": " &
"'&request.form('"&Trim(rsCampos("cam_nom"))&"')&'<BR>"
                                end if
                        end if
                        rsCampos.movenext
                        loop
                        if tipo_bd = "Access" then
                                codigo = codigo & "set verifica = conn.execute('select COUNT(id) as qtde from
"&tabela&"')<PULA>"
                                codigo = codigo & "if verifica('qtde') = 0 then<PULA>"
                                codigo = codigo & " id = 1<PULA>"
                                codigo = codigo & " else<PULA>"
                                codigo = codigo & " set ultimo = conn.execute('select MAX(id) as ultimo from
"&tabela&"')<PULA>"
                                codigo = codigo & " id = ultimo('ultimo') + 1<PULA>"
                                codigo = codigo & "end if<PULA>"
                                codigo = codigo & " conn.execute('insert into "&tabela&"(id,"&left(var,len(var)-1)&")
values('&id&',"&left(var2,len(var2)-1)&")')<PULA>"
                        else
                                codigo = codigo & " conn.execute('insert into "&tabela&"("&left(var,len(var)-1)&")
values("&left(var2,len(var2)-1)&")')<PULA>"
                        end if
                        codigo = codigo & " set ult = conn.execute('select MAX(id) as ultimo from "&tabela&"')<PULA>"
                        codigo = codigo & " msg = 'Registro cadastrado com sucesso.'<PULA>"
                        if Trim(rsCads("for_email")) <> "" then
                                codigo = codigo & " call enviaemail(1,'"& Trim(rsCads("for_subject")) &"','"&
Trim(rsCads("for_email")) &"','"&Trim(rsCads("for_email"))&"','Novo registro cadastrado em:
"&tabela&"<BR><BR>"&var3&"')<PULA>"
                                codigo = codigo & " msg = msg & ' E-mail enviado.'<PULA>"
                        end if
                        codigo = codigo & " if request.QueryString('tipo') <> '' then<PULA>"
                        codigo = codigo & "                "&action
                        codigo = codigo & " end if<PULA>"
                        codigo = codigo & "end if<PULA>"
                        codigo = codigo & "</asp><PULA>"
                        codigo = codigo & "<table width='94%' align='center' style='border-collapse: collapse' cellpadding='2'
cellspacing='1' border='1' bordercolor='#9C9CB2' ID='Table9'><PULA>"
                        codigo = codigo & "  <tr bgcolor='#E9E9E9'> <PULA>"
                        codigo = codigo & " <td> <table width='100%' align='center' cellpadding='1' cellspacing='0' border='0'
bordercolor='#9C9CB2'><PULA>"
                        codigo = codigo & "        <tr> <PULA>"
                        codigo = codigo & "          <td align='right'><PULA>"
                        codigo = codigo & "                                  <p><a href=<asp> if request.QueryString('tipo') = 'inserir'
then</asp>#<asp>else</asp>'"&tabela&".asp?campo=<asp>=request.QueryString('campo')</asp>&descricao=<asp>=request.
QueryString('descricao')</asp><asp>end if</asp>' <asp> if request.QueryString('tipo') = 'inserir'
then</asp>onClick='window.close();'<asp>end if</asp> class='nounderlinelink'><img src='img/voltar.gif'
border='0'></a></p><PULA>"
                        codigo = codigo & "                </td><PULA>"
                        codigo = codigo & "       </tr><PULA>"
                        codigo = codigo & "     </table><PULA>"
                        codigo = codigo & "  </td><PULA>"
                        codigo = codigo & "  </tr><PULA>"
                        codigo = codigo & "</table><PULA>"
                        codigo = codigo & "<div align='center' class='mensagem'><br><asp>=msg</asp></div><br><PULA>"
                        codigo = codigo & "<table border='0' align='center' cellpadding='0' cellspacing='0'><PULA>"
                        codigo = codigo & "  <tr> <PULA>"
                        codigo = codigo & "    <td><table background='img/header.tile.gif' align='center' class='tablestyle'
bordercolor='#9C9CB2' width='100%' cellpadding='0' cellspacing='0' border='0' ID='Table12'><PULA>"
                        codigo = codigo & "        <tr> <PULA>"
                        codigo = codigo & "          <td width='15' height='24'><img src='img/header.left.gif' width='15'
height='24'></td><PULA>"
                        codigo = codigo & "          <td><table><PULA>"
                        codigo = codigo & "             <tr> <PULA>"
                        codigo = codigo & "               <td width='18'><img src='img/novo.gif'></td><PULA>"
                        codigo = codigo & "               <td width='130'>  <span class='smallBold'>Novo
Registro</span></td><PULA>"
                        codigo = codigo & "             </tr><PULA>"
                        codigo = codigo & "           </table></td><PULA>"
                        codigo = codigo & "          <td align='right'><img src='img/header.right.gif' width='15'
height='24'></td><PULA>"
                        codigo = codigo & "       </tr><PULA>"
                        codigo = codigo & "     </table></td><PULA>"
                        codigo = codigo & "  </tr><PULA>"
                        codigo = codigo & "  <tr> <PULA>"
```

