

UNIVERSIDADE REGIONAL DE BLUMENAU
CENTRO DE CIÊNCIAS EXATAS E NATURAIS
CURSO DE CIÊNCIAS DA COMPUTAÇÃO
(Bacharelado)

**PROTÓTIPO DE APLICATIVO PARA ESCRITÓRIO DE
ADVOCACIA COM ACESSO A INTERNET BASEADO EM
ASP**

TRABALHO DE CONCLUSÃO DE CURSO SUBMETIDO À UNIVERSIDADE
REGIONAL DE BLUMENAU PARA OBTENÇÃO DOS CRÉDITOS DE DISCIPLINA
COM O NOME EQUIVALENTE NO CURSO DE CIÊNCIAS DA COMPUTAÇÃO –
BACHARELADO

GILMAR VEREANO RADUENZ

BLUMENAU, JUNHO DE 2000

2000/1-27

PROTÓTIPO DE APLICATIVO PARA ESCRITÓRIO DE ADVOCACIA COM ACESSO A INTERNET BASEADO EM ASP

GILMAR VEREANO RADUENZ

ESTE TRABALHO DE CONCLUSÃO DE CURSO FOI JULGADO ADEQUADO
PARA OBTENÇÃO DOS CRÉDITOS DA DISCIPLINA DO TRABALHO DE
CONCLUSÃO DE CURSO OBRIGATÓRIA PARA OBTENÇÃO DO TÍTULO DE:

BACHAREL EM CIÊNCIAS DA COMPUTAÇÃO

Prof. Wilson Pedro Carli — Orientador na FURB

Prof. José Roque Voltolini da Silva — Coordenador do TCC

BANCA EXAMINADORA

Prof. Wilson Pedro Carli

Prof. Marcel Hugo

Prof. Oscar Dalfovo

DEDICATÓRIA

Dedico este trabalho à minha noiva Vanessa pelo apoio dado neste trabalho, em especial à minha mãe Irma que indiretamente colaborou muito para que fosse possível a realização deste curso e ao meu orientador Wilson Pedro Carli que me auxiliou com muita dedicação, sem o qual este trabalho não teria se concretizado.

AGRADECIMENTOS

Ao coordenador do trabalho de conclusão de curso, Prof. José Roque Voltolini da Silva por toda a atenção e apoio dispensados.

A todos aqueles que, direta ou indiretamente, contribuíram para a realização deste trabalho.

SUMÁRIO

DEDICATÓRIA	III
AGRADECIMENTOS.....	IV
SUMÁRIO	V
LISTA DE FIGURAS	VIII
LISTA DE ABREVIATURAS.....	X
RESUMO.....	XII
ABSTRACT.....	XIII
1. INTRODUÇÃO	1
1.1 OBJETIVOS	2
1.2 ORGANIZAÇÃO DO TEXTO	2
2. ADVOCACIA	3
3. A INTERNET.....	6
3.1 WWW – WORLD WIDE WEB	6
3.2 ASP – ACTIVE SERVER PAGES	10
3.2.1 PROCESSAMENTO DE UM ARQUIVO ASP	10
3.2.2 RECURSOS DO ASP	12
3.2.3 VANTAGENS E DESVANTAGENS DO ASP	13
3.2.4 ARQUITETURA NECESSÁRIA PARA EXECUTAR PÁGINAS ASP	14

3.2.5	A HIERARQUIA DOS OBJETOS DAS PÁGINAS ASP.....	17
3.2.6	ACESSO A BANCO DE DADOS COM ASP	18
3.3	ASP COMPARADO A OUTRAS TECNOLOGIAS	20
3.3.1	COMMON GATEWAY INTERFACE (CGI).....	20
3.3.2	INTERNET SERVER API.....	21
3.3.3	INTERNET DATABASE CONNECTOR.....	22
3.3.4	TECNOLOGIAS ADICIONAIS PARA IIS	22
4.	PROJETO DO SOFTWARE	24
4.1	FERRAMENTAS UTILIZADAS	25
4.1.1	FRONTPAGE 2000.....	25
4.1.2	POWERDESIGNER 6.1	26
4.1.3	ACCESS 97	26
4.1.4	VISUAL BASIC 6.0.....	27
4.2	FUNCIONALIDADE	28
4.3	ESPECIFICAÇÃO	29
4.4	APRESENTAÇÃO DO PROTÓTIPO	33
4.4.1	MENU PRINCIPAL.....	33
4.4.2	MANTER CADASTRO DE PROCESSOS	34
4.4.3	MANTER CADASTRO DE AUTORES E RÉUS	34
4.4.4	MANTER CADASTRO DE CUSTAS DE PROCESSO	35
4.4.5	MANTER CADASTRO DE ANDAMENTO DE PROCESSOS.....	36
4.4.6	MANTER CADASTRO DE CLIENTES	37
4.4.7	MANTER CADASTRO DE ADVOGADOS	37
4.4.8	MANTER CADASTRO DE COMARCAS	37
4.4.9	MANTER CADASTRO DE TIPOS DE PROCESSO.....	39
4.4.10	MANTER CADASTRO DE TIPOS DE AÇÃO.....	39
4.4.11	MANTER CADASTRO TIPOS DE FASE PROCESSUAL.....	39
4.4.12	CONSULTAS DO APLICATIVO.....	40
4.4.13	RELATÓRIOS DO APLICATIVO	40

4.5	PÁGINAS NA INTERNET	41
4.5.1	PÁGINA PRINCIPAL NA INTERNET	42
4.5.2	PÁGINA CONSULTA DE PROCESSOS	42
4.5.3	PÁGINA CADASTRO DE CLIENTES	45
4.5.4	PÁGINA ESCRITÓRIOS ASSOCIADOS	47
5.	CONCLUSÃO	48
5.1	DIFICULDADES.....	49
5.2	SUGESTÕES PARA TRABALHOS FUTUROS.....	49
6.	REFERÊNCIA BIBLIOGRÁFICA	50

LISTA DE FIGURAS

Figura 3.1.1 : funcionamento da Internet	8
Figura 3.2.1.1 : processamento de um arquivo ASP	11
Figura 3.2.2.1 : documentos com hipertexto	13
Figura 3.2.5.1 : hierarquia dos objetos das páginas ASP	18
Figura 3.3.1 : diferenças entre tecnologias ISAPI,ASP,CGL,IDC	20
Figura 4.1.1.1 : tela principal do Frontpage 2000	25
Figura 4.1.2.1 : tela principal do Powerdesign 6.1	26
Figura 4.1.3.1 : tela principal do Access 97	27
Figura 4.1.4.1 : tela principal do Visual Basic 6.0	28
Figura 4.2.1 : consulta de processos	28
Figura 4.2.2 : rotina de replicação	29
Figura 4.3.1 : diagrama de contexto	30
Figura 4.3.2 : diagrama de fluxo de dados	30
Figura 4.3.3 : modelo entidade relacionamento (MER) lógico	32
Figura 4.4.1.1 : tela do menu principal.....	33
Figura 4.4.2.1 : tela de cadastro de processos	34
Figura 4.4.3.1 : tela de cadastro de autores/réus.....	35
Figura 4.4.4.1 : tela de custas de processos	35
Figura 4.4.5.1 : tela de andamento de processos	36
Figura 4.4.6.1 : tela de cadastro de clientes	37
Figura 4.4.7.1 : tela de cadastro de advogados.....	37
Figura 4.4.8.1 : tela cadastro de comarcas.....	37
Figura 4.4.9.1 : tela cadastro de tipos de processo	38
Figura 4.4.10.1 : tela cadastro de tipos de ação	38
Figura 4.4.11.1 : tela cadastro de tipos de fase processual.....	38
Figura 4.4.12.1 : consulta de clientes	39
Figura 4.4.12.2 : consulta relação de processos	39
Figura 4.4.13.1 : exemplo de relatório de clientes.....	40

Figura 4.4.13.2 : exemplo de relação de processos	40
Figura 4.5.1 : tela principal do <i>Personal Web Server</i>	41
Figura 4.5.1.1 : página principal do escritório na Internet	42
Figura 4.5.2.1 : página consulta de processos	42
Figura 4.5.2.2 : código ASP da página consulta de processos	43
Figura 4.5.3.1 : página cadastro de clientes.....	44
Figura 4.5.3.2 : página cadastro cliente do sistema	44
Figura 4.5.3.2 : código ASP da página cadastrar cliente	45
Figura 4.5.4.1 : página escritórios associados	46

LISTA DE ABREVIATURAS

ADO	<i>Activex Data Object</i>
ARPA	<i>Advanced Research and Projects Agency</i>
ASP	<i>Active Server Pages</i>
CERN	<i>Conseil Européen pour la Recherche Nucléaire</i>
CGI	<i>Common Gateway Interface</i>
CNAS	Conselho Nacional de Assistência Social
COM	<i>Component Object Model</i>
DAO	<i>Data Access Objects</i>
DLL	<i>Dynamic Link Libraries</i>
DSN	<i>Data Source Name</i>
FTP	<i>File Transfer Protocol</i>
HTML	<i>Hipertext Markup Language</i>
HTTP	<i>Hipertext Transfer Protocol</i>
IDC	<i>Internet Database Connector</i>
IDE	<i>Integrated Development Environment</i>
ISAPI	<i>Internet Server API</i>
ISM	<i>Optional Internet Service Manager</i>

IIS	<i>Internet Information Server</i>
MMC	<i>Microsoft Management Console</i>
MTS	<i>Microsoft Transaction Server</i>
NCSA	<i>National Center for Supercomputing Applications</i>
ODBC	<i>Open Database Connectivity</i>
OLE DB	<i>Object Linking and Embedding Database</i>
PERL	<i>Practical Extraction And Report Language</i>
PWS	<i>Personal Web Server</i>
RDO	<i>Remote Data Objects</i>
RDS	<i>Remote Data Services</i>
SGML	<i>Standard Generalize Markup Language</i>
SQL	<i>Structured Query Language</i>
TCP/IP	<i>Transmission Control Protocol / Internet Protocol</i>
URL	<i>Uniform Resource Locators</i>
VBA	<i>Visual Basic for Applications</i>
WWW	<i>World Wide Web</i>

RESUMO

Este trabalho visa demonstrar através de um protótipo de software aplicativo para Escritório de Advocacia, o intercâmbio das informações de uma base de dados do sistema e a Internet através de *Active Server Pages* (ASP), um ambiente de *script* de servidores. O software foi desenvolvido na linguagem Visual Basic versão 6.0 com acesso a dados *ActiveX Data Object* (ADO), utilizando como base de dados o Access 97.

ABSTRACT

This work seeks to demonstrate through a prototype of a specific software for Lawer's Office, the exchange of the information of a base of data of the system and the Internet through Active Server Pages (ASP), a language of script of servers. The software was developed in the Visual Basic language version 6.0 with data access Activex Data Object (ADO), using as base of data Access 97.

1. INTRODUÇÃO

Sabe-se que a Internet evolui dia a dia em um ritmo cada vez mais acelerado, e o volume de acessos e de negócios cresce exponencialmente. De acordo com [WIL1999], as páginas da Internet ficam normalmente guardadas em servidores, onde são colocados uns endereços nas páginas, ou seja, um endereço universal. Ao invés de c:\meusdocumentos..., utiliza-se `http://www.endereço.tipo_da_pagina`. Para conseguirmos mais recursos com um tamanho de páginas menores, foi criada uma linguagem chamada HTML (linguagem de marcação de hipertexto), que diminuía muito o tamanho dos arquivos e facilitava a veiculação destes pela rede. Para vê-los era necessário ter um programa próprio, o *browser*. Também foram criados os *e-mails*, para trocas de mensagens, ainda sem poder anexar arquivos.

A internet foi crescendo e tornou-se um campo de possibilidades incontáveis, ampliaram-se as propriedades da linguagem HTML, surgiu o *Javascript* e em seguida o *Jscript*, que junto com o HTML permitem dar mais efeitos a página. A partir de 1997 os servidores com sistema operacional Unix começaram uma revolução na Internet: surgiram os *scripts* que rodam no servidor com uma interface de comunicação comum, a *Common Gateway Interface* (CGI), onde o usuário entrava com dados em um formulário e enviava para um *script* em CGI, que lia, interpretava e processava estes *scripts*. ([WIL1999]).

As *Active Server Pages* (ASP) surgiram do *Vbscript*, uma linguagem de *Script* que utiliza os mesmos padrões do Visual Basic e que foi utilizada em CGI. ASP é uma linguagem da Microsoft que permite desenvolver páginas da WEB dinâmicas, possibilitando desde processos simples até o desenvolvimento de aplicativos avançados e acesso a banco de dados. As rotinas de ASP podem ser escritas junto com o HTML, porém elas são interpretadas por um servidor WEB e enviadas para o *browser* no formato de HTML, portanto todo processamento do ASP é feito no servidor. Segundo [WIL1999], *Active Server Pages* (ASP) é uma forma de fazer páginas mais dinâmicas. Quando cria-se uma página ASP, tem-se a disposição vários recursos. Pode-se usar *Activex*, *VBScript*, *JScript*, *JavaScript* e juntar essas linguagens com o HTML.

Para facilitar a consulta dos processos em andamento no escritório de advocacia, a busca e o envio de informações relativas a estes processos, viu-se a necessidade de

desenvolver um software incorporando os recursos e meios de acessos a dados através da internet utilizando as páginas de servidor ativas (ASP). Sugere-se que o aplicativo disponibilize em uma página na internet informações específicas sobre os processos, como o andamento de processos, despesas referente aos processos, agenda de audiências, e ofereça consultas baseadas em critérios definidos pelo cliente, permitindo receber ou enviar dados aos usuários, buscando informações em *sites* e alimentando a base de dados. O protótipo do software também tem a finalidade de fazer o controle interno do escritório, como cadastramento de clientes, processos, advogados, gerar relatórios e disponibilizar opções de consultas. Na especificação do protótipo será utilizada a metodologia de análise estruturada e a ferramenta case *Powerdesigner* [SYB2000], para gerar os diagramas e a modelagem dos dados.

1.1 OBJETIVOS

O objetivo principal do trabalho proposto é implementar um protótipo de um Software Aplicativo para Escritório de Advocacia com recursos voltados à Internet através de Páginas de Servidor Ativas (ASP) que possibilitam enviar informações referentes ao andamento de processos para os clientes que acessam a internet, bem como buscar informações em determinados *sites* e atualizá-los na base de dados do software aplicativo.

1.2 ORGANIZAÇÃO DO TEXTO

O capítulo 1 apresenta uma introdução sobre o trabalho com seus objetivos.

O capítulo 2 tem como finalidade dar uma visão geral de um escritório de advocacia, com definições e principais áreas em que atuam.

O capítulo 3 conta a história da Internet desde seu surgimento, as tecnologias e formas de acesso as páginas, linguagens utilizadas e descreve com maiores detalhes o ASP, seu funcionamento, arquitetura e requisitos de sistema necessários para sua execução.

O capítulo 4 está voltado ao desenvolvimento do protótipo, onde são apresentadas a modelagem de dados, as telas de entrada de dados e também as páginas da internet para o acesso aos dados do software aplicativo. O capítulo 5 finaliza o trabalho com uma conclusão e sugestões para trabalhos futuros.

2. ADVOCACIA

Os Escritórios de Advocacia são organizações voltadas ao atendimento das necessidades de pessoas físicas e jurídicas nas áreas de direito, como o tributário, civil, comercial, trabalhista, internacional, filantrópica e outras [THE1991]. Alguns dos serviços ou segmentos de consultoria e assessoria jurídica em que estes escritórios atuam são descritos a seguir.

Os escritórios executam operações no segmento tributário, atuando na recuperação de créditos, planejamento fiscal e teses jurídico-tributárias, visando planejar, administrar e recuperar tributos.

Na área comercial, atuam principalmente na análise e discussão judicial de contratos bancários, marcas e patentes, bem como no contencioso civil visando preservar os interesses de seus clientes nas disputas judiciais em todos os segmentos, principalmente em questões de cunho indenizatório de dano moral, material e estético, desde o juízo singular até os tribunais superiores.