```
                codigo = codigo & "    <td><table border='1' align='center' cellpadding='4' cellspacing='1'
bordercolor='#9C9CB2'><PULA>"
                codigo = codigo & "       <form name='form1' method='post'
action='"&tabela&"_inserir.asp?salvar=sim&tipo=<asp>=request.QueryString('tipo')</asp>&campo=<asp>=request.QueryString(
'campo')</asp>&cam=<asp>=request.QueryString('cam')</asp>&descricao=<asp>=request.QueryString('descricao')</asp>'
><PULA>"
                rsCampos.moveFirst
                dependentes = 0
                do while not rsCampos.eof
                documentacao = Trim(rsCampos("cam_doc"))
                relacionamento = rsCampos("cam_idrelac")
                dependencia = rsCampos("cam_iddepende")
                insere = rsCampos("cam_inserir")
                local = rsCampos("cam_localiza")
                if Trim(rsCampos("cam_nom")) <> "id" then
                        if relacionamento <> 0 then
                                set rsTemp1 = conn.execute("select * from net_campos where cam_iddepende =
"&rsCampos("cam_id"))
                                if not rsTemp1.eof then
                                        dependente = "update_"&Trim(rsTemp1("cam_desc"))&"();"
                                  else
                                        dependente = ""
                                end if
                                inserir = ""
                                if insere <> 0 then
                                        set rsTab = conn.execute("select t.tab_nome from net_campos c, net_tabelas
t where c.cam_id = "&relacionamento&" and t.tab_id=c.tab_id")
                                        inserir = " <a href='javascript:
MM_openBrWindow(j'"&Trim(rsTab("tab_nome"))&"_inserir.asp?tipo=inserir&cam="&Trim(rsCampos("cam_desc"))&"j')'><img
src='img/novo.gif' border='0'></a>"
                                        set rsTab = nothing
                                        rsTab.close
                                end if
                                localiza = ""
                                if local <> 0 then
                                        set rsTab = conn.execute("select t.tab_nome from net_campos c, net_tabelas
t where c.cam_id = "&relacionamento&" and t.tab_id=c.tab_id")
                                        if dependencia <> 0 then
                                                set rsTempor = conn.execute("select cam_desc from net_campos
where cam_id = "&dependencia)
                                                localiza = " <a href='javascript:
MM_openBrWindow(j'"&Trim(rsTab("tab_nome"))&".asp?tipo=localiza&executar="&dependente&"&cam="&Trim(rsCampos("ca
m_desc"))&"&depende=j'+document.form1."&Trim(rsTempor("cam_desc"))&".value+j'&cam_depende="&Trim(rsTempor("cam_d
esc"))&"j')'><img src='img/localizar.gif' border='0'></a>"
                                          else
                                                localiza = " <a href='javascript:
MM_openBrWindow(j'"&Trim(rsTab("tab_nome"))&".asp?tipo=localiza&executar="&dependente&"&cam="&Trim(rsCampos("ca
m_desc"))&"j')'><img src='img/localizar.gif' border='0'></a>"
                                                end if
                                end if
                                var =
Mid(Trim(rsCampos("cam_nom")),InStr(rsCampos("cam_nom"),"_")+1,len(Trim(rsCampos("cam_nom"))))
                                var = left(var,InStr(var,"_")-1)
                                set rsTemp = conn.execute("select tab_nome from net_tabelas where tab_id = "&var)
                                input = "<asp>set rs"&Trim(rsCampos("cam_desc"))&" = conn.execute('select * from
"&Trim(rsTemp("tab_nome"))&"')</asp><PULA>"
                                input = input & "<select name='"&Trim(rsCampos("cam_desc"))&"'
id='"&Trim(rsCampos("cam_desc"))&"' class='campo' onChange='"&dependente&"' <asp>if request('tipo') = "
then</asp>onFocus='foco(this,j'"&documentacao&"j',j'#D8E4EBj');' onBlur='foco(this,j'j',j'#FFFFFFj');' <asp>end
if</asp>><PULA>"
                                input = input & "<option></option><PULA>"
                                input = input & "<asp>do while not
rs"&Trim(rsCampos("cam_desc"))&".eof</asp><PULA>"
                                input = input & "<option
value='<asp>=rs"&Trim(rsCampos("cam_desc"))&"('id')</asp>'><asp>=rs"&Trim(rsCampos("cam_desc"))&"('"&Trim(rsCampos(
"cam_desc"))&"')</asp></option><PULA>"
                                input = input & "<asp><PULA>"
                                input = input & "rs"&Trim(rsCampos("cam_desc"))&".movenext<PULA>"
                                input = input & "loop<PULA>"
                                input = input & "</asp><PULA>"
                                input = input & "</select><PULA>"
                                input = input & "<asp> if request.QueryString('tipo') <> 'inserir' then</asp>" & inserir &
" " & localiza & "<asp>end if</asp>"
                                        if dependencia <> 0 then
```

```
                                                    set rsTemp2 = conn.execute("select c.cam_nom, c.cam_desc,
t.tab_nome from net_campos c, net_tabelas t where c.cam_id = "&rsCampos("cam_iddepende")&" and t.tab_id=c.tab_id")
                                                    if rsTemp2("cam_idrelac") <> 0 then
                                                              cam = Trim(rsTemp2("cam_desc"))
                                                     else
                                                              cam = Trim(rsTemp2("cam_nom"))
                                                    end if
                                                    tab   = Trim(rsTemp2("tab_nome"))
                                                    campo = Trim(rsCampos("cam_desc"))

                                                    session("cam"&dependentes) = cam
                                                    session("tab"&dependentes) = tab
                                                    session("campo"&dependentes) = campo
                                                    session("tabela"&dependentes) = Trim(rsTemp("tab_nome"))

                                                    if dependentes = 0 then

                                                    input = input & "          <asp> j'*** UPDATE_"&campo&" ***
</asp><PULA>"
                                                    input = input & "          <SCRIPT language=javascript><PULA>"
                                                    input = input & "                              arr_"&campo&" = new
Array()<PULA>"
                                                    input = input & "                              arr_"&campo&"[0] = new
Array( j'j', j'Escolha uma opção no campo acimaj' )<PULA>"
                                                    input = input & "                              ids_arr_"&campo&" = new
Array() <PULA>"
                                                    input = input & "                              ids_arr_"&campo&"[0] =
new Array( j'j', j'xj' )<PULA>"
                                                    input = input & "                              <asp> <PULA>"
                                                    input = input & "                              i = 1<PULA>"
                                                    input = input & "                              Text  = ''<PULA>"
                                                    input = input & "                              Value = ''<PULA>"
                                                    input = input & "                              if not rs"&cam&".BOF then
rs"&cam&".MoveFirst<PULA>"
                                                    input = input & "                              while not rs"&cam&".EOF
<PULA>"
                                                    input = input & "                                        br = chr(10) +
chr(13)<PULA>"
                                                    input = input & "                                        set rs"&campo&"
= server.CreateObject('ADODB.RECORDSET')<PULA>"
                                                    input = input & "                                        sql = ' SELECT *
FROM "&Trim(rsTemp("tab_nome"))&"  '<PULA>"
                                                    input = input & "                                        sql = sql & '
where "&cam&" = ' & rs"&cam&"('id') <PULA>"
                                                    input = input & "                                        sql = sql & ' order
by "&campo&" '<PULA>"
                                                    input = input & "
              rs"&campo&".Open sql,conn<PULA>"
                                                    input = input & "                                        Text  = Text &
'arr_"&campo&"[' & i & '] = new Array( j'j' '<PULA>"
                                                    input = input & "                                        Value = Value & '
ids_arr_"&campo&"[' & i & '] = new Array( j'j'"<PULA>"
                                                    input = input & "                                        cont = 1<PULA>"
                                                    input = input & "                                        while not
rs"&campo&".EOF <PULA>"
                                                    input = input & "                                                  Text  =
Text & ', j'' & rs"&campo&"(""&campo&"") & 'j' '<PULA>"
                                                    input = input & "                                                  Value =
Value & ',j''& rs"&campo&"('id') & 'j' '<PULA>"
                                                    input = input & "
              rs"&campo&".MoveNext<PULA>"
                                                    input = input & "                                                  Cont =
Cont + 1<PULA>"
                                                    input = input & "                                        wend<PULA>"
                                                    input = input & "                                        Text = Text & ') '
& br<PULA>"
                                                    input = input & "                                        Value = Value & ')
' & br<PULA>"
                                                    input = input & "                                        i = i + 1<PULA>"
                                                    input = input & "
              rs"&cam&".MoveNext<PULA>"
                                                    input = input & "                              wend<PULA>"
                                                    input = input & "                              response.write
Text<PULA>"
```