Na esfera trabalhista, pode-se destacar o planejamento estratégico concernente ao direito preventivo, objetivando minimizar ao máximo eventuais riscos expostos pelas empresas, negociações coletivas, na defesa de ações trabalhistas, em todas as áreas, com representação e acompanhamento, amigável e litigioso em todos os âmbitos judiciais, inclusive em instâncias superiores.

Os escritórios de advocacia também desempenham planejamento tributário-fiscal, renovação e cadastramento de certificados de filantropia, e recursos administrativos frente ao Conselho Nacional de Assistência Social (CNAS) na área filantrópica. E no âmbito internacional atuam no assessoramento de abertura de filiais no exterior, planejamento de exportações, contencioso internacional, aos seus clientes serviços de consultoria e assessoria jurídica especializada que agregam efetiva segurança aos seus negócios.

Para compreender melhor este trabalho que foi desenvolvido para um escritório de advocacia foram relacionados mais alguns conceitos importantes ligados à área jurídica. O trabalho no escritório está diretamente relacionado com processos, que segundo [THE1991] é

o conteúdo sistemático de uma série de atos que formam o procedimento judicial, isto é, a forma de agir em juízo, que se impõe entre o pedido da parte e o provimento jurisdicional. Um processo pode ter vários processos incidentes. O processo incidente é uma relação jurídica que surge paralelamente ao processo principal, ou seja, é um novo processo que segue procedimentos distintos, porém tem sua dependência vinculada ao processo principal. Todo processo é encaminhado a uma Vara ou Jurisdição que tem a finalidade de formular e fazer atuar praticamente a regra jurídica concreta que, por força do direito vigente, disciplina determinada situação jurídica. A sede onde se realiza a atividade jurisdicional é denominada Comarca.

No processo, dependendo do tipo de ação, procedimento ou fase processual, a denominação das partes varia, na lei e na terminologia forense. Assim, autor e réu são denominações usuais no processo de conhecimento em geral. Denomina-se ação como sendo o conflito de interesse qualificado por uma pretensão resistida.

Pode-se ter vários tipos de processo, que são uma separação doutrinária que é utilizada para determinar as diferentes especializações dos vários ramos do direito, tendo em vista a complexidade e a que sociedade atualmente se encontra. Desta forma pode-se citar os seguintes tipos de processo mais conhecidos:

- a) trabalhista: que está vinculado aos órgãos de julgamento federal de natureza especializada, responsáveis pela resolução das questões trabalhistas;
- b) administrativo: encontra-se vinculado às resoluções adstritas dos diversos órgãos da administração, quer seja ela federal, estadual ou municipal. São os diversos órgãos das respectivas repartições responsáveis pelo julgamento administrativo em que a pessoa física ou jurídica esteja discutindo com estado, ou vice versa. Superada esta fase administrativa tem-se a fase judicial;
- c) civil: o tipo de processo civil é de responsabilidade de julgamento do respectivo estado e assim sendo ficam englobadas no processo civil todas as discussões com exceção das reclamatórias trabalhistas e aqueles em que a União for parte;
- d) tributário: no processo tributário relaciona-se a discussão vinculada aos tributos municipais, estaduais e federais.

Na seqüência, temos a Fase Processual que é definida como sendo a instância em que o processo se encontra em julgamento, neste sentido a título exemplificativo, pode-se dizer que

um processo está em primeira instância quando se encontra sob a análise e julgamento de um juiz singular, também conhecido como monocrático; e em segunda instância quando está sob apreciação de análise e julgamento de uma Turma composta por mais de um juiz. A Turma é a divisão do conjunto de julgadores responsável pela análise dos processos em nível de segundo grau. Nos processos de segundo grau existem divisões de Turmas especializadas na análise e julgamento de tipos de processos específicos a qual chama-se Câmara. Exemplo: Câmara de processos civis, criminais, entre outras.

Com o levantamento de informações sobre os escritórios de advocacia verificou-se que as operações realizadas são inicializadas com a chegada ou contato de um cliente. Quando o cliente chega ao escritório pela primeira vez, os seus dados são cadastrados. Em seguida são colhidos dados do problema e é aberto um novo processo. Este processo é cadastrado e são informadas informações como: o tipo da ação processual, o tipo do processo, a situação do processo, o nome dos Autores e Réus e seus respectivos advogados. Cada processo pode ter um ou vários Autores e Réus. As informações sobre cada advogado também são cadastradas. Cada processo pode ter vários incidentes, que são na verdade novos processos. Por exemplo, se um processo A tem um incidente B, B é um novo processo resultante de A, que por sua vez poderá ter novos incidentes: C, D e assim sucessivamente.

Os escritórios de advocacia emitem mensalmente uma relação dos clientes com seus respectivos processos em andamento e uma relação de processos por comarca. Quando da necessidade de cada advogado, são impressos também os processos individualmente para análise.

3. A INTERNET

A internet surgiu a partir da *Advanced Research and Projects Agency* (ARPA), uma agência norte-americana que objetivava conectar os computadores dos seus departamentos de pesquisa. A Internet nasceu a partir da ARPANET, que interligava quatro instituições: Universidade da Califórnia, em Los Angeles e Santa Bárbara, o Instituto de Pesquisa de Stanford e a Universidade de Utah, tendo início em 1969. Os pesquisadores e estudiosos do assunto receberam o projeto a disposição, para trabalhar. A ARPA requisitou um sistema de rede que fosse descentralizado. De forma que em caso de guerra, por mais que uma base fosse atingida, a comunicação não seria interrompida. Deste estudo que perdurou na década de 70, nasceu o *Transmission Control Protocol / Internet Protocol* (TCP/IP), grupo de protocolos que é a base da Internet desde aqueles tempos até hoje. ([NAS2000]). No final da década de 80 estudava-se um projeto que seria usado como meio de transportar informações através de organizações que se localizavam distantes uma das outras, e que teria dado início a Web e a uma interface para o *World Wide Web* (WWW).

3.1 WWW – WORLD WIDE WEB

A definição oficial descreve o *World Wide Web* como uma "iniciativa de busca de informação hipermídia a longa distância visando dar acesso a um vasto universo de documentos". De acordo com [NAS2000], o que o projeto *World Wide Web* (WWW) fez foi permitir que usuários em redes de computadores com um método de acesso compatível pudessem acessar inúmeras mídias de maneira simples. Usando uma interface chamada *Mosaic*, o projeto Web mudou a maneira das pessoas verem e criarem informação - foi criada então a primeira rede hipermídia global. O funcionamento da Web utiliza o hipertexto como meio de interação com os usuários. Hipertexto é basicamente o mesmo que um texto regular - pode ser armazenado, lido, ou editado - com uma importante diferença: o hipertexto contém conexões dentro do texto para outros documentos. A Hipermídia é um hipertexto com uma diferença - a hipermídia contém *links* não somente para outras partes de texto, mas também para outras formas de mídia - sons, imagens, e filmes. Imagens podem estar ligadas a sons ou a outros documentos.

A história da Web começou em Março de 1989, quando Tim Berners-Lee do *Conseil Européen pour la Recherche Nucléaire* (CERN), uma organização Européia de pesquisadores de Física em altas-energias, propôs que o projeto fosse usado como um meio de transportar efetivamente pesquisa e idéias através da organização. A comunicação a esse nível era um objetivo do CERN há muitos anos, uma vez que os seus membros localizam-se em diversos países. Meses após a proposta original do CERN, o *National Center for Supercomputing Applications* (NCSA) iniciou um projeto para criar uma interface para o *World Wide Web*. Uma das missões do NCSA era auxiliar a comunidade científica a produzir software amplamente disponível, e não comercial. Outro dos seus objetivos era investigar novas tecnologias na esperança de que interesses comerciais surgissem. Assim, o Grupo de Desenvolvimento de Software iniciou o desenvolvimento de uma interface versátil, multiplataforma para o *World Wide Web*, e o chamaram de *Mosaic* - o primeiro browser WWW como se conhece atualmente ([NAS2000]).

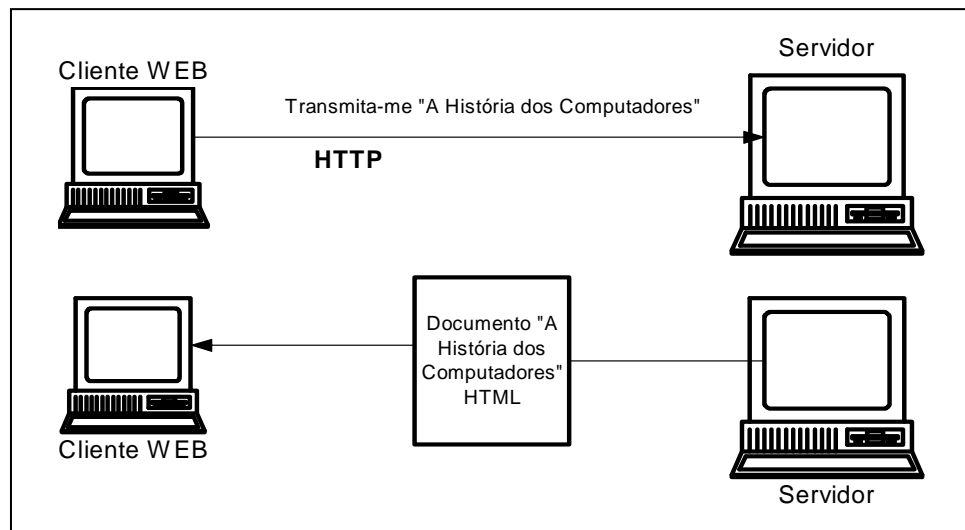
Devido ao número de serviços tradicionais que realizava, e devido a sua fácil interface "*point-and-click*", o *Mosaic* logo se tornou o browser mais popular para a Web. Mas atualmente existem inúmeros outros *browsers* disponíveis, como o Netscape, Spry e outros. Existem versões para as mais diversas plataformas, como Suns, Silicon Graphics, PCs (Microsoft Windows), Macintosh, e outros computadores rodando várias outras formas de sistemas operacionais UNIX. Como características de um cliente WWW (*browser*) pode-se descrever:

- a) interface gráfica orientada por mouse;
- b) exibição de documentos hipertexto e hipermídia;
- c) exibição de texto numa grande variedade de fontes, em negrito, itálico;
- d) formatação de documentos segundo elementos como parágrafos, listas;
- e) suporte para sons (Macintosh, formato Sun, audio, e outros);
- f) suporte para filmes (MPEG-1 e *QuickTime*);
- g) exibição de caracteres definidos no conjunto ISO 8859 (pode exibir caracteres acentuados);
- h) suporte para formulários eletrônicos interativos, com uma grande variedade de elementos, tais como *fields*, *check boxes*, e *radio buttons*;
- i) suporte para gráficos interativos (em GIF, XBM e JPEG);
- j) suporte para imagens sensíveis;

k) links hipermídia e suporte para os seguintes serviços de rede: *ftp*, *gopher*, *telnet*, *nntp*.

A Web funciona segundo o popular modelo cliente-servidor. Um servidor Web é um programa cujo único propósito é servir documentos para os clientes quando requerido. Um cliente Web é um programa que faz a interface com o usuário e realiza pedidos de documentos a um servidor especificado pelo usuário. A carga causada pelo servidor normalmente é pequena, uma vez que não realiza nenhum processamento pesado e só opera quando um documento é pedido pelo cliente ([MAR2000]). A figura 3.1.1 mostra um exemplo de como o processo funciona:

Figura 3.1.1 : funcionamento da internet



Fonte: [MAR2000]

Através de um cliente Web (também chamado de *browser*), o usuário seleciona uma parte do hipertexto ligada a outro documento - "A História dos Computadores". O cliente Web conecta-se a um computador especificado por um endereço em algum lugar da Internet e pede ao servidor Web desse computador pelo documento "A História dos Computadores". O servidor responde enviando o texto e qualquer outro elemento de mídia existente no documento (figuras, sons, ou filmes) para a tela do usuário.

A linguagem que os clientes e servidores Web usam para se comunicarem é chamada de *HyperText Transfer Protocol* (HTTP). Todos os clientes e servidores Web precisam "falar" HTTP para poderem enviar e receber documentos hipermídia. Por essa razão, servidores Web

são normalmente chamados de servidores HTTP. O termo "*World Wide Web*" é freqüentemente usado para se referenciar ao conjunto de redes de servidores que se comunicam com a linguagem HTTP e a coleção global de informação disponível através do protocolo.

A linguagem padrão que a Web usa para criar e reconhecer documentos de hipermídia é o *HyperText Markup Language* (HTML) que é uma derivação do *Standard Generalized Markup Language* (SGML), uma linguagem utilizada para formatação de documentos. Os documentos HTML nada mais são que arquivos ASCII com códigos de formatação que contém informação sobre o layout (estilos de texto, títulos de documentos, parágrafos, listas, etc.) e seus *hiperlinks*. O padrão de HTML que está sendo desenvolvido atualmente, chamado HTML+, suporta criação de formulários interativos, regiões sensíveis em imagens, layout mais versátil com mais opções de formatação e estilos, tabelas formatadas, etc.

O HTML usa os *Uniform Resource Locators* (URL) para representar *links* de hipermídia e *links* para outros serviços de rede dentro dos documentos. É possível representar quase que todos os serviços da Internet com uma URL. A primeira parte da URL (antes das duas barras) especifica o método de acesso. A segunda parte é tipicamente o endereço do computador, os dados ou a localização do serviço. Outras partes podem especificar nomes de arquivos, a porta a conectar, ou o texto para se buscar em um banco de dados. Exemplo de URL: "http://endereco_do_site/diretorio/book.html" que conecta-se a um servidor HTTP e recupera um arquivo HTML. *Hiperlinks* podem referenciar não somente outros textos e mídias, mas também a outros serviços de rede. *Browsers* Web não são simplesmente clientes Web, mas são também clientes FTP, *Gopher* e *Telnet*. O HTML+ também inclui um URL de email, assim *hiperlinks* podem ser feitos para enviar um *email* automaticamente. Por exemplo, selecionando um trecho do hipertexto, abre-se um programa de *mail* pronto para enviar um *email* para o endereço referenciado. Depois do HTML, os servidores em UNIX começaram a rodar um programa que comanda o endereçamento de páginas, uma interface de comunicação comum, o CGI, e em seguida surgiu o *Vbscript* que em pouco tempo foi utilizada em CGI dando origem as *Active Server Pages* (ASP) ([MAR2000]).

3.2 ASP – ACTIVE SERVER PAGES

O ASP (*Active Server Pages* - Páginas de Servidor Ativas) é um ambiente para programação por *scripts* no servidor, que se pode usar para criar páginas dinâmicas, interativas e de alta performance. Com as páginas ASP, os *scripts* são executados no servidor e não no cliente. É o próprio servidor que transforma os scripts em HTML padrão, fazendo com que qualquer *browser* do mercado seja capaz de acessar um site que usa ASP. Um arquivo que contém código ASP é identificado pela extensão “.asp”.

ASP é uma página em HTML estendida por comandos ASP que também controlam a aparência da página Web e *tags* de HTML, fora dos blocos de *script*, que é executado no fluxo do programa do *script* de servidor. Ela contém, basicamente, código HTML e *scripts* que irão rodar no servidor, sendo o *Vbscript* a linguagem de *scripts* mais utilizada. Pode-se usar também *Javascript*, *CGI*, *Pearl*, *C*, dentre outras.

O *Active Server Pages* é um software que acompanha o *Internet Information Server (IIS) for NT Server*, com o *Personal Web Server for Windows NT Workstation* ou o *Windows 98*. Na verdade o ASP surgiu juntamente com o lançamento do *Internet Information Server 3.0 do Windows NT*, uma solução da empresa Microsoft, que exige que o servidor rode um sistema windows 98, NT ou superior para processar páginas dinâmicas na internet. Este requisito é solicitado apenas no servidor, o ASP aceita qualquer linguagem o que significa que o programa não é limitado ao *Vbscript*, pode ser utilizada qualquer outra linguagem desde que essa linguagem suporte o modelo *Activex Scripting* e que seja alterado nas configurações do IIS. ([MAR2000]).