```
                                                input = input & "                        response.write
Value<PULA>"
                                                input = input & "                </asp><PULA>"
                                                input = input & "<PULA>"
                                                input = input & "                        "&campo&" =
document.getElementById( j'"&campo&"j' )<PULA>"
                                                input = input & "                        "&cam&" =
document.getElementById( j'"&cam&"j' )<PULA>"
                                                input = input & "                        <PULA>"
                                                input = input & "                        function
esvazia_cadastre_"&campo&"() {<PULA>"
                                                input = input & "                         while( "&campo&".length )
{<PULA>"
                                                input = input & "
        "&campo&".remove( "&campo&".length - 1 )<PULA>"
                                                input = input & "                        }<PULA>"
                                                input = input & "                                option =
document.createElement(j'OPTIONj')<PULA>"
                                                input = input & "                                option.text = '
';<PULA>"
                                                input = input & "                                option.value = '
';<PULA>"
                                                input = input & "                                (
"&campo&".options.add ) ? "&campo&".options.add( option ) : "&campo&".add( option, "&campo&".options["&campo&".length]
);<PULA>"
                                                input = input & "                        <PULA>"
                                                input = input & "                                option =
document.createElement('OPTION')<PULA>"
                                                input = input & "                                option.text =
j'Cadastre um registro neste campoj';<PULA>"
                                                input = input & "                                option.value = j'
j';<PULA>"
                                                input = input & "                                (
"&campo&".options.add ) ? "&campo&".options.add( option ) : "&campo&".add( option, "&campo&".options["&campo&".length]
);<PULA>"
                                                input = input & "
        "&campo&".selectedIndex = 0;<PULA>"
                                                input = input & "                        }<PULA>"
                                                input = input & "                        function
esvazia_"&campo&"() {<PULA>"
                                                input = input & "                         while( "&campo&".length )
{<PULA>"
                                                input = input & "
        "&campo&".remove( "&campo&".length - 1 )<PULA>"
                                                input = input & "                        }<PULA>"
                                                input = input & "                        <PULA>"
                                                input = input & "                                option =
document.createElement(j'OPTIONj')<PULA>"
                                                input = input & "                                option.text = j'
j';<PULA>"
                                                input = input & "                                option.value = j'
j';<PULA>"
                                                input = input & "                                (
"&campo&".options.add ) ? "&campo&".options.add( option ) : "&campo&".add( option, "&campo&".options["&campo&".length]
);<PULA>"
                                                input = input & "
        "&campo&".selectedIndex = 0;<PULA>"
                                                input = input & "                        }<PULA>"
                                                input = input & "                        function
update_"&campo&"(opc) {<PULA>"
                                                input = input & "
        if(opc!=j'inicialj'){"&dependente&"}<PULA>"
                                                input = input & "                        while( "&campo&".length )
{<PULA>"
                                                input = input & "
        "&campo&".remove( "&campo&".length - 1 )<PULA>"
                                                input = input & "                        }<PULA>"
                                                input = input & "                        while(
arr_"&campo&"["&cam&".selectedIndex].length > "&campo&".length ) {<PULA>"
                                                input = input & "                                option =
document.createElement(j'OPTIONj')<PULA>"
                                                input = input & "                                option.text =
arr_"&campo&"["&cam&".selectedIndex]["&campo&".length]<PULA>"
```

```
                                                input = input & "                                option.value =
ids_arr_"&campo&"["&cam&".selectedIndex]["&campo&".length];<PULA>"
                                                input = input & "                                (
"&campo&".options.add ) ? "&campo&".options.add( option ) : "&campo&".add( option, "&campo&".options["&campo&".length]
);<PULA>"
                                                input = input & "                                <asp> if
index"&campo&"&"<>" then </asp><PULA>"
                                                input = input & "
        "&campo&".selectedIndex = <asp>=index"&campo&"</asp>;<PULA>"
                                                input = input & "                                <asp> end
if</asp><PULA>"
                                                input = input & "                            }<PULA>"
                                                input = input & "                           }<PULA>"
                                                input = input & "                    update_"&campo&"(j'inicialj')<PULA>"
                                                input = input & "       </SCRIPT><PULA>"

                                                elseif dependentes = 1 then

                                                input = input & "          <SCRIPT language=javascript><PULA>"
                                                input = input & "                           <asp> <PULA>"
                                                input = input & "                           i = 1<PULA>"
                                                input = input & "                           br = chr(10) +
chr(13)<PULA>"
                                                input = input & "                           Text = br & 'arr_"&campo&"
= new Array()' & br & 'arr_"&campo&"[0] = new Array( )' & br & 'arr_"&campo&"[0][0] = new Array( j'j', j'Escolha uma opção no
campo acimaj' )' & br<PULA>"
                                                input = input & "                           Value = br & br &
'ids_arr_"&campo&" = new Array()' & br & 'ids_arr_"&campo&"[0] = new Array()' & br & 'ids_arr_"&campo&"[0][0] = new Array(
j'j', j'j' )' & br<PULA>"
                                                input = input & "                           if not
rs"&session("cam0")&".BOF then rs"&session("cam0")&".MoveFirst<PULA>"
                                                input = input & "                           while not
rs"&session("cam0")&".EOF <PULA>"
                                                input = input & "                                   set
rs"&session("campo0")&" = server.CreateObject('ADODB.RECORDSET')<PULA>"
                                                input = input & "                                   sql = 'SELECT *
FROM "&session("tabela0")&"  '<PULA>"
                                                input = input & "                                   sql = sql & '
where "&session("cam0")&" = ' & rs"&session("cam0")&"('id') <PULA>"
                                                input = input & "                                   sql = sql & ' order
by "&cam&" '<PULA>"
                                                input = input & "
        rs"&session("campo0")&".Open sql,conn<PULA>"
                                                input = input & "                                   Text = Text &
'arr_"&campo&"[' & i & '] = new Array()' & br & 'arr_"&campo&"[' & i & '][0] = new Array( j'j', j'Escolha uma opção no campo
acimaj' )' & br<PULA>"
                                                input = input & "                                   Value = Value &
'ids_arr_"&campo&"[' & i & '] = new Array()' & br & 'ids_arr_"&campo&"[' & i & '][0] = new Array( j'j', j'j' )' & br<PULA>"
                                                input = input & "                           i2 = 1<PULA>"
                                                input = input & "                           if not
rs"&session("campo0")&".BOF then rs"&session("campo0")&".MoveFirst<PULA>"
                                                input = input & "                           while not
rs"&session("campo0")&".EOF <PULA>"
                                                input = input & "                                   set rs"&campo&"
= server.CreateObject('ADODB.RECORDSET')<PULA>"
                                                input = input & "                                   sql = ' SELECT *
FROM "&Trim(rsTemp("tab_nome"))&"  '<PULA>"
                                                input = input & "                                   sql = sql & '
where "&cam&" = ' & rs"&session("campo0")&"('id') <PULA>"
                                                input = input & "                                   sql = sql & ' order
by "&campo&" '<PULA>"
                                                input = input & "
        rs"&campo&".Open sql,conn<PULA>"
                                                input = input & "                                   Text = Text &
'arr_"&campo&"[' & i& ']['& i2 & '] = new Array( j'j' '<PULA>"
                                                input = input & "                                   Value = Value &
'ids_arr_"&campo&"[' & i& ']['& i2 & '] = new Array( j'j"<PULA>"
                                                input = input & "                                   cont = 1<PULA>"
                                                input = input & "                                   while not
rs"&campo&".EOF <PULA>"
                                                input = input & "                                           Text =
Text & ', j'' & rs"&campo&"('"&campo&"') & 'j' '<PULA>"
                                                input = input & "                                           Value =
Value & ',j''& rs"&campo&"('id') & 'j' '<PULA>"
```