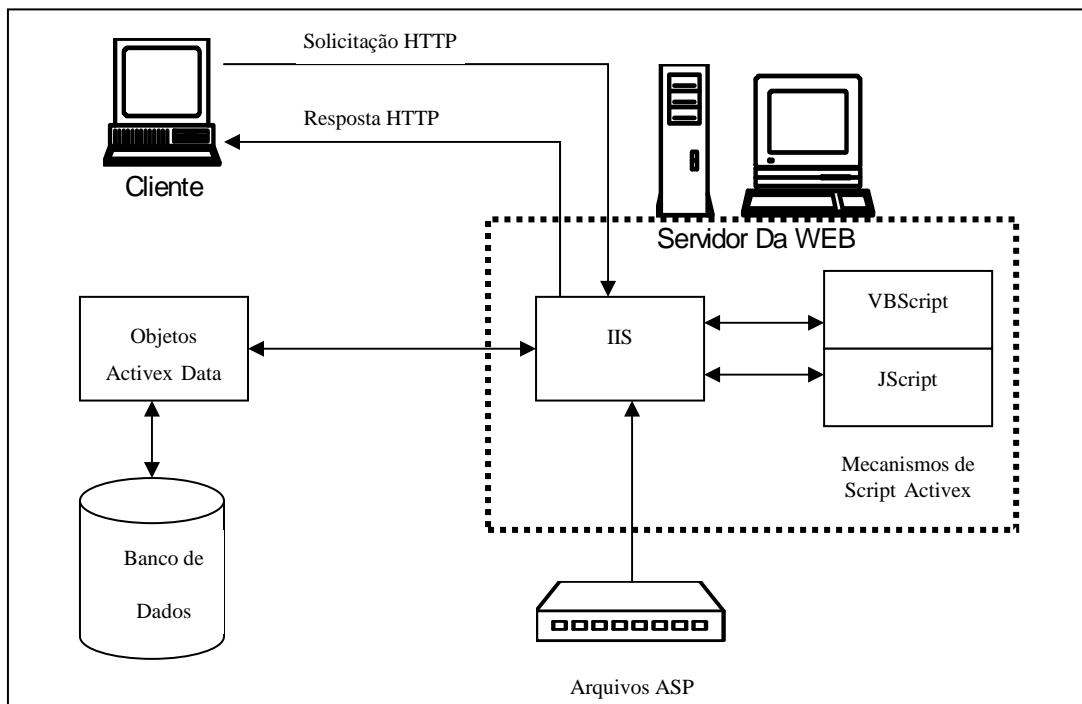
3.2.1 PROCESSAMENTO DE UM ARQUIVO ASP

O processamento de um arquivo ASP, desde sua armazenagem no disco rígido e sua manipulação no *Internet Information Server (IIS)* até o envio do resultado em HTML ao *browser* é exemplificado na figura 3.2.1.1. É importante salientar que as páginas ASP processadas são HTML básico após o servidor da Web ter concluído o processamento da página ASP. O cliente solicita a página xyz.asp do servidor. O servidor da Web verifica a extensão para ver se um programa especial (como por exemplo, o mecanismo *Active Server*

Pages) deve ser invocado para processar a solicitação. Se não houver extensão *.asp*, o servidor da Web determina que deve invocar o ASP para processar essa página.

Se essa página nunca tiver sido solicitada antes ou se tiver sido alterada desde a última solicitação, ela deverá ser analisada e a sintaxe verificada e compilada pelo servidor da Web. Caso contrário, a página poderá ser lida de uma memória de páginas processadas recentemente, o que auxilia no desempenho. Durante o processo de análise, o HTML e o código de *script* são separados. O IIS determina que mecanismo de *script* é responsável por qual parte do *script* e delega o trabalho de verificação da sintaxe e de compilação ao mecanismo de *script* adequado, como por exemplo *VBScript*.

figura 3.2.1.1 : processamento de um arquivo ASP



Fonte: [MAR2000]

Agora o código é executado pelo mecanismo de script usando recursos do IIS, que é *host* do mecanismo de *scripts*. Todos os objetos que o mecanismo de linguagem não pode manipular são solicitados do IIS, que também é responsável pela manipulação de entradas e saídas para objetos *Activex* externos criados e usados dentro do *script*. Caso ele não possa fornecer o objeto, é gerado um erro.

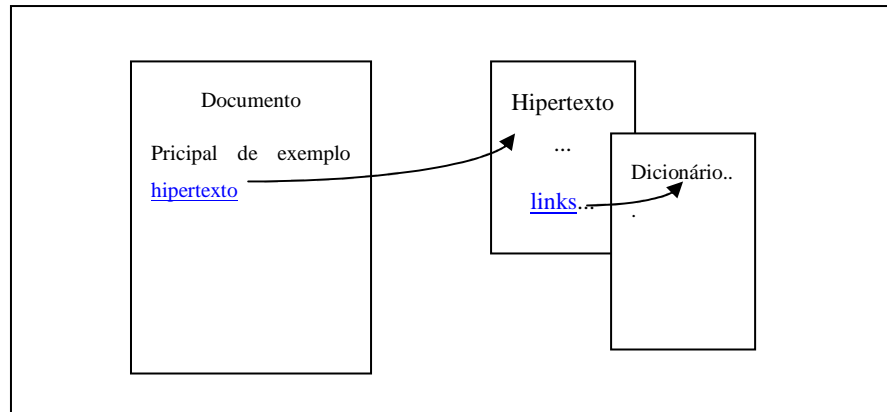
A saída de *script* e o código de HTML estático no arquivo ASP são mesclados. O HTML final é enviado ao usuário em uma resposta HTTP. Um objeto *Activex* fornece código reutilizável armazenado em um *wrapper*, que fornece acesso ao objeto através de propriedades, métodos e eventos. Adicionalmente, o código do objeto *Activex* está sendo executado no mesmo espaço de processamento da aplicação que o utiliza. O *Activex* fornece um mecanismo padrão para estender qualquer linguagem de programação ou de *script* com o uso do mesmo objeto reutilizável. *Activex Data Objects* (ADO) são objetos *Activex* que fornecem uma interface com OLE DB para que se possa acessar qualquer banco de dados através do ADO. ([MAR2000]).

3.2.2 RECURSOS DO ASP

ASP é um ambiente para programação de páginas dinâmicas e interativas que utiliza recursos do HTML, suporte a linguagens de scripts, acesso à base de dados e sessões de persistência.

O Hypertext Markup Language (HTML) ou Linguagem de marcadores de hipertexto, é a linguagem com a qual escreve-se as páginas armazenadas nos servidores e que serão visualizadas pelos usuários; o funcionamento da Web utiliza o hipertexto como meio de interação com os usuários. ([SAV1997]). Hipertexto é basicamente o mesmo que um texto regular - pode ser armazenado, lido, ou editado - com uma importante diferença: o hipertexto contém conexões dentro do texto para outros documentos. Por exemplo, suponha que você pudesse de alguma maneira selecionar (com um mouse ou com o seu dedo) a palavra "hipertexto" no parágrafo anterior a esse. Em um sistema hipertexto você poderia ter então um ou mais documentos relacionados a hipertexto - uma história do hipertexto, por exemplo, ou uma definição de um dicionário. Estes novos documentos por sua vez poderiam ter *links* e conexões para outros documentos. Dessa maneira, *links* hipertexto, chamados de *hyperlinks*, podem criar uma complexa "teia" (web) virtual de conexões. A figura 3.2.2.1 representa o que foi exposto anteriormente.

figura 3.2.2.1 : documentos com hipertexto



Fonte: [BAR2000]

O suporte a linguagens de *Scripts* como *Vbscript* ou *JavaScript* que atuam no servidor e auxiliam a incrementar outra linguagem que é o HTML, possuem opções para aplicação cliente servidor o que permite realizar operações mais avançadas com banco de dados. Existem *plug-ins ActiveX* para dar suporte para outras linguagens como *REXX*, *PERL*, e *Python*.

O acesso à Base de Dados realizada com uma aplicação ASP pode ser usada com qualquer banco de dados compatível com ODBC. Isto inclui dados do Access, Microsoft SQL Server, Oracle, Sybase, Informix, DB2, entre outros.

Outro recurso são as sessões de persistência de informações no servidor, as aplicações ASP podem armazenar dados que são mantidos durante toda uma sessão. Desta forma, um usuário pode fornecer seu nome somente uma vez em uma página e as demais páginas podem obter este dado automaticamente. Este recurso é ideal para aplicações de venda pela Internet.

3.2.3 VANTAGENS E DESVANTAGENS DO ASP

Na utilização de páginas dinâmicas com ASP foram verificadas algumas vantagens e desvantagens que podem influenciar na escolha do ambiente a ser utilizado para a criação das páginas para a internet. De acordo com [MAR2000] pode-se citar como vantagens do ASP o seguinte:

- a) independência do Browser – os *scripts* rodam no servidor e somente os resultados são enviados ao usuário, sendo que qualquer navegador pode visualizar as páginas ASP, ou seja, todos os *browsers* suportam ASP. Isto acontece pelo fato das páginas

ASP serem processadas pelo servidor. O que o cliente recebe é somente código HTML padrão;

- b) acesso a Base de Dados – contém um mecanismo fácil para visualizar, atualizar e adicionar informações à base de dados que suportam ODBC e servidores SQL;
- c) segurança do código fonte – o cliente visualiza somente o resultado do processamento no formato HTML;
- d) administração do estado – através de objetos internos da ASP pode-se controlar o estado do aplicativo com base no usuário, no aplicativo e no servidor;
- e) semelhança com Visual Basic ou VBA – caso se conheça uma dessas linguagens, o conhecimento é imediatamente aproveitado, pois pode-se usar o Vbscript e ADO.

As desvantagens mais relevantes do ASP de acordo com [MAR2000] são:

- a) a mistura de códigos HTML, Vbscript, ASP dificulta a compreensão do código fonte;
- b) ASP é vinculado aos sistemas operacionais da Microsoft, Windows. Embora existam versões de ASP para outros sistemas operacionais, estas versões não são muitas conhecidas.

3.2.4 ARQUITETURA NECESSÁRIA PARA EXECUTAR PÁGINAS ASP

De acordo com [MAR2000], para que se possa utilizar e trabalhar com páginas ASP é necessário um servidor com *Windows NT* ou *Windows NT Workstation* ou *Windows 95* ou *Windows 98*, observando que esta exigência é apenas do lado do servidor. No servidor deverá estar instalado o *Microsoft Internet Information (IIS)* no caso do *Windows NT Server*, o *Peer Web Servidor (PWS)* no *Windows NT Workstation* ou o *Personal Web Server (PWS)* nos sistemas operacionais *Windows 95* e *98*.

Para usar recursos de acesso a dados, um *driver* de ODBC deverá estar instalado e funcionando neste servidor. No caso de acessar uma base de dados Access, um *driver* ODBC Access. A instalação do Access versão 7.0 ou superior no servidor garante a presença do *driver* ODBC e facilita a criação dos bancos de dados. A existência de uma "Fonte de Dados" ODBC para este banco de dados é feita usando a opção "ODBC" do "Painel de Controle" do Windows. Para usar o SQL Server, cria-se um usuário com direito para acessar este banco de dados.

Outro detalhe importante é que no servidor deve ser criado um diretório com direito de execução para guardar as páginas ASP. As páginas ASP só podem ser rodadas a partir de um diretório com o direito de "Execução". Inclui-se este diretório na lista do *Internet Information Server* (IIS) e *Personal Web Server* (PWS), dando direito de "Execute/Execução". Para aumentar a segurança é aconselhável manter habilitada a opção de "Read/Leitura". É necessário também um navegador (*Browser*) como, por exemplo, Internet Explorer, Netscape, Spry etc...

Uma solução caseira que funciona muito bem em máquinas locais acessando base de dados Access é instalar o *Personal Web Server* incluído no Windows 98 e usar um *driver* ODBC do Access. Esta configuração permite desenvolver e testar *sites* dinâmicos com acesso a base de dados sem estar conectado a um servidor nem expor seu código na WEB até ele estar totalmente depurado ([MAR2000]).

Uma das primeiras escolhas que se precisa fazer é em que plataforma se deseja desenvolver aplicações ASP. O ASP acha-se hoje disponível em várias plataformas, mas foi disponibilizado primeiramente para IIS em *Windows NT Server* – que é o mais indicado, seguido de perto pelo *Windows NT Workstation* e *Windows 95/98 Personal Web Server* (PWS). Essas três escolhas são gratuitas, mas vinculadas a um sistema operacional Windows.

A entrada em baixo nível no desenvolvimento e publicação na Web é o *Personal Web Server* (PWS) disponível para Windows 95 e 98. O PWS foi desenvolvido tendo-se em mente a publicação pessoal, voltada para intranets corporativas de baixo volume. Os seguintes recursos principais acham-se disponíveis:

- a) serviço WWW;
- b) interface *Easy Web* para administração;
- c) suporte para *Microsoft Transaction Server*;
- d) suporte para *Microsoft Message Queue Server*;
- e) *Front Page Extensions* para facilitar a atualização de sites;
- f) suporte para *ActiveX Data Objects* (ADO) e *Remote Data Services* (RDS).

Como pode-se ver, tudo está disponível para o desenvolvimento de páginas da Web dinâmicas, orientadas a bancos de dados. É importante relatar que – como ele é um servidor da Web de baixo volume, a Microsoft o limitou a um máximo de dez usuários simultâneos.

O *Personal Web Server* para *Windows NT Workstation*, antes chamado *Peer Web Services* no *Windows NT Workstation*, trata-se de um servidor da Web para desenvolvimento de *sites* com recursos completos e pode ser usado para publicação pessoal em uma intranet corporativa. Esta plataforma também possui um limite de dez usuários, no entanto, vêm com mais recursos. Alguns dos recursos adicionais são:

- a) serviço de FTP;
- b) interface com *Optional Internet Service Manager* (ISM) para administração;
- c) suporte para autenticação;
- d) publicação com o uso de unidades locais e de rede.

A interface com o ISM é fornecida pelo *Microsoft Management Console* (MMC), que é o console de gerenciamento usado por todos os novos produtos *Microsoft BackOffice*, como por exemplo o *SQL Server 7.0*. No *Windows 2000*, ele será a ferramenta de gerenciamento padrão.

A força do ASP é percebida quando usada com o *Internet Information Server* no *Windows NT Server*. Obtem-se mais recursos que são extremamente úteis na construção de *sites* da Web modernos, altamente sofisticados. Esses recursos são:

- a) *Index Server* para indexação textual e pesquisa do conteúdo de um site;
- b) *Site Server Express* para a análise do conteúdo e do uso do site;
- c) serviço de SMTP para o envio de e-mail;
- d) serviço de NNTP para a hospedagem de *newsgroups* particulares;
- e) *Certificate Server* para a emissão de certificados que podem ser usados para a garantia de *sites* da Web.

O IIS adiciona recursos de pesquisa, recursos de e-mail e um nível adicional de segurança, bem como recursos de análise de *sites*. Devido ao seu poder e a sua versatilidade, todas as tarefas administrativas apresentadas são para o IIS e seu irmão, PWS, em *Windows NT Workstation*. O *Internet Service Manager* (ISM) é o nome da ferramenta de administração do IIS.

O *Microsoft Management Console* (MMC) é um console de gerenciamento comum que fornece uma estrutura consistente que pode ser usada por todos os programas de administração de rede como, por exemplo, IIS ou MTS.

O *Microsoft Transaction Server* (MTS) é um sistema de processamento de transações para a construção, administração e posicionamento de aplicações robustas para servidores da Internet. Uma transação é um conjunto de processos que devem ser executados, todos ou nenhum. Pôr exemplo, se você efetuar pagamentos com seu cartão de crédito, há dois processos: a) A soma paga por você é debitada de sua conta. b) A soma paga por você é creditada na conta do revendedor. Ambos os processos devem ser executados, ou nenhum deles; caso contrário, haveria um aumento ou diminuição mágica do dinheiro.

A versão do ASP que não é vinculada ao sistema operacional Windows é *Chili!ASP* da *Chili!Soft*. O *Chili!ASP* é uma versão para ambientes operacionais como o Unix. Ele suporta servidores da Web executados nas plataformas: *Netscape Enterprise*, *Netscape FastTrack* e *Lotus Domino*. Caso se utilize um servidor da Web como estes, o *Chili!ASP* é a solução mais indicada e oferece a mesma funcionalidade do ASP de um servidor da Web Microsoft. Encontra-se documentação e maiores informações na *home page* da *Chili!Soft* ([CHI2000]).