```
                                                input = input & "
        rs"&campo&".MoveNext<PULA>"
                                                input = input & "                                                Cont =
Cont + 1<PULA>"
                                                input = input & "                                        wend<PULA>"
                                                input = input & "                                        Text = Text & ') '
& br<PULA>"
                                                input = input & "                                        Value = Value & ')
' & br<PULA>"
                                                input = input & "                                        i2 = i2 +
1<PULA>"
                                                input = input & "
        rs"&session("campo0")&".MoveNext<PULA>"
                                                input = input & "                                wend<PULA>"
                                                input = input & "            i = i + 1<PULA>"
                                                input = input & "
        rs"&session("cam0")&".MoveNext<PULA>"
                                                input = input & "                                wend<PULA>"
                                                input = input & "                                response.write
Text<PULA>"
                                                input = input & "                                response.write
Value<PULA>"
                                                input = input & "                                </asp><PULA>"
                                                input = input & "                                "&campo&" =
document.getElementById( '"&campo&"' )<PULA>"
                                                input = input & "                                "&session("campo0")&" =
document.getElementById( '"&session("campo0")&"' )<PULA>"
                                                input = input & "                                "&session("cam0")&" =
document.getElementById( '"&session("cam0")&"' )<PULA>"
                                                input = input & "<PULA>"
                                                input = input & "                                function
esvazia_"&campo&"() {<PULA>"
                                                input = input & "                                 while( "&campo&".length )
{<PULA>"
                                                input = input & "
        "&campo&".remove( "&campo&".length - 1 )<PULA>"
                                                input = input & "                                 }<PULA>"
                                                input = input & "                                <PULA>"
                                                input = input & "                                        option =
document.createElement('OPTION')<PULA>"
                                                input = input & "                                        option.text = '
';<PULA>"
                                                input = input & "                                        option.value = '
';<PULA>"
                                                input = input & "                                        (
"&campo&".options.add ) ? "&campo&".options.add( option ) : "&campo&".add( "&campo&",
"&campo&".options["&campo&".length] );<PULA>"
                                                input = input & "
        "&campo&".selectedIndex = 0;<PULA>"
                                                input = input & "                                 }<PULA>"
                                                input = input & "                                function
esvazia_cadastre_"&campo&"() {<PULA>"
                                                input = input & "                                 while( "&campo&".length )
{<PULA>"
                                                input = input & "
        "&campo&".remove( "&campo&".length - 1 )<PULA>"
                                                input = input & "                                 }<PULA>"
                                                input = input & "                                        option =
document.createElement('OPTION')<PULA>"
                                                input = input & "                                        option.text = '
';<PULA>"
                                                input = input & "                                        option.value = '
';<PULA>"
                                                input = input & "                                        (
"&campo&".options.add ) ? "&campo&".options.add( option ) : "&campo&".add( option, "&campo&".options["&campo&".length]
);<PULA>"
                                                input = input & "                                <PULA>"
                                                input = input & "                                        option =
document.createElement('OPTION')<PULA>"
                                                input = input & "                                        option.text =
'Cadastre um registro';<PULA>"
                                                input = input & "                                        option.value = '
';<PULA>"
```

```
                                                    input = input & "                                   (
"&campo&".options.add ) ? "&campo&".options.add( option ) : "&campo&".add( option, "&campo&".options["&campo&".length]
);<PULA>"
                                                    input = input & "
        "&campo&".selectedIndex = 0;<PULA>"
                                                    input = input & "                               }<PULA>"
                                                    input = input & "                               function
esvazia_cadastre_acima_"&campo&"() {<PULA>"
                                                    input = input & "                               while( "&campo&".length )
{<PULA>"
                                                    input = input & "
        "&campo&".remove( "&campo&".length - 1 )<PULA>"
                                                    input = input & "                               }<PULA>"
                                                    input = input & "                               <PULA>"
                                                    input = input & "                                       option =
document.createElement('OPTION')<PULA>"
                                                    input = input & "                                       option.text = '
';<PULA>"
                                                    input = input & "                                       option.value = '
';<PULA>"
                                                    input = input & "                                       (
"&campo&".options.add ) ? "&campo&".options.add( option ) : "&campo&".add( option, "&campo&".options["&campo&".length]
);<PULA>"
                                                    input = input & "                               <PULA>"
                                                    input = input & "                                       option =
document.createElement('OPTION')<PULA>"
                                                    input = input & "                                       option.text =
'Cadastre um registro para o campo acima';<PULA>"
                                                    input = input & "                                       option.value = '
';<PULA>"
                                                    input = input & "                                       (
"&campo&".options.add ) ? "&campo&".options.add( option ) : "&campo&".add( option, "&campo&".options["&campo&".length]
);<PULA>"
                                                    input = input & "
        "&campo&".selectedIndex = 0;<PULA>"
                                                    input = input & "                               }<PULA>"
                                                    input = input & "                               function
update_"&campo&"() {<PULA>"
                                                    input = input & "                               if
(verifica(document.form1."&session("campo0")&")) {<PULA>"
                                                    input = input & "
        "&dependente&"<PULA>"
                                                    input = input & "                               while( "&campo&".length )
{<PULA>"
                                                    input = input & "
        "&campo&".remove( "&campo&".length - 1 )<PULA>"
                                                    input = input & "                               }<PULA>"
                                                    input = input & "                               while(
arr_"&campo&"["&session("cam0")&".selectedIndex]["&session("campo0")&".selectedIndex].length > "&campo&".length )
{<PULA>"
                                                    input = input & "                                       option =
document.createElement('OPTION')<PULA>"
                                                    input = input & "                                       option.text =
arr_"&campo&"["&session("cam0")&".selectedIndex]["&session("campo0")&".selectedIndex]["&campo&".length]<PULA>"
                                                    input = input & "                                       option.value =
ids_arr_"&campo&"["&session("cam0")&".selectedIndex]["&session("campo0")&".selectedIndex]["&campo&".length];<PULA>"
                                                    input = input & "                                       (
"&campo&".options.add ) ? "&campo&".options.add( option ) : "&campo&".add( option, "&campo&".options["&campo&".length]
);<PULA>"
                                                    input = input & "                                       <asp> if
index"&campo&"&"<>" then </asp><PULA>"
                                                    input = input & "
        "&campo&".selectedIndex = <asp>=index"&campo&"</asp>;<PULA>"
                                                    input = input & "                                       <asp> end
if</asp><PULA>"
                                                    input = input & "                               }<PULA>"
                                                    input = input & "                               }<PULA>"
                                                    input = input & "                               }<PULA>"
                                                    input = input & "
        update_"&campo&"()<PULA>"
                                                    input = input & "       </SCRIPT><PULA>"

                                                    end if
                                                    dependentes = dependentes + 1
```

```
                                        end if
                                    input = input & "<asp>if request.QueryString('tipo')='inserir' then</asp><script
language='JavaScript'>seleciona_select(document.form1."&Trim(rsCampos("cam_desc"))&",opener.document.form1."&Trim(rs
Campos("cam_desc"))&".value);</script><asp>end if</asp>"
                                    rsTemp.close
                                    set rsTemp = nothing
                                    rsTemp1.close
                                    set rsTemp1 = nothing
                                else
                                    if Trim(rsCampos("cam_tipo")) = "text" then
                                        if Trim(rsCampos("cam_valida")) = "cpf" then
                                            input = "<input name='"&Trim(rsCampos("cam_nom"))&"' type='text'
maxlenght='"&rsCampos("cam_caracte")&"' size='"&rsCampos("cam_tam")&"' class='campo'
onChange='return(valida_CPF("&Trim(rsCampos("cam_nom"))&"))' <asp>if request('tipo') = ''
then</asp>onFocus='foco(this,j'"&documentacao&"j',j'#D8E4EBj');' onBlur='foco(this,j'j',j'#FFFFFFj');' <asp>end if</asp>>"
                                        elseif Trim(rsCampos("cam_valida")) = "cnpj" then
                                            input = "<input name='"&Trim(rsCampos("cam_nom"))&"' type='text'
maxlenght='"&rsCampos("cam_caracte")&"' size='"&rsCampos("cam_tam")&"' class='campo'
onChange='return(valida_CNPJ(document.form."&Trim(rsCampos("cam_nom"))&".value))' <asp>if request('tipo') = ''
then</asp>onFocus='foco(this,j'"&documentacao&"j',j'#D8E4EBj');' onBlur='foco(this,j'j',j'#FFFFFFj');' <asp>end if</asp>>"
                                        else
                                            input = "<input name='"&Trim(rsCampos("cam_nom"))&"' type='text'
maxlenght='"&rsCampos("cam_caracte")&"' size='"&rsCampos("cam_tam")&"' class='campo' <asp>if request('tipo') = ''
then</asp>onFocus='foco(this,j'"&documentacao&"j',j'#D8E4EBj');' onBlur='foco(this,j'j',j'#FFFFFFj');' <asp>end if</asp>>"
                                        end if
                                    elseif Trim(rsCampos("cam_tipo")) = "textarea" then
                                        input = "<textarea name='"&Trim(rsCampos("cam_nom"))&"'
maxlenght='"&rsCampos("cam_caracte")&"' cols='"&rsCampos("cam_tam")&"' rows='5' class='campo' <asp>if request('tipo') = ''
then</asp>onFocus='foco(this,j'"&documentacao&"j',j'#D8E4EBj');' onBlur='foco(this,j'j',j'#FFFFFFj');' <asp>end
if</asp>></textarea>"
                                    elseif Trim(rsCampos("cam_tipo")) = "select" then
                                        input = "<select name='"&Trim(rsCampos("cam_nom"))&"' class='campo'
<asp>if request('tipo') = '' then</asp>onFocus='foco(this,j'"&documentacao&"j',j'#D8E4EBj');' onBlur='foco(this,j'j',j'#FFFFFFj');'
<asp>end if</asp>><PULA>"
                                        input = input & "<option></option><PULA>"
                                        set rsTemp = conn.execute("select o.opc_opcao, o.opc_valor from
net_seleciona s, net_opcoes o where o.sel_id = s.sel_id and s.sel_id = "&rsCampos("sel_id"))
                                        do while not rsTemp.eof
                                            input = input & "<option
value='"&Trim(rsTemp("opc_valor"))&"'>"&Trim(rsTemp("opc_opcao"))&"</option><PULA>"
                                        rsTemp.movenext
                                        loop
                                        input = input & "</select>"
                                    elseif Trim(rsCampos("cam_tipo")) = "checkbox" then
                                        input = ""
                                        set rsTemp = conn.execute("select o.opc_opcao, o.opc_valor from
net_seleciona s, net_opcoes o where o.sel_id = s.sel_id and s.sel_id = "&rsCampos("sel_id"))
                                        do while not rsTemp.eof
                                            input = input & "<input type='checkbox'
name='"&Trim(rsCampos("cam_nom"))&"' value='"&Trim(rsTemp("opc_valor"))&"' class='campo' <asp>if request('tipo') = ''
then</asp>onFocus='foco(this,j'"&documentacao&"j',j'#D8E4EBj');' onBlur='foco(this,j'j',j'#FFFFFFj');' <asp>end
if</asp>>"&Trim(rsTemp("opc_opcao"))&"  <PULA>"
                                        rsTemp.movenext
                                        loop
                                    end if
                                end if
                                    codigo = codigo & "          <tr> <PULA>"
                                    codigo = codigo & "            <td
bgcolor='E9E9E9'>"&Trim(rsCampos("cam_desc"))&"</td><PULA>"
                                    codigo = codigo & "            <td bgcolor='#FFFFFF'>"&input&"<PULA>"
                                    codigo = codigo & "            </td><PULA>"
                                    codigo = codigo & "          </tr><PULA>"
                        end if
                        rsCampos.movenext
                        loop