3.2.5 A HIERARQUIA DOS OBJETOS DAS PÁGINAS ASP

A utilização dos objetos nas páginas ASP permitem fazer um controle maior das informações enviadas do cliente para o servidor em nível de sessão e de aplicação. Cada objeto possui uma série de propriedades, métodos e eventos semelhantes aos de uma linguagem de programação visual, que facilitam a construção de uma aplicação usando ASP. Pode-se citar como objetos importantes utilizados na elaboração de páginas para a internet, o objeto *request*, *response*, *application*, *server* e *session*.

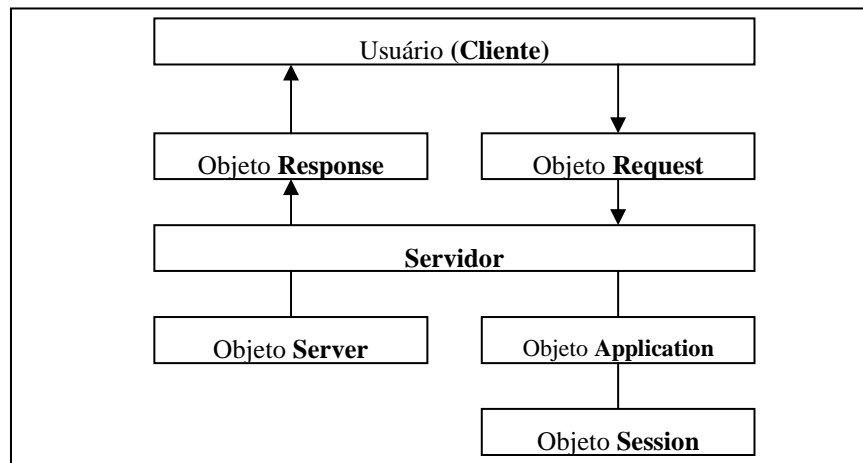
O Objeto *request* recebe informação do cliente para o servidor, ou seja, retorna os valores das requisições feitas pelo *browser* do cliente ao servidor durante uma requisição http. O objeto *response* envia informações do servidor para o cliente.

Os objetos *Application* e *Session* juntos gerenciam o armazenamento de informação no 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. O objeto *session* é semelhante ao objeto *application* no sentido de que ele também contém eventos dentro do arquivo “*global.asa*”. Entretanto, ao contrário do objeto *Application*, o objeto *session* armazena informações de uma sessão de usuário em particular. Esse objeto persiste para a sessão inteira

e proporciona assim uma solução elegante para o problema comum da persistência do estado. O servidor Web cria automaticamente um objeto *session* quando uma página é requisitada pelo usuário que ainda não tenha uma sessão. A sessão é destruída quando for abandonada ou quando expirar.

O Objeto *Server* é usado para gerenciar e criar objetos, permitindo o processamento de *scripts* e de acesso à base de dados. Este objeto fornece métodos e propriedades que permitem interagir com a máquina em que seu aplicativo está sendo executado, ou seja, o servidor da WEB. A figura 3.2.5.1 mostra como este processo funciona. ([MAR2000]).

Figura 3.2.5.1 : hierarquia dos objetos das páginas ASP



Fonte: [MAR2000]

3.2.6 ACESSO A BANCO DE DADOS COM ASP

O componente de acesso a dados incluído com ASP é o *Activex Data Objects* (ADO). O acesso à base de dados envolve grande parte das aplicações, pois quando se trata de aplicações comerciais se está tratando com informações que precisam ser armazenadas e recuperadas constantemente. Pode-se dizer que o ADO é uma solução genérica proposta pela Microsoft para acesso uniforme a dados armazenados em diferentes fontes de informação: Web, Cliente/Servidor, Textos, Isam entre outros. Ele é baseado na OLE DB o qual é um conjunto de interfaces do tipo *Component Object Model* (COM). O ADO é a interface entre a OLE DB e a sua aplicação. A utilização de ADO permite criar aplicações para acessar e manipular dados em um servidor de banco de dados via um provedor OLE DB. Ele pode ser utilizado em qualquer ferramenta que a suporte, tais como: Visual Basic, Access, ASP, Excel

entre outras. Naturalmente, o ADO possui características-chaves para se desenvolver aplicações Cliente/Servidor e para a Internet e, ao contrário do *Data Access Objects* (DAO) ou *Remote Data Objects* (RDO), não possui uma rígida estrutura hierárquica de objetos, pois seus objetos podem ser criados independentemente uns dos outros. O modelo, no entanto, é parecido com o modelo DAO e RDO, só que bem mais simples. A vantagem do modelo ADO é, segundo a Microsoft, a alta velocidade, a facilidade de uso e a baixa utilização de memória. Para usar ADO via ODBC você vai precisar de um *Data Source Name* (DSN) registrado. O ADO, através da tecnologia OLE DB, trabalha com provedores OLE DB, e o OLE DB é uma implementação COM para acesso de dados universal.([MAR2000]). O ADO fornece um modelo de objetos para acesso ao banco de dados, ou seja, funciona com qualquer ambiente de trabalho que possa lidar com os objetos *Activex/Com*.

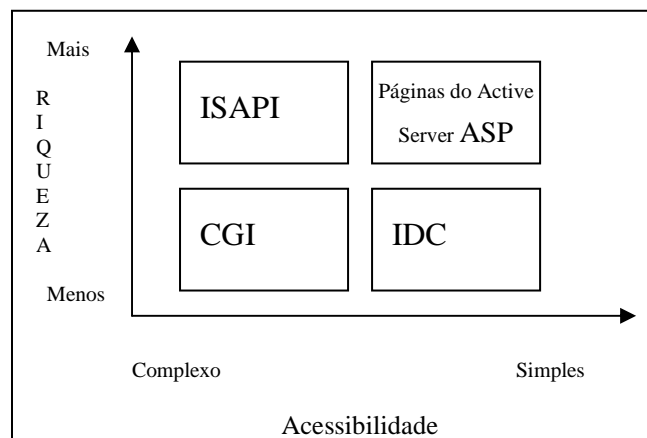
O *Activex* é um termo amplo que geralmente refere-se aos objetos do software que se comunicam entre si. Não é um produto e nem uma tecnologia. É mais do que um termo geral para muitas tecnologias relacionadas que normalmente têm relação com os objetos e a comunicação entre os processos. No caso do ADO os objetos *Activex* são programados para interagir com o OLE DB para o acesso ao banco de dados, pode-se dizer que são modelos de objetos que fornecem interfaces no OLE DB. De acordo com [McM1999] “Um componente *Activex* é o nível do meio entre uma arquitetura cliente/servidor que funcionam como outros tipos de componentes *Activex*, no sentido de que processam os dados e produzem objetos. Estes componentes são especificamente adequados para o acesso ao banco de dados cliente/servidor”.

Para entender uma arquitetura cliente/servidor é importante observar que isto é um conceito lógico. Os processos cliente enviam pedidos a um processo servidor, que responde com os resultados para esses pedidos. Como o nome indica, os processos servidores oferecem serviços aos seus clientes, normalmente por meio de um processamento específico que só eles podem fazer. O processo cliente, livre da complexidade e esforço adicional do processamento da transação, pode realizar outro trabalho útil. A interação entre os processos cliente e servidor é uma troca cooperativa, transacional, em que o cliente é ativo o servidor é reativo. Essa é a principal distinção entre cliente/servidor e outros paradigmas menos restritos ([McM1999]).

3.3 ASP COMPARADO A OUTRAS TECNOLOGIAS

Desde a primeira disponibilização do *Internet Information Server*, a Microsoft lançou três novas versões (a versão 2 veio com o primeiro lançamento do Windows NT 4). Com o lançamento do *Windows NT Option Pack*, o número da versão do IIS havia chegado a 4, e o IIS com todos os seus componentes adicionados havia feito um gigantesco progresso. O modo como você pode adicionar conteúdo dinâmico ao seu site da Web, no entanto, ainda é o mesmo da versão 3: aplicações CGI, extensões e filtros *Internet Server API (ISAPI)*, *Internet Database Connector (IDC)* e ASP. A figura 3.3.1 mostra as diferenças entre as quatro tecnologias em termos de acessibilidade e riqueza de conteúdo. ([WIL1999]).

figura 3.3.1 : diferenças entre tecnologias ISAPI,ASP,CGI,IDC



fonte: [WILL1999]

3.3.1 COMMON GATEWAY INTERFACE (CGI)

A discussão nesta seção é voltada para oferecer-lhe uma visão geral da CGI, seu uso e, principalmente, porque não usá-la ao posicionar seu *site* na Web do IIS. O CGI surgiu em servidores com o sistema operacional UNIX. São *scripts* que rodam no servidor e comandam o endereçamento de páginas com uma interface de comunicação comum, onde o usuário entra com os dados em um formulário e envia para um *script* em CGI que roda no servidor e retorna com uma resposta ao usuário.

O suporte para CGI é construído em quase todos os servidores da Web e não é mais um modo desejável de adicionar conteúdo dinâmico a um *site* da Web especialmente porque o IIS oferece modos melhores.

Em computadores UNIX, a maioria das aplicações CGI é escrita com o uso de *Practical Extraction and Report Language* (PERL). Um uso popular de Perl é a criação de formulários de e-mail para *feedback*. Outra linguagem usada para a criação de aplicações CGI é C, que permite aos desenvolvedores a criação de aplicações rápidas e ricas em recursos. As aplicações CGI em IIS podem ser aplicações WIN32 totalmente preparadas que permitem que você manipule diretamente todas as partes do sistema operacional. Grandes desvantagens dessas aplicações sob o ponto de vista do desenvolvedor são sua difícil manutenção e sua depuração igualmente difícil. Sob o ponto de vista do desempenho do servidor da Web, a maior desvantagem do uso do CGI é que para cada cliente que solicite uma página da Web com CGI, um novo processo precisa ser criado. Isso consome grande volume de recursos e de tempo de processamento, especialmente quando vários clientes solicitam uma página da Web em CGI. Se você precisar de acesso ao sistema operacional do seu *site* da Web e não desejar sacrificar o desempenho, poderá escolher o *Internet Server API*. ([WIL1999]).

3.3.2 INTERNET SERVER API

Para contornar uma importante desvantagem das aplicações em CGI (a necessidade de gerar um novo processo a cada solicitação do usuário) foi criada uma solução baseada em *dynamic link libraries* (DLL) chamado de *Internet Server API* (ISAPI).

As aplicações ISAPI são implementadas como DLL e são carregadas no espaço de processamento do *Internet Server*, o que permite execução mais rápida. Com o ISAPI podem ser criados dois tipos diferentes de aplicações:

- a) filtros: os filtros, que são transparentes para o cliente, são usados para monitorizar solicitações, esquemas de autenticação personalizados, conversão de dados durante o processo e muito mais. Os filtros são carregados no espaço de processamento do servidor da Web na inicialização do servidor e permanecem na memória até que o servidor seja desligado;
- b) extensões: muito parecido com o modo como as aplicações em CGI funcionam – solicitações de serviços para usuários específicos, as extensões ISAPI são usadas para processar formulários, recuperar dados de bancos de dados, executar lógica profissional e muito mais. Exatamente como os filtros ISAPI, as extensões são carregadas no espaço de processamento do servidor da Web. No entanto, isso

ocorre somente quando o primeiro usuário solicita um serviço usando a extensão, e não quando o servidor é iniciado. Você também pode usar extensões para ganhar acesso a níveis do sistema operacional. As DLLs ISAPI são carregadas no espaço de processamento do servidor da Web. A aparente desvantagem deste modo é que uma aplicação ISAPI escrita de modo deficiente pode ser desastrosa para todo o servidor da Web. Outra desvantagem é que você ainda está preso ao uso de uma linguagem de programação mais ou menos complexa, como por exemplo, C/C++ ou Delphi.

3.3.3 INTERNET DATABASE CONNECTOR

Um recurso especial do IIS desde a sua primeira versão é o *Internet Database Connector* (IDC). Você pode criar páginas preenchidas dinamicamente com dados provenientes de qualquer banco de dados acessível com ODBC. Você pode facilmente acessar dados de bancos de dados e apresentá-los ao cliente sem ter de criar uma aplicação CGI ou uma extensão ISAPI.

Para fazer o IDC funcionar, devem ser criados dois arquivos: um arquivo de consulta que contém as instruções em SQL para a recuperação de dados de um banco de dados ODBC e uma arquivo modelo que contém comandos que mesclam os resultados da consulta com o modelo HTML. Pode ser criado um *front end* de qualquer banco de dados que se deseje, usando esta tecnologia.

Uma nota final sobre o *Internet Database Connector*: embora a programação de banco de dados para consulta ou inserção de informações seja fácil, existe uma limitação à programação de bancos de dados somente – não é possível adicionar nenhuma lógica de servidor. ([WIL1999]).

3.3.4 TECNOLOGIAS ADICIONAIS PARA IIS

Com o *Internet Information Server* pode-se realizar várias tarefas, mas há mais duas tecnologias interessantes que podem ser utilizadas para a produção de aplicações com o uso do IIS: o *Allaire Coldfusion* e as aplicações do Visual Basic para Web:

- a) *Allaire ColdFusion*: Este é um servidor de aplicações da Web que permite construir páginas da Web altamente sofisticadas. Ele usa um ambiente de *script* de

servidor potente e abrangente, baseado em *tags*. Ele possui um *Integrated Development Environment* (IDE) e pode ser usado para criar aplicativos dimensionáveis. O IIS é usado como um servidor da Web. Você pode encontrar mais informações na *home page* da empresa, no endereço <http://www.allaire.com/>;

- b) aplicações do Visual Basic para a Web: Altamente integradas com o ASP, as aplicações do VB6 para a Web podem ser visualizadas como “páginas da Web encapsuladas”. Todo o código está contido em um componente do Visual Basic, permitindo que você acesse os mesmos recursos que acessaria em uma página do ASP, porém com as melhorias de desempenho do processamento em uma linguagem compilada. ([WIL1999]).

4. PROJETO DO SOFTWARE

Neste capítulo apresenta-se características e requisitos de software e hardware. Para esta aplicação desenvolveram-se várias telas, consultas e alguns relatórios, mas o enfoque principal do trabalho é o estudo do acesso à base de dados do sistema através de uma página na internet utilizando o ambiente ASP.

No levantamento de informações, fez-se o levantamento de todas as variáveis utilizadas no protótipo, identificando quais os procedimentos que são executados desde a entrada até o encerramento de um processo no escritório, os campos e as funções necessárias para o controle destes processos e de outras informações relevantes para o sistema de trabalho no escritório. Foram definidas também quais as páginas que serão disponibilizadas na internet para facilitar o acompanhamento dos processos pelos clientes, a formatação das páginas e dados que podem ser consultados.

A escolha das ferramentas utilizadas para o desenvolvimento do protótipo teve como principal motivo o conhecimento básico sobre algumas delas, o que facilitou a utilização para alcançar o objetivo proposto pelo trabalho. A modelagem de dados foi realizada com a utilização da ferramenta Powerdesigner.

Já a implementação das bases de dados (tabelas) foi realizada com o gerenciador de banco de dados Access97 e o desenvolvimento da aplicação, ou seja, toda a parte de cadastro, alteração, exclusão e consulta de dados utilizou-se a ferramenta de programação Visual Basic 6.0; e no desenvolvimento das páginas para Internet e digitação dos códigos ASP utilizou-se o Frontpage2000.

É importante frisar que no protótipo, não foi dada ênfase à ergonomia e à segurança do sistema e nem todas as opções exigidas para um controle eficaz das informações do escritório puderam ser incluídas devido à complexidade e a abrangência da proposta, mas foram desenvolvidas as funções mais importantes e necessárias para exemplificar com clareza o objetivo do trabalho.