                        codigo = codigo & "          <tr> <PULA>"
                        codigo = codigo & "            <td bgcolor='E9E9E9' > </td><PULA>"
                        rsCampos.moveFirst
                        var = ""
                        do while not rsCampos.eof
                            if Trim(rsCampos("cam_valida")) = "e-mail" then
                                var = var & "j'"&Trim(rsCampos("cam_nom"))&"j',j'j',j'RisEmailj',"
                            end if
```

```
                                    if Trim(rsCampos("cam_valida")) = "n&uacute;mero" then
                                            var = var & "j'"&Trim(rsCampos("cam_nom"))&"j',j'j',j'RisNumberj',"
                                    end if
                                    if (rsCampos("cam_req")) and ((Trim(rsCampos("cam_tipo")) = "text") or
(Trim(rsCampos("cam_tipo")) = "textarea")) then
                                            var = var & "j'"&Trim(rsCampos("cam_nom"))&"j',j'j',j'Rj',"
                                    end if
                                    if rsCampos("cam_idrelac") <> 0 then
                                            var = var & "j'"&Trim(rsCampos("cam_desc"))&"j',j'j',j'RisNumberj',"
                                    end if
                            rsCampos.movenext
                            loop
                            if var <> "" then
                                    codigo = codigo & "          <td bgcolor='E9E9E9'> <input name='imageField' type='image'
onClick='MM_validateForm("&left(var,len(var)-1)&");return document.MM_returnValue;' src='img/salvar.gif' width='85' height='21'
border='0'> <PULA>"
                             else
                                    codigo = codigo & "          <td bgcolor='E9E9E9'> <input name='imageField' type='image'
src='img/salvar.gif' width='85' height='21' border='0'> <PULA>"
                            end if
                            codigo = codigo & "            </td><PULA>"
                            codigo = codigo & "         </tr><PULA>"
                            codigo = codigo & "       </form><PULA>"
                            codigo = codigo & "     </table><PULA>"
                            codigo = codigo & "</td><PULA>"
                            codigo = codigo & "  </tr><PULA>"
                            codigo = codigo & "</table><PULA>"

                            call CriaArquivo(pasta,tabela&"_inserir.asp",html(codigo))
                            'FIM DA GERAÇÃO DAS TELAS DE INSERÇÃO

'*************************************************************************

...
```