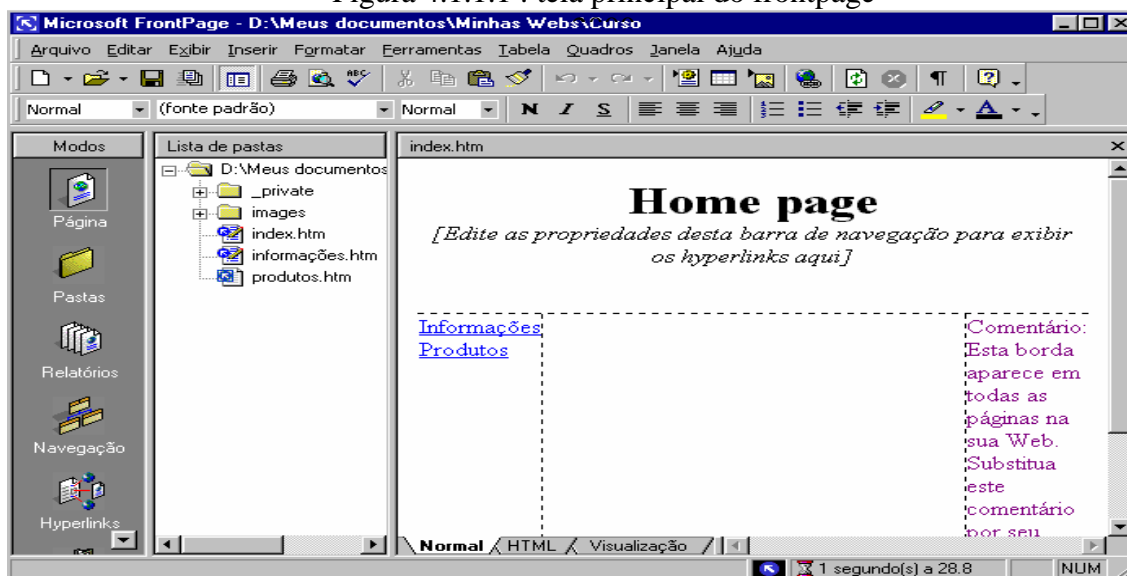
4.1 FERRAMENTAS UTILIZADAS

Para a implementação do protótipo foram utilizadas as ferramentas Frontpage 2000, Access97, Powerdesigner, Visual Basic 6.0. O Frontpage 2000 foi utilizado para a elaboração das páginas do protótipo que ficam disponíveis na internet para acessar os dados dos processos através de consultas. Inclui-se nesta fase a interface da página e a digitação dos códigos da linguagem ASP para acesso à banco de dados. O Powerdesigner foi empregado no desenvolvimento da modelagem dos dados do protótipo e o diagrama de contexto. A base de dados do protótipo, isto é, as tabelas e definição dos campos foram criadas no Access 97. No Visual Basic 6.0 desenvolveu-se o protótipo, ou seja, os formulários e a parte da programação para cadastramento, consulta, alteração e exclusão dos dados.

4.1.1 FRONTPAGE 2000

É um software que permite criar e editar páginas para Internet. Utilizou-se neste trabalho para a criação da página principal e a página que executa a consulta de processos. Com o FrontPage pode-se editar o HTML e os *scripts* existentes (incluindo ASP) e fazer uma integração com o Banco de Dados. É possível também atribuir várias formatações às páginas através de temas predefinidos, definir a estrutura e ordem de navegação entre as páginas, inserir *hiperlinks*, elementos gráficos e animações. A tela principal do FrontPage está apresentada na figura 4.1.1.1.

Figura 4.1.1.1 : tela principal do frontpage

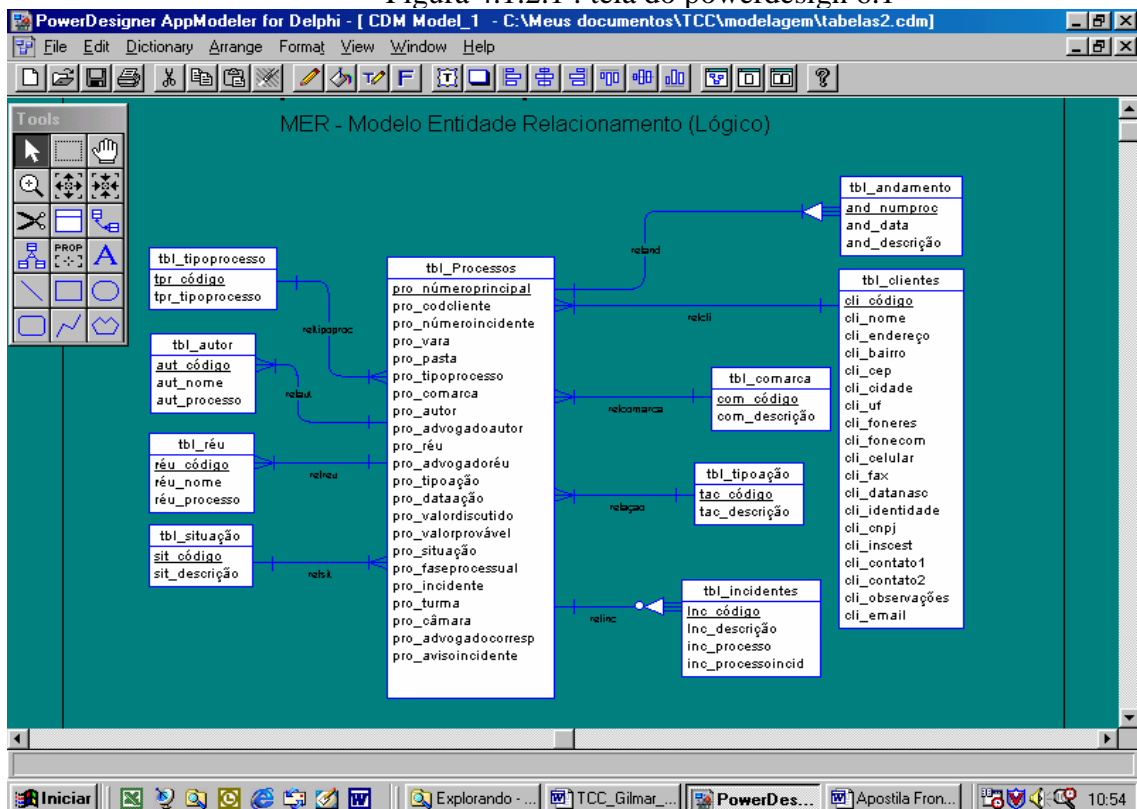


Fonte: [PRO2000]

4.1.2 POWERDESIGNER 6.1

É uma ferramenta Case para modelagem de banco de dados. A ferramenta permite desenvolver uma estrutura lógica global de um banco de dados que é independente de qualquer software ou estrutura de armazenamento de dados. Dá-se uma representação formal dos dados que são necessários para a implementação. A tela principal do PowerDesign está apresentada na figura 4.1.2.1.

Figura 4.1.2.1 : tela do powerdesign 6.1

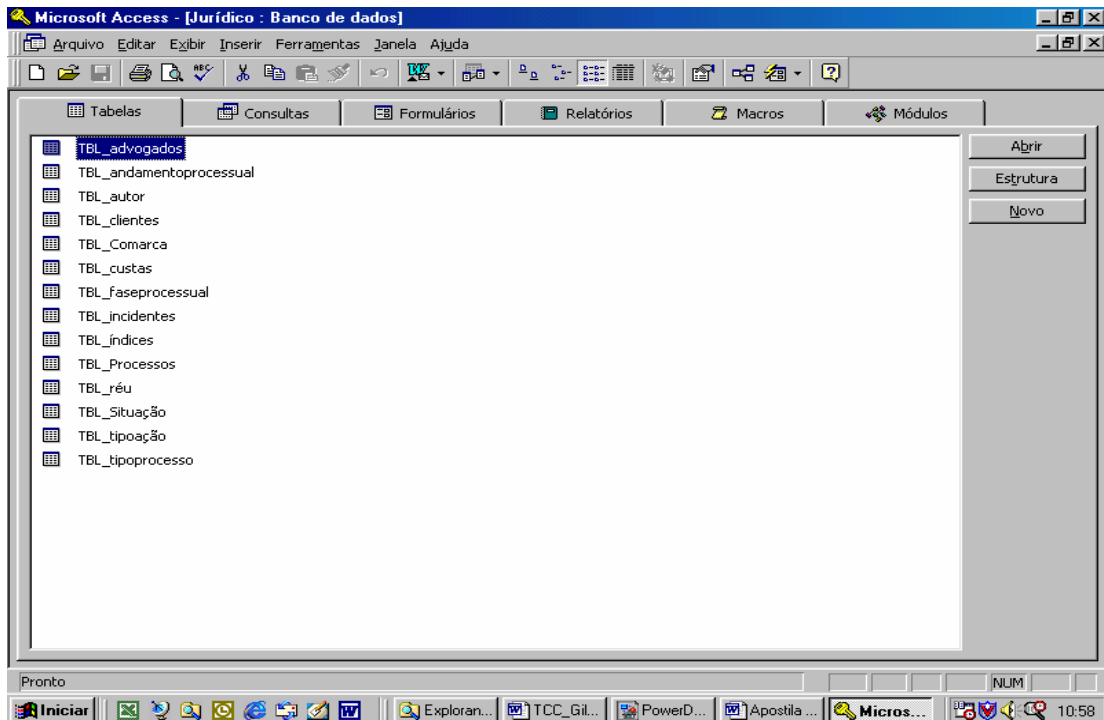


Fonte: [SYB2000]

4.1.3 ACCESS 97

É um aplicativo que permite fazer o gerenciamento de um banco de dados. Nele pode-se criar uma base de dados (tabelas), consultas, formulários para a entrada de dados, macros e desenvolver rotinas de programação em módulos. No Access foram elaboradas as tabelas do sistema, ou seja, a base de dados do software aplicativo, objeto deste trabalho. A tela principal do Access é apresentada na figura 4.1.3.1.

Figura 4.1.3.1 : tela principal do access 97

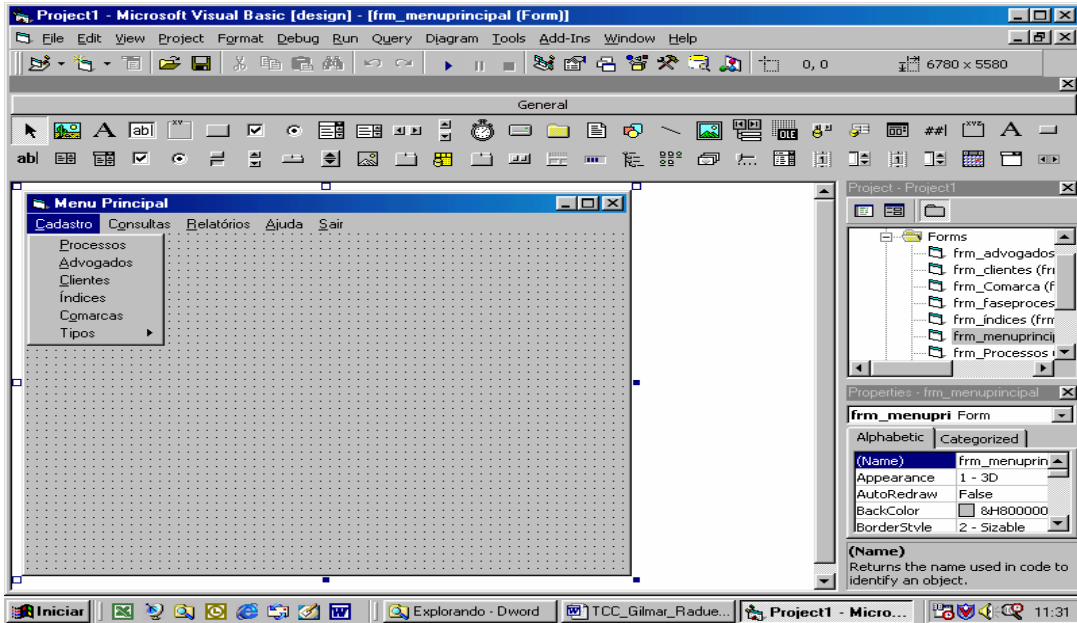


Fonte: [SAN1996]

4.1.4 VISUAL BASIC 6.0

O Visual Basic é um ambiente de programação para o sistema operacional Windows que se utilizou neste trabalho para desenvolver o protótipo do software aplicativo para o escritório de advocacia. Este ambiente forma o que se chama de *Rapid Application Development* (RAD) que permite desenvolver aplicativos visuais rapidamente com o uso de componentes gráficos pré-definidos. De acordo com [McK1997] o Visual Basic trabalha com eventos organizados em torno de um conjunto de objetos, que são estruturas combinando dados e rotinas em uma mesma entidade. A tela principal do Visual Basic é mostrada na figura 4.1.4.1.

Figura 4.1.4.1 : tela principal do visual basic 6.0

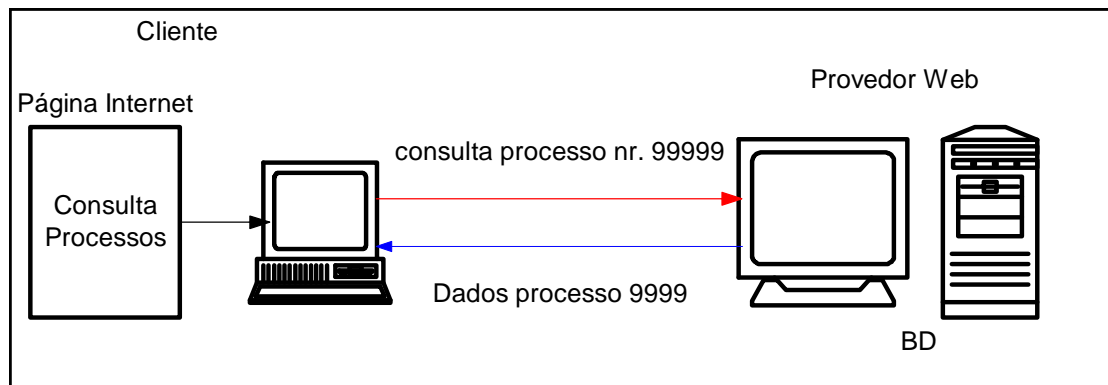


Fonte: [McM1999]

4.2 FUNCIONALIDADE

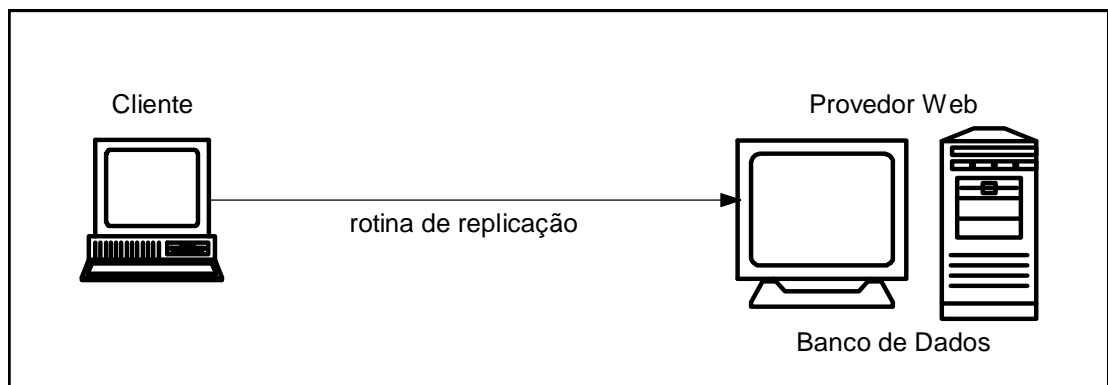
O Sistema proposto tem como objetivo permitir o acesso de informações referente ao andamento de processos através de uma página na internet. Para isto o cliente deverá acessar a página na internet com o endereço <http://www.index.com.br>, digitar o número do processo que deseja consultar e clicar na opção consultar andamento. A página ASP será processada pelo provedor que fará o acesso a base de dados e devolverá a resposta ao cliente conforme está apresentado na figura 4.2.1

figura 4.2.1 : consulta de processos



A atualização da base de dados no provedor é efetuada através de uma rotina de replicação, que fará a comparação dos dados que estão no provedor e os que foram alterados no sistema do escritório. Ressaltando que o sistema não está on-line com a base de dados no provedor, a replicação pode ser feita diariamente conectando-se ao provedor para executar a rotina somente. A figura 4.2.2 mostra como o processo acontece.

figura 4.2.2 : rotina de replicação

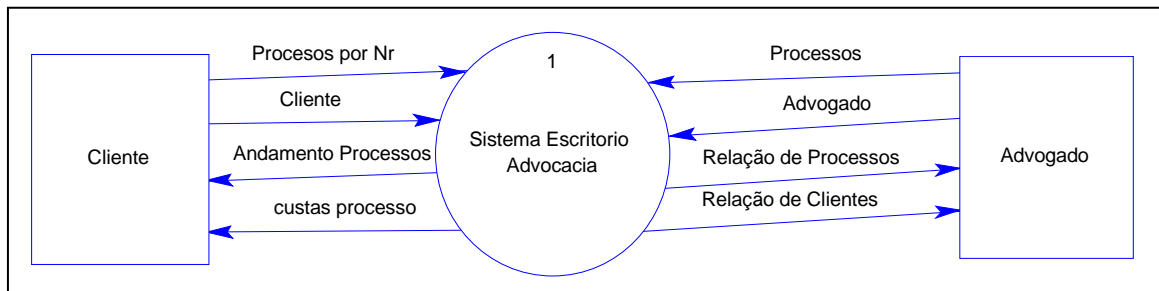


4.3 ESPECIFICAÇÃO

A metodologia utilizada no desenvolvimento da especificação do sistema foi a análise estruturada. A análise estruturada tem papel importante no sucesso do desenvolvimento estruturado de sistemas. A Análise estruturada se propõe a fornecer um meio de comunicação comum entre o usuário e o analista, um meio de lapidar o sistema do usuário no início do projeto e uma especificação manutenível, que seja não-redundante, não-ambígua e completa. O resultado é que a análise estruturada aumenta as chances de automatizar o sistema da maneira que o usuário desejar. ([KEL1990]).

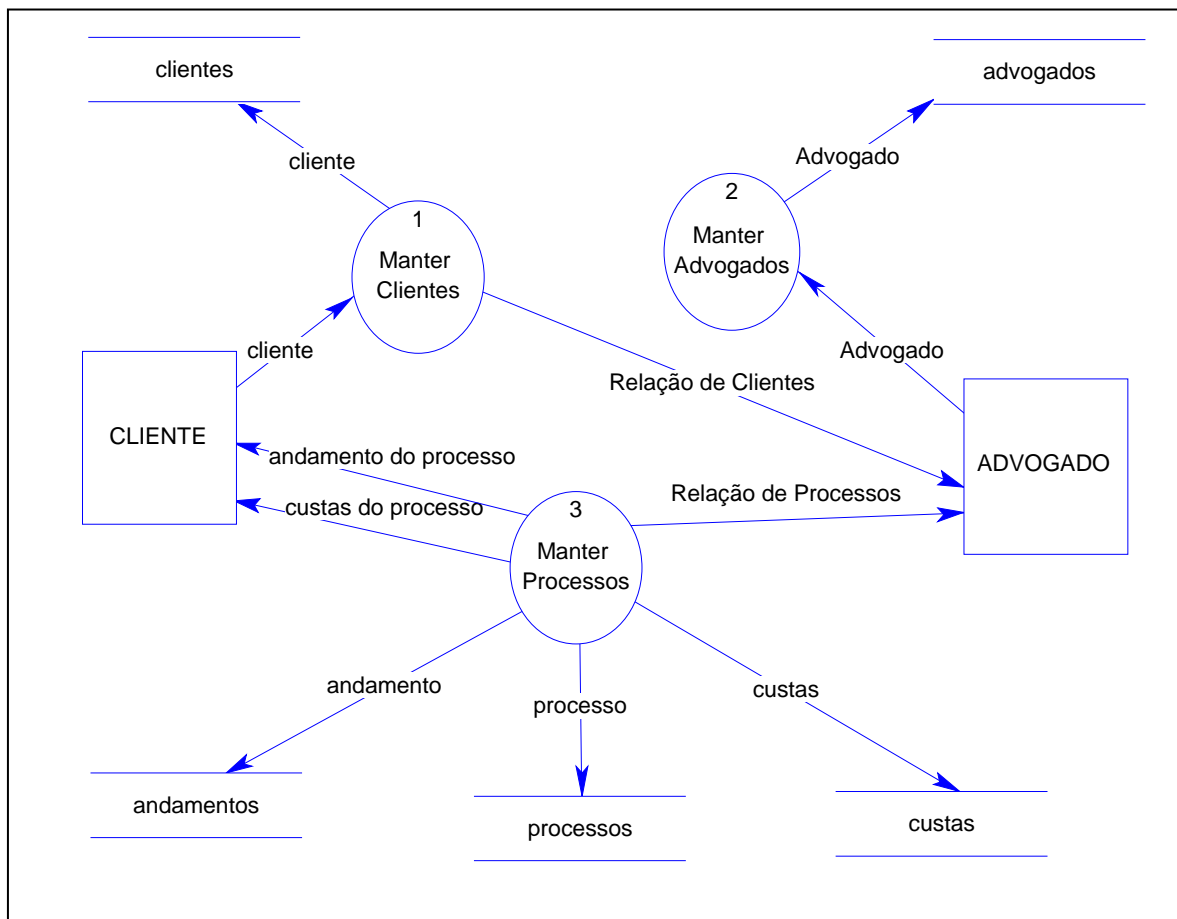
Com o auxílio da ferramenta case Powerdesign foram criados o diagrama de contexto e o modelo entidade e relacionamento (MER). No diagrama de contexto apresentado na figura 4.3.1 são mostradas as entidades externas envolvidas no sistema do escritório de advocacia. A entidade cliente está relacionada com o sistema devido as consultas realizadas; a entidade advogado, colocada como a entidade responsável pelo controle e administração do sistema, relaciona-se através dos processos, que são cadastrados e conferidos através de uma relação de processos por cliente. As outras entidades são a Comarca, Autor e Réu, todos os processos pertencem a uma comarca e podem possuir diversos Autores e Réus.

figura 4.3.1 : diagrama de contexto



Na figura 4.3.2 é mostrado o Diagrama de Fluxo de Dados (DFD) de nível zero de acordo com a técnica de Yourdon/Demarco. ([KEL1990]).

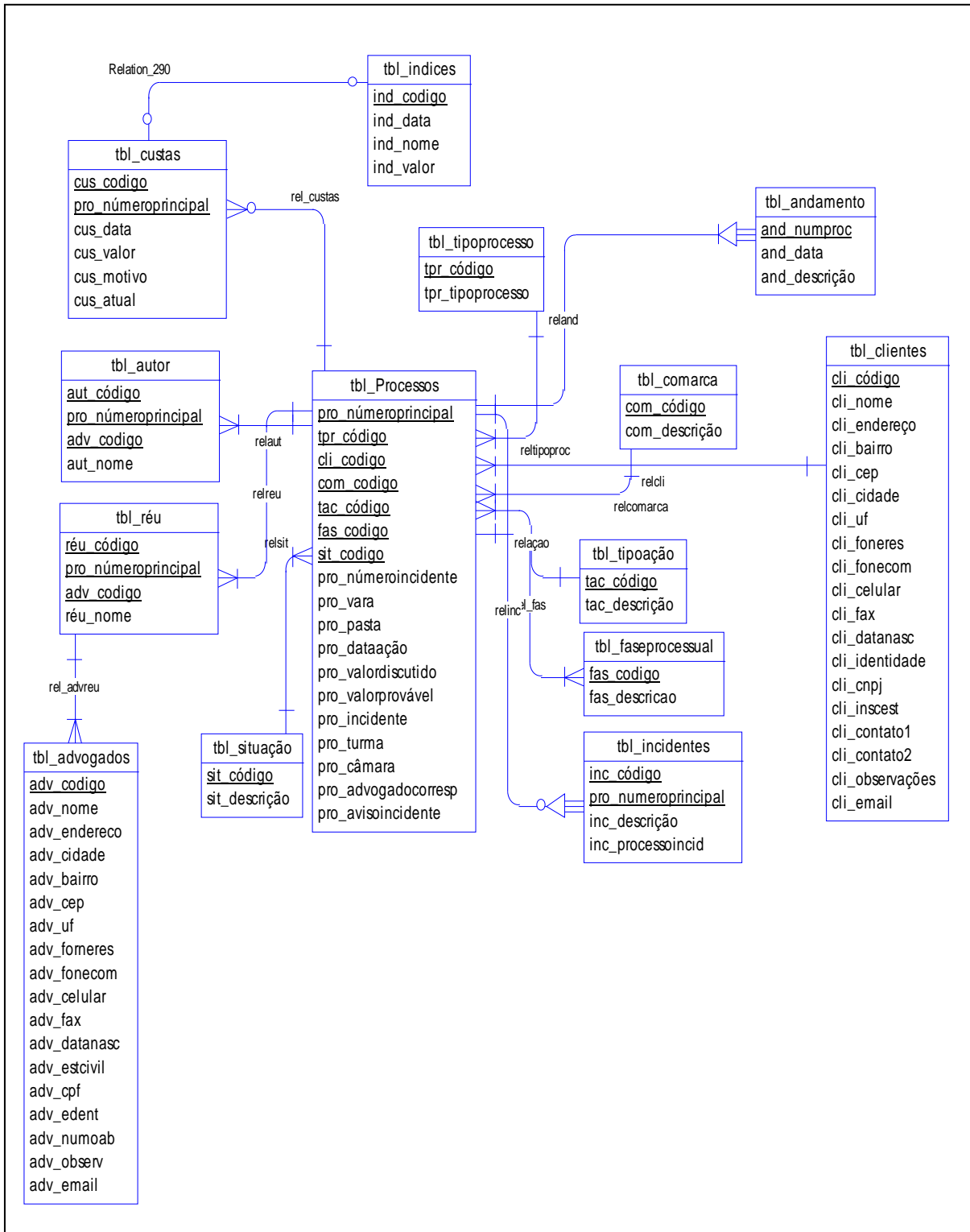
figura 4.3.2 digrama de fluxo de dados – nível zero



A figura 4.3.3 apresenta o modelo de entidade e relacionamento (MER), onde são apresentadas as tabelas do sistema e seus relacionamentos. Para explicar melhor o que a figura 4.3.2 mostra, tem-se o seguinte:

- a) `tbl_processos`: é a tabela principal do sistema, armazena todos os dados do processo e tem ligação com todas as demais tabelas;
- b) `tbl_andamento`: armazena dados sobre o andamento dos processos;
- c) `tbl_clientes`: armazena dados dos clientes que podem ser pessoas físicas ou jurídicas;
- d) `tbl_advogados`: armazena informações sobre os advogados envolvidos nos processos;
- e) `tbl_autor`: contém informações sobre os autores de um processo; cada processo pode ter um ou mais autores;
- f) `tbl_réu`: contém informações sobre os réus de um processo; da mesma forma que os autores, o processo pode ter um ou mais réus;
- g) `tbl_tipoprocesso`: mantém a descrição do tipo do processo;
- h) `tbl_situação`: mantém a descrição das situações em que o processo pode se encontrar;
- i) `tbl_tipoação`: mantém a descrição do tipo de ação a que se refere o processo;
- j) `tbl_incidentes`: cada processo pode ter vários processos incidentes, esta tabela mantém a descrição da incidência;
- k) `tbl_comarca`: mantém a descrição das comarcas;
- l) `tbl_custas`: mantém armazenado as custas dos processos;
- m) `tbl_índices`: mantém um cadastro de índices econômicos.

figura 4.3.2. : modelo entidade relacionamento (MER)



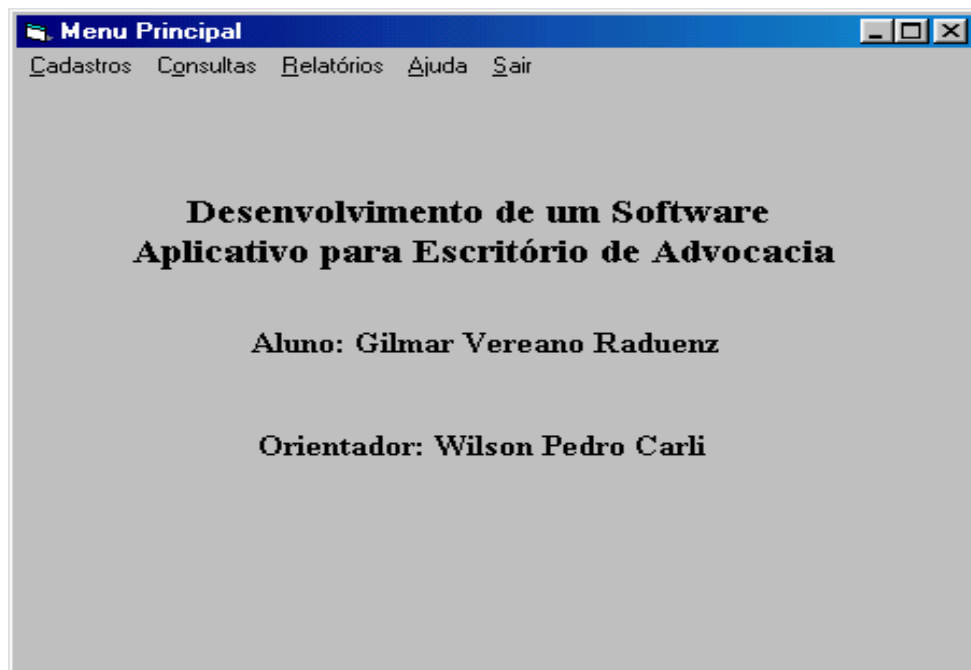
4.4 APRESENTAÇÃO DO PROTÓTIPO

A implementação do protótipo foi organizada em cinco módulos principais (Cadastros, Consultas, Relatórios, Ajuda, Sair) e cada módulo com suas respectivas funções. Serão apresentadas as funções mais importantes de cada módulo, deixando de lado detalhes de implementação.

4.4.1 MENU PRINCIPAL

A figura 4.4.1.1 mostra a tela do menu principal onde o usuário tem acesso a todas as demais opções do protótipo. Na opção de Cadastros o usuário cadastra, altera e exclui, processos, advogados, clientes, índices econômicos, comarcas, tipos de processo, ação e fase. No menu de Consultas podem ser feitas consultas de processos, clientes, advogados, consulta personalizada, relação de clientes, processos, tipos de ação e de processo. No menu Relatórios são emitidos os relatórios do sistema, por exemplo, relação de clientes, processos e outros. O Ajuda fornece informações pertinentes ao sistema e a opção Sair encerra o sistema.

figura 4.4.1.1 : tela do menu principal



4.4.2 MANTER CADASTRO DE PROCESSOS

Esta opção permite ao usuário fazer o cadastro, a alteração e exclusão de processos; pode-se dizer que esta é a tela principal do aplicativo onde são inseridos todos os detalhes de cada processo. O formulário está dividido em quatro pastas conforme aparece na figura 4.4.2.1 (Processos, Autores/Réus, Custas, Andamento) que serão vistas em seguida.

Figura 4.4.2.1 : tela de cadastro de processos

A imagem mostra a janela de software 'Cadastro de Processos' com uma barra de menu contendo quatro abas: 'Processos' (selecionada), 'Autores/Réus', 'Custas' e 'Andamento'. O formulário principal contém os seguintes campos:

- Número principal: 1111
- Cliente: Gilmar Vereano Raduenz
- Vara: 2 | Pasta: 2
- Tipo do processo: Trabalhista
- Comarca: Blumenau - SC
- Tipo da ação: Mandado de segurança | Data da ação: 12/12/1999
- Valor discutido: 10 | Valor provável: 15
- Situação: 1 | Fase processual: Primeira Instância
- Turma: 2 | Câmara: 2
- Advogado corresp.: Gilmar Vereano Raduenz
- Incidente: (campo vazio)
- Número do incidente: 0

Na base da janela, há uma barra de navegação com setas e quatro botões: 'Incluir', 'Alterar', 'Excluir' e 'Fechar'.

4.4.3 MANTER CADASTRO DE AUTORES E RÉUS

Esta função faz parte do cadastro de processos onde são incluídos os autores e réus correspondentes a cada processo. Cada processo pode conter um ou mais autores ou réus que são representados por advogados de acordo com a figura 4.4.3.1.

4.4.4 MANTER CADASTRO DE CUSTAS DE PROCESSO

Esta função faz parte do cadastro de informações de processos. Nesta tela são lançadas as custas de cada processo conforme a figura 4.4.4.1.

Figura 4.4.3.1 : tela de cadastro de autores/réus

Cadastro de Processos

Processos Autores/Réus Custas Andamento

Número do Processo: 1111

Autor(es):

Nome
▶ Carlos Eduardo da silva
Maria de Souza Neto
João de Oliveira
*

Réu(s):

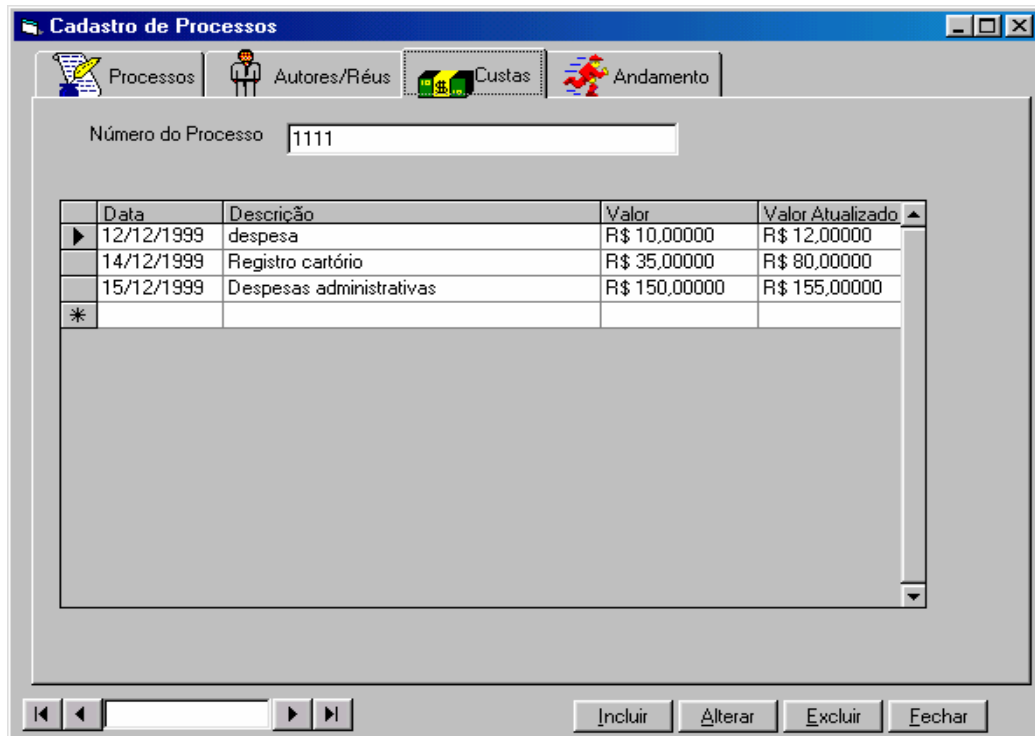
Nome
▶ José Soares
Jorge Silveira
Marlene Mattos
*

Advogado do autor: Gilmar Vereano Raduenz

Advogado do réu: Gilmar Vereano Raduenz

Incluir Alterar Excluir Fechar

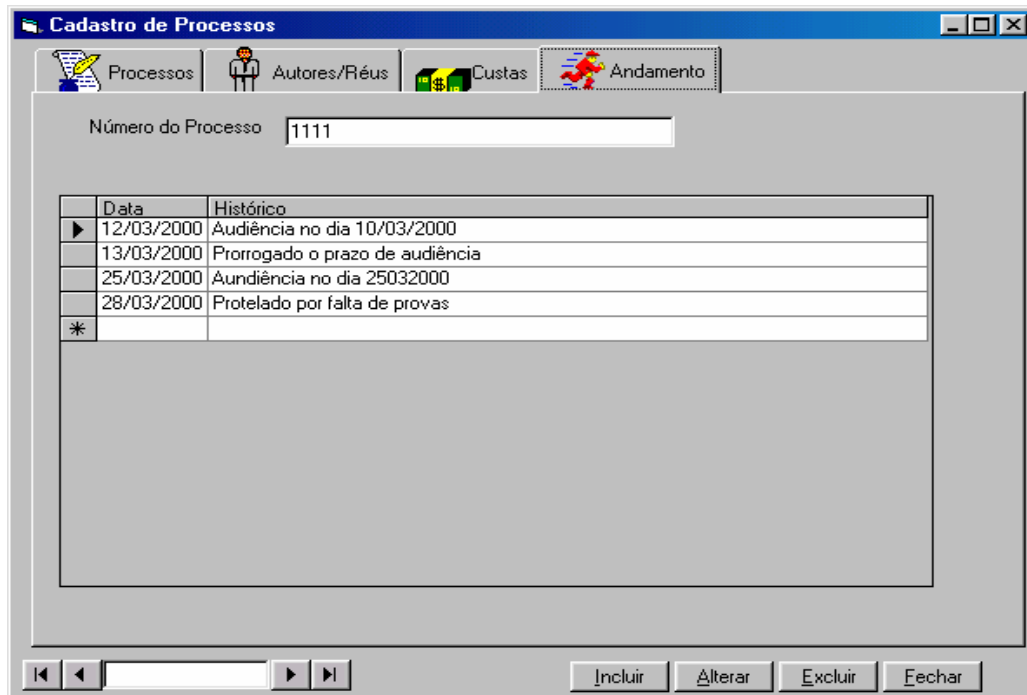
Figura 4.4.4.1 : tela de custas de processo



4.4.5 MANTER CADASTRO DE ANDAMENTO DE PROCESSOS

Esta função também faz parte do cadastro de informações sobre os processos. A finalidade é incluir detalhes sobre o andamento de cada processo conforme a figura 4.4.5.1.

Figura 4.4.5.1 tela de andamento de processos



4.4.6 MANTER CADASTRO DE CLIENTES

Esta função tem como objetivo cadastrar, alterar e excluir clientes do escritório de advocacia, que pode ser pessoas físicas ou jurídicas. O exemplo é apresentado na figura 4.4.6.1.

4.4.7 MANTER CADASTRO DE ADVOGADOS

Nesta opção o usuário pode cadastrar, alterar e excluir informações sobre os advogados. O exemplo da tela é apresentado na figura 4.4.7.1.

4.4.8 MANTER CADASTRO DE COMARCAS

Esta função tem como objetivo permitir a inclusão, alteração e exclusão de comarcas, representado na figura 4.4.8.1

figura 4.4.6.1 : tela de cadastro de clientes

Cadastro de Clientes

Código: 1

Nome: Gilmar Vereano Raduenz

Endereço: Rua São Bento, 478

Bairro: Quintino Bocaiúva

Cep: 89120000

Cidade: Timbó

UF: SC

Fone residencial: 014473820578

Fone comercial: 014473820578

Celular: 99814729

Fax:

Email: gvr@inf.furb.rct-sc.br

Tipo: Física Jurídica

Data nascimento: 07/03/71

Estado civil: Solteiro

CPF: 65426900100

Identidade:

Inscrição Estadual:

CNPJ:

Contato1:

Contato2:

Observações: Cliente bom

Incluir Alterar Excluir Fechar

figura 4.4.7.1 : tela de cadastro de advogados

Cadastro de Advogados

Código: 1

Nome: Gilmar Vereano Raduenz

Endereço: Exp. Jose da Silva

Bairro: Centro

Cep: 89120-000

Cidade: Timbó

UF: sc

Fone residencial:

Fone comercial:

Celular:

Fax:

Email: gvr@inf.furb.rct-sc.br

Data nascimento:

Estado civil:

CPF:

Identidade:

Numero OAB:

Observações:

Incluir Alterar Excluir Fechar

figura 4.4.8.1 : tela cadastro de comarcas

Cadastro de Comarcas

Código: 2

Descrição: Joinville - SC

Incluir Alterar Excluir Fechar

4.4.9 MANTER CADASTRO DE TIPOS DE PROCESSO

Esta função tem como objetivo permitir a inclusão, alteração e exclusão de tipos de processo, representado na figura 4.4.9.1

figura 4.4.9.1 : tela cadastro de tipos de processo

A janela 'Cadastro de Tipos de Processo' possui um campo 'Código' com o valor '1' e um campo 'Tipo de processo' com o valor 'Trabalhista'. Abaixo dos campos há uma barra de navegação com setas e quatro botões: 'Incluir', 'Alterar', 'Excluir' e 'Fechar'.

4.4.10 MANTER CADASTRO DE TIPOS DE AÇÃO

Esta função tem como objetivo permitir a inclusão, alteração e exclusão de tipos de ação, representado na figura 4.4.10.1.

figura 4.4.10.1 : tela cadastro de tipos de ação

A janela 'Cadastro do Tipo da Ação' possui um campo 'Código' com o valor '1' e um campo 'Descrição' com o valor 'Mandado de segurança'. Abaixo dos campos há uma barra de navegação com setas e quatro botões: 'Incluir', 'Alterar', 'Excluir' e 'Fechar'.

4.4.11 MANTER CADASTRO TIPOS DE FASE PROCESSUAL

Esta função tem como objetivo permitir a inclusão, alteração e exclusão de tipos de fase processual, representado na figura 4.4.11.1.

figura 4.4.11.1 : tela cadastro de tipos de fase processual

A janela 'Cadastro Fase Processual' possui um campo 'Código' com o valor '1' e um campo 'Descrição' com o valor 'Primeira Instância'. Abaixo dos campos há uma barra de navegação com setas e quatro botões: 'Incluir', 'Alterar', 'Excluir' e 'Fechar'.

figura 4.4.13.1 : exemplo relatório de clientes

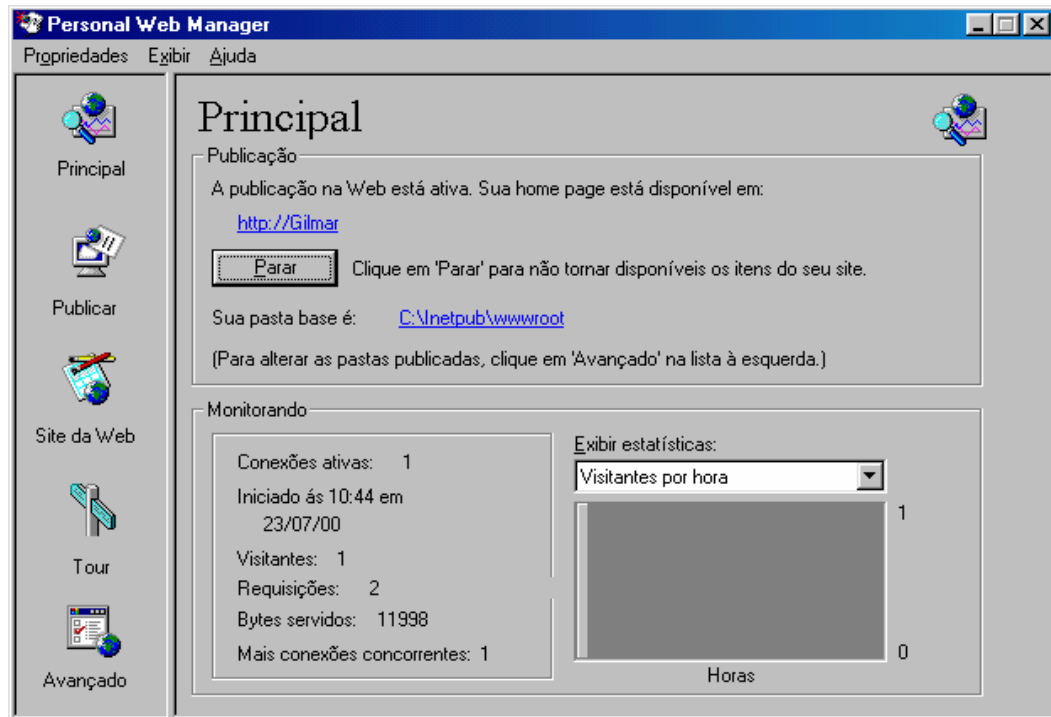
Relatório de Clientes		13/06/2000		
<u>Nome</u>	<u>Cidade</u>	<u>UF</u>	<u>Fone</u>	<u>Email</u>
Geovani da Silva Xavier		SC		
EEEE	EEEEEEEE	sc	54354	4342
Gilmar Vereano Raduenz	Timbó	SC	014473820578	gvm@inf.fub.rct-sc.br
Joyce de Almeida	mmmmmmwwwww	SC		
Vanessa Carla		SC		

figura 4.4.13.2 : exemplo relação de processos

Relação de Processos		13/06/2000				
<u>Nome</u>	<u>Número Processo</u>	<u>Tipo</u>	<u>Tipo da Ação</u>	<u>Nr.incidente</u>	<u>Data da ação</u>	<u>Vlr.discutido</u>
Gilmar Vereano Raduenz						
Gilmar Vereano Raduenz	1111	2	2	0	12/12/1999	R\$10,00
Gilmar Vereano Raduenz	1112	12.122	0	454545		R\$0,00
Gilmar Vereano Raduenz						R\$10,00

4.5 PÁGINAS NA INTERNET

A seguir são mostradas as páginas do escritório de advocacia que podem ser acessadas pelos clientes através da internet. Todas as páginas desenvolvidas para o protótipo possuem a extensão “.asp”. A interface das páginas e o código em ASP foram desenvolvidas no Frontpage e para testar a execução do código ASP utilizou-se o *Personal Web Server* (PWS) em uma máquina local. O PWS é um servidor web que deve ser instalado na máquina local caso se necessite testar as aplicações em ASP. Na instalação padrão do PWS é criado um diretório *Inetpub\wwwroot* o qual é identificado como diretório de trabalho com direito à execução. É dentro deste diretório que foram salvas as páginas criadas em ASP para que elas pudessem ser testadas antes de ser efetivamente disponibilizadas na internet. Pode-se também criar novos subdiretórios, mas é importante lembrar que todos eles devem estar configurados “com direito à execução”; Esta configuração é efetuada no gerenciador do PWS. Um exemplo da tela principal do *Personal Web Server* é apresentada na figura 4.5.1.

figura 4.5.1 tela principal do *Personal Web Server*

4.5.1 PÁGINA PRINCIPAL NA INTERNET

Através desta página principal na internet conforme a figura 4.5.1.1, os clientes terão acesso a informações sobre o escritório de advocacia e também poderão fazer consultas sobre o andamento de seus processos.

4.5.2 PÁGINA CONSULTA DE PROCESSOS

Através desta página os clientes poderão fazer consultas aos processos digitando o número do processo. A página fará o acesso a base de dados e mostrará informações sobre o andamento e as custas do processo. A página é apresentada na figura 4.5.2.1. O código em ASP que é executado no momento em que o usuário clicar no botão consultar está demonstrado na figura 4.5.2.2.

figura 4.5.1.1 : página principal do escritório na internet

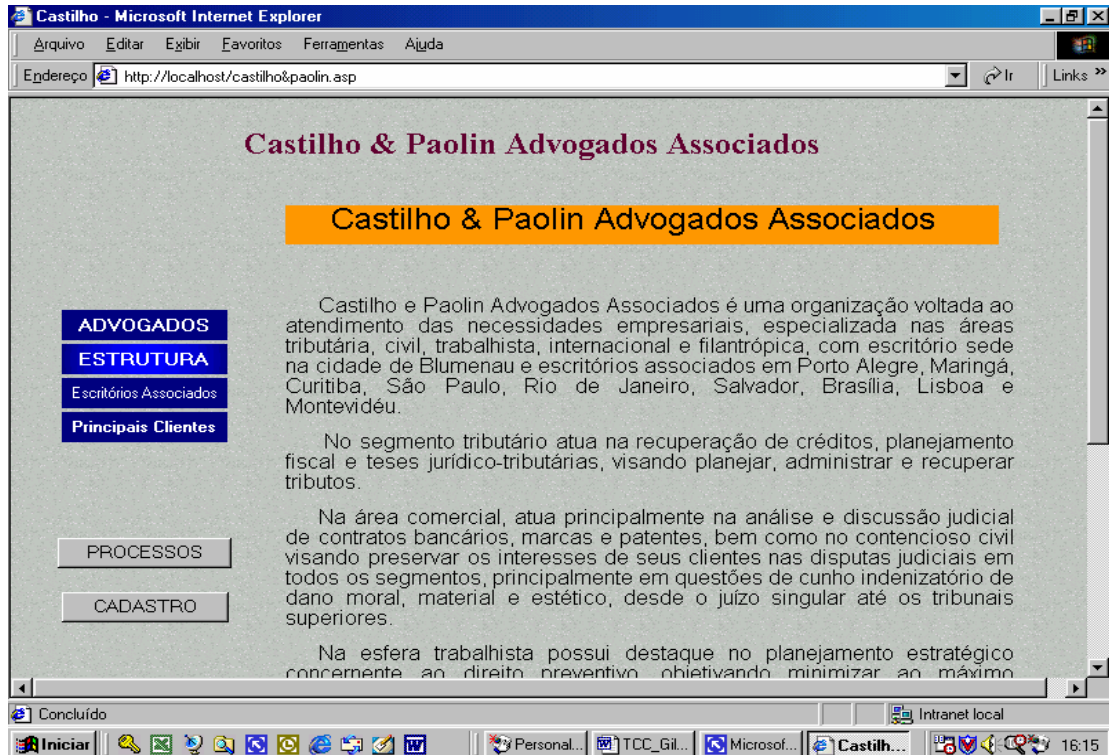


figura 4.5.2.1 : página consulta de processos

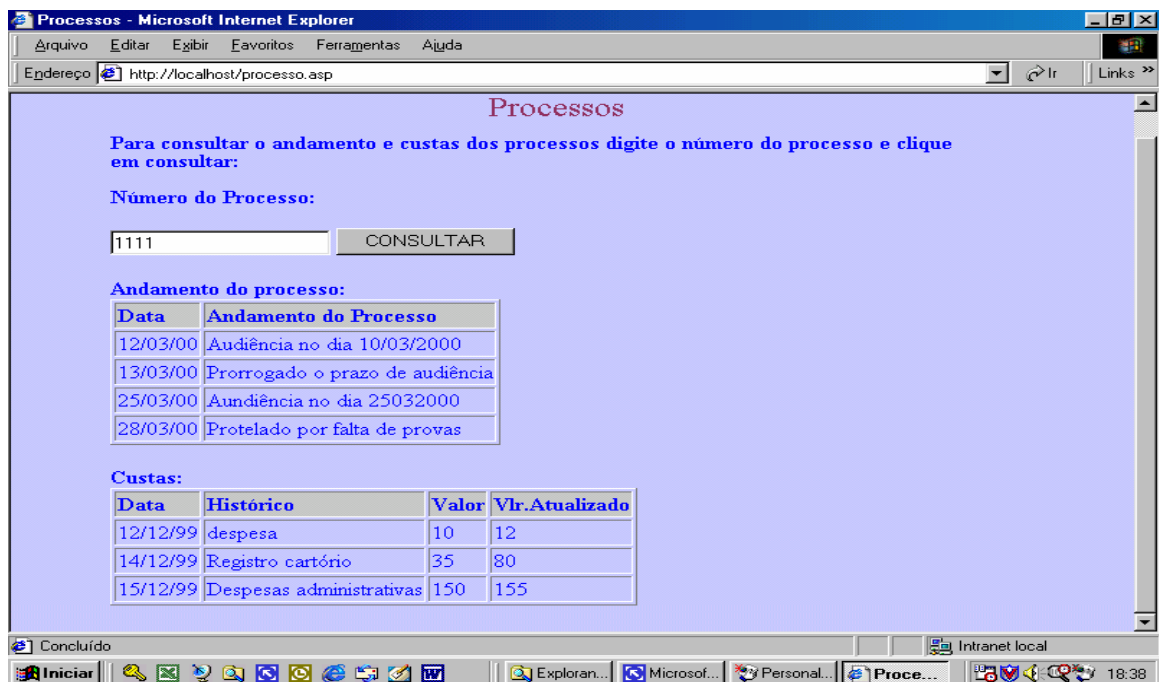


Figura 4.5.2.2 código ASP da página consulta de processos

```

<%
PUBLIC NRO Subrotina que zera o campo número do processo na página
NRO = REQUEST.Form("NRO")
IF nro="" then
    NRO=CINT(0)
else
    NRO = NRO
end if

SUB CABECALHOA Subrotina para gerar o cabeçalho do Andamento de Processos
Response.Write "<table border>" & _
"<tr bgcolor=#cccccc><td>    <b>Data</b></td><td><b>Andamento do Processo</b></td></tr>"
END SUB

SUB CABECALHOC Subrotina para gerar o cabeçalho Custas de processo
Response.Write "<table border>" & _
"<tr bgcolor=#cccccc><td>    <b>Data</b></td><td><b>Histórico</b></td><td><b>Valor    R$    </b></td><td><b>Vlr.Atual    R$</b></td></tr>"
END SUB

Subrotina que acessa a tabela Andamentoprocessual no banco de dados e retorna os dados do processos selecionado.
SUB ANDAMENTO
set rs = server.createobject("adodb.recordset")
sql = "select * from TBL_andamentoprocessual where and_numproc = " & NRO & ""
rs.open sql, "DSN=Jurídico;"
CABECALHOA
do until rs.eof
    response.write("<tr>")
    response.write("<td>")
    response.write(rs.fields("and_data"))
    response.write("</td>")
    response.write("<td>")
    response.write(rs.fields("and_descrição"))
    response.write("</td>")
    rs.movenext
    response.write("</tr>")
loop
response.Write "</table>"
END SUB

```

4.5.3 PÁGINA CADASTRO DE CLIENTES

Através desta página o advogado poderá fazer o cadastramento das informações mais importantes de novos clientes. O objetivo desta página é permitir à distância, incluir novos clientes no banco de dados para um posterior contato. Os dados são digitados na página e incluídos no arquivo de dados do provedor. Após a execução da rotina de replicação os dados estarão disponíveis também no arquivo local do escritório de advocacia, e poderão ser alterados, consultados através do sistema. A página mostra também a funcionalidade do ASP na inclusão de dados através da internet. A figura 4.5.3.1 mostra um exemplo desta página. Um exemplo do código ASP desta página está demonstrado na figura 4.5.3.2. Pode-se verificar a mistura que existe de comandos em HTML e ASP.

figura 4.5.3.1 : página cadastro cliente

The screenshot shows a web browser window titled "Cadastro Informações de Cliente - Microsoft Internet Explorer". The address bar shows "http://localhost/Cadastrarcil.asp". The main content area has a light blue background and is titled "Cadastrar Cliente". It contains a form with the following fields and values:

- Nome: Carlos de Andrade
- Endereço: Rua Florianópolis, 104
- Bairro: Centro
- Cep: 89120000
- Cidade: Timbó
- UF: SC
- Fone Residencial: 04733821022
- Fone Comercial: 04733821212
- Email: cda@inf.furb-rct.com.br

There are two radio buttons for "Física" (selected) and "Jurídica". Below the form are "Incluir" and "Cancelar" buttons. At the bottom left, it says "Incluindo registro..." and at the bottom center, there is a link "Voltar ao Menu Principal". The taskbar at the bottom shows several open applications and the system clock at 11:36.

Figura 4.5.3.2 : página cadastro cliente do sistema

The screenshot shows a desktop application window titled "Cadastro de Clientes". It contains a form with the following fields and values:

- Código: 25
- Nome: Carlos de Andrade
- Endereço: Rua Florianópolis, 104
- Bairro: Centro
- Cep: 89120000
- Cidade: Timbó
- UF: SC
- Fone residencial: 04733821022
- Fone comercial: 04733821212
- Celular: (empty)
- Fax: (empty)
- Email: cda@inf.furb-rct.com.br
- Tipo: Física (selected), Jurídica
- Data nascimento: (empty)
- Estado civil: (empty)
- CPF: (empty)
- Identidade: (empty)
- Inscrição Estadual: (empty)
- CNPJ: (empty)
- Contato1: (empty)
- Contato2: (empty)
- Observações: (empty text area)

At the bottom of the window are navigation buttons: "Incluir", "Alterar", "Excluir", and "Fechar".

Figura 4.5.3.2 código ASP da página cadastrar cliente

```

<HTML>
<HEAD>
<TITLE> Cadastro Informações de Cliente </TITLE>
</HEAD>
<BODY BGCOLOR="#CCCCFF" TEXT="#0000FF">
Subrotina que acessa a tabela TBL_Clientes do banco de dados e inclui os dados digitados na página.
<%
SUB ABRETABELA
DIM cn
DIM rs
Const adopenkeyset = 1
Const adlockOptimistic = 3
set cn = server.createobject("adodb.connection")
set rs = server.createobject("adodb.recordset")
cn.open "Jurídico"
rs.open "Select * from tbl_clientes", cn, adopenkeyset, adlockOptimistic
rs.addnew
    rs.fields("cli_nome") = trim(request.form("nome"))
    rs.fields("cli_endereço") = trim(request.form("ender"))
    rs.fields("cli_bairro") = trim(request.form("bairro"))
    rs.fields("cli_cep") = trim(request.form("cep"))
    rs.fields("cli_cidade") = trim(request.form("cidade"))
    rs.fields("cli_uf") = trim(request.form("uf"))
    rs.fields("cli_foneres") = trim(request.form("foneres"))
    rs.fields("cli_fonecom") = trim(request.form("fonecom"))
    rs.fields("cli_email") = trim(request.form("email"))
    if request.form("tipof") = CHECKED then
        rs.fields("cli_tipo") = 1
    else
        rs.fields("cli_tipo") = 0
    end if
rs.update
rs.close
response.write(request.form("nome"))
response.write("<p> Gravado com sucesso!")
END SUB
%>

<center>
<H2>Cadastrar Cliente</H2>
</center>
<br>
<TABLE ALIGN="CENTER">
<TR>
<TD>Nome: </TD>
<TD>
<FORM METHOD="Post" ACTION="Cadastrarli.ASP">
<INPUT TYPE="text" NAME="nome" SIZE="50" MAXLENGTH="50" >
</TD>
</TR>
<TR>
<TD colspan="2">
* A parte dos campos em HTML devem ser definidos nesta parte.
<TR ALIGN="RIGHT">
<TD><Br>
</TD>
<TD>
<p align="center">
<INPUT TYPE="submit" NAME="Incluir" VALUE=" Incluir "><INPUT TYPE="RESET" NAME=" Cancela "
VALUE="Cancelar" >
</p>
</TD>
</TR>
</TABLE>
<%if trim(request.form("Incluir")) <> "Incluir" then
response.write("Incluindo registro...")
else
ABRETABELA
end if %>
</Form>
<center>
<A HREF="Castilho&Paolin.asp" TARGET="_self" >Voltar ao Menu Principal</A>
</center>
</BODY>
</HTML>

```

4.5.4 PÁGINA ESCRITÓRIOS ASSOCIADOS

Através desta página os clientes podem visualizar informações referentes ao escritório de advocacia. No caso específico os escritórios que estão associados a este escritório de advocacia.

Figura 4.5.4.1 : página escritórios associados



5. CONCLUSÃO

Com o estudo feito para a realização deste trabalho, percebeu-se um grande potencial do ASP como técnica de acesso a banco de dados através da Internet. Especificamente as páginas dinâmicas utilizadas na internet para a manipulação de dados têm contribuído muito neste aspecto, permitindo que se faça uma filtragem dos dados para uma melhor análise e compreensão. Isto tem ajudado não só na consulta de informações na Internet pelos clientes, mas também tem facilitado o acesso e a disponibilização de informações em Intranets auxiliando na tomada de decisões.

No atual mercado econômico é importante que os escritórios de advocacia mantenham sua atenção voltada para a Internet para que permaneçam competitivos e possam atender com eficiência as necessidades de seus clientes. Sabe-se que a economia digital está mudando o mundo e que a Internet será em breve o principal ambiente para negócios de todo tipo, de compra e venda de produtos, serviços, ações, cultura, lazer, etc..., e a empresa que não estiver conectada no novo espaço criado pela rede poderá perder grande fatia de mercado pela concorrência eletrônica. Neste sentido os escritórios de advocacia devem buscar uma ferramenta que auxilie no gerenciamento das informações e a integração à Internet, para que se possa recuperar dados atualizados com rapidez o que proporciona uma vantagem competitiva sobre os concorrentes.

A contribuição básica deste trabalho é a implementação do protótipo e o desenvolvimento das páginas dinâmicas, *Active Server Pages (ASP)*, para acesso ao banco de dados integrado a um software aplicativo para escritório de advocacia; que no caso proporciona uma maior comodidade ao cliente na consulta de informações sobre os processos, e também para o escritório, no sentido de organizar melhor os dados e agilizar a tramitação interna de documentos. O protótipo estará em funcionamento no escritório de advocacia Castilho & Paolin Advogados Associados suprimindo a falta de um software aplicativo para gerenciar todos os processos internos.

E finalmente, a contribuição maior é o conhecimento das ferramentas utilizadas no desenvolvimento do trabalho, em especial o Visual Basic na versão 6.0, que possui uma excelente técnica de acesso a dados (ADO) que se relaciona muito bem com ASP para a construção de aplicativos voltados para a Internet.

5.1 DIFICULDADES

A maior dificuldade encontrada durante a elaboração deste trabalho foi o estudo das plataformas necessárias para o desenvolvimento do objetivo proposto. No caso do HTML, não possuía nenhum conhecimento prévio e no caso do ASP, apesar de ter encontrado vários artigos com exemplos na Internet e apenas três livros nas livrarias da região.

5.2 SUGESTÕES PARA TRABALHOS FUTUROS

Quanto a sugestões para pesquisas futuras, especificamente para continuação deste trabalho, sugere-se incluir novas opções de consultas e relatórios e também o estudo de outras tecnologias de acesso a dados como PHP, *ColdFusion*, *Chili!ASP*, para outros sistemas operacionais que não sejam vinculados ao Windows.

6. REFERÊNCIA BIBLIOGRÁFICA

- [BAR2000] BARRETO, Alexandre. **Desenvolvendo websites**. 2000. Endereço eletrônico : <http://www.educativanet.com.br/asp>
- [CHI2000] CHILI!SOFT. **Chili!ASP**. 2000. Endereço eletrônico : <http://www.chilisoft.com>
- [KEL1990] KELLER, Robert; **Análise Estruturada na Prática**. São Paulo : McGraw-Hill, 1990.
- [MAR2000] MARINHO, Paulo. **ASP e banco de dados: uma introdução**. 2000. Endereço eletrônico : <http://www.webmasterguide.com.br>
- [McM1999] McMANUS, Jeffrey P. **Visual basic 6 banco de dados**. Rio de Janeiro : Ciência Moderna , 1999.
- [McK1997] McKELVY, Mike; MARTINSEN, Ronald; WEBB, Jeff. **Usando visual basic 5**. Rio de Janeiro : Campus, 1997.
- [NAS2000] NASCIMENTO, Humberto. **História da internet**. 2000. Endereço eletrônico: <http://wmg.ewebcity.com/homefull.asp>
- [PRO2000] PROWAY INFORMÁTICA LTDA. **Curso básico de frontpage 2000**. Blumenau, 2000.
- [SAN1996] SANTOS JÚNIOR, Mozart J.F. **Access 97 for windows**. Goiânia : Terra, 1996.
- [SAV1997] SAVOLA, Tom; WESTENBROEK, Alan; HECK, Joseph. **Usando HTML**. Rio de Janeiro : Campus, 1997.
- [SYB2000] SYBASE, Inc. **Powerdesign**. 2000. Endereço eletrônico : <http://www.sybase.com./products/powerdesign>
- [THE1991] THEODORO JÚNIOR, Humberto. **Curso de direito processual civil**. Rio de Janeiro : Forense, 1991.

[WIL1999] WILLE, Christoph; KOLLER, Christian. **Aprenda em 24 horas active server pages**. Rio de Janeiro : Campus, 1999.