

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

**SISTEMA DE CONSULTA PROCESSUAL
BASEADO NA TECNOLOGIA WAP**

TRABALHO DE ESTÁGIO SUPERVISIONADO SUBMETIDO À UNIVERSIDADE
REGIONAL DE BLUMENAU PARA A OBTENÇÃO DOS CRÉDITOS NA
DISCIPLINA COM NOME EQUIVALENTE NO CURSO DE CIÊNCIAS DA
COMPUTAÇÃO — BACHARELADO

JAISON LUIZ FUSINATO

BLUMENAU, JUNHO/2003

2003/1-37

SISTEMA DE CONSULTA PROCESSUAL BASEADO NA TECNOLOGIA WAP

JAISON LUIZ FUSINATO

ESTE TRABALHO DE ESTÁGIO SUPERVISIONADO, FOI JULGADO ADEQUADO
PARA OBTENÇÃO DOS CRÉDITOS NA DISCIPLINA DE ESTÁGIO
SUPERVISIONADO OBRIGATÓRIA PARA OBTENÇÃO DO TÍTULO DE:

BACHAREL EM CIÊNCIAS DA COMPUTAÇÃO

Prof. Mauro Marcelo Mattos — Supervisor na FURB

Dionei Starke — Orientador na Empresa

Prof. José Roque Voltolini da Silva — Coordenador na
FURB do Estágio Supervisionado

BANCA EXAMINADORA

Prof. Mauro Marcelo Mattos

Prof. Francisco Adell Péricas

Prof. Oscar Dalfovo

DEDICATÓRIA

Aos meus pais, Valdemar e Osmarina Fusinato, que foram novamente fonte de incentivo e apoio, desta vez para a conclusão deste trabalho, que sem dúvida restará como mais um marco alcançado na minha vida.

AGRADECIMENTOS

A Deus, a quem sempre posso recorrer e também confiar.

Ao Professor Mauro Marcelo Mattos, que se dispôs de pronto, independente de ser a figura de supervisor, ajudar em tudo o que fosse necessário para a realização deste trabalho, orientando, sanando minhas dúvidas e fornecendo todo o tipo de apoio, como de fato o fez.

Ao amigo Dionei Starke, que sem dúvida superou seu papel de orientador, uma vez que de todas as maneiras possíveis, buscou suprir as necessidades tecnológicas e burocráticas para a realização deste trabalho.

Aos meus familiares e amigos, por acreditarem na minha capacidade e apostarem no meu potencial.

À Diretoria de Informática do Tribunal de Justiça do Estado de Santa Catarina, por todo o apoio e atenção dispensados.

A todos que de alguma maneira contribuíram para a conclusão deste trabalho.

“O único lugar onde o sucesso vem antes do trabalho é no dicionário.”

Albert Einstein

RESUMO

O presente trabalho descreve o projeto de um sistema para consulta de informações processuais do Tribunal de Justiça do Estado de Santa Catarina, baseado na tecnologia *Wireless Application Protocol* (WAP), que é uma das recentes tecnologias para acesso à internet destinada especificamente a dispositivos sem fio, associada à tecnologia *Active Server Pages* (ASP) para acesso a banco de dados e criação de páginas dinâmicas na linguagem *Wireless Markup Language* (WML). O trabalho contém, além da implementação e especificação do referido sistema, aspectos teóricos e práticos relativos às tecnologias mencionadas.

ABSTRACT

The present work describes the project of a system to consult procedural information about Court of Justice of Santa Catarina State, based on the technology Wireless Application Protocol (WAP), that is one of the recent technologies for access internet destined specifically to without wire devices, associated with the technology Active Server Pages (ASP) for access the database and creation of dynamic pages in Wireless Markup Language (WML). The work contains, beyond the implementation and specification of the related system, theoretical and practical aspects to the mentioned technologies.

LISTA DE FIGURAS

Figura 1 – Comparação entre os modelos de rede WWW e WAP.....	17
Figura 2 – Pilha da arquitetura WAP.....	19
Figura 3 – Analogia entre os protocolos internet e WAP.....	20
Figura 4 – Tramitação processual.....	29
Figura 5 – Diagrama de contexto	31
Figura 6 – DFD consultas.asp.....	32
Figura 7 – DFD partes.asp.....	32
Figura 8 – DFD advogados.asp	33
Figura 9 – DFD processos.asp.....	33
Figura 10 – DER lógico.....	34
Figura 11 – Infra-estrutura associada à implementação	36
Figura 12 – EasyPad Waptor	37
Figura 13 – M3Gate.....	38
Figura 14 – Personal Web Server	39
Figura 15 – Internet Explorer 6.0	40
Figura 16 – Macro-fluxo do sistema	41
Figura 17 – Fluxograma do sistema	43
Figura 18 – Tela inicial e de aviso.....	44
Figura 19 – Tela de consulta.....	45
Figura 20 – Preenchimento dos campos para pesquisa	46
Figura 21 – Resultado da consulta.....	47
Figura 22 – Obtenção das informações processuais.....	48

SUMÁRIO

1	INTRODUÇÃO	11
1.1	OBJETIVOS DO TRABALHO	13
1.2	ESTRUTURA DO TRABALHO	13
2	TECNOLOGIA WAP	15
2.1	MODELO DE REDE WAP	16
2.2	ARQUITETURA WAP	18
2.2.1	<i>WIRELESS APPLICATION ENVIRONMENT (WAE)</i>	20
2.2.2	<i>WIRELESS SESSION PROTOCOL (WSP)</i>	21
2.2.3	<i>WIRELESS TRANSACTION PROTOCOL (WTP)</i>	21
2.2.4	<i>WIRELESS TRANSPORT LAYER SECURITY (WTLS)</i>	23
2.2.5	<i>WIRELESS DATAGRAM PROTOCOL (WDP)</i>	23
2.2.6	<i>BEARERS (PORTADORAS)</i>	24
2.3	LINGUAGENS PARA PROGRAMAÇÃO WAP.....	24
2.3.1	<i>WIRELESS MARKUP LANGUAGE (WML)</i>	24
2.3.2	<i>WIRELESS MARKUP LANGUAGE SCRIPT (WMLSCRIPT)</i>	26
2.4	USO DA TECNOLOGIA ASP PARA GERAÇÃO DE PÁGINAS WML.....	27
3	DESENVOLVIMENTO DO SISTEMA	28
3.1	O SISTEMA JUDICIÁRIO.....	28
3.2	CARACTERÍSTICAS DO PROBLEMA	30
3.3	ESPECIFICAÇÃO	30
3.3.1	DIAGRAMA DE CONTEXTO.....	30
3.3.2	DIAGRAMAS DE FLUXO DE DADOS (FD)	31
3.3.3	DIAGRAMA ENTIDADE-RELACIONAMENTO (DER)	34

3.4 IMPLEMENTAÇÃO	35
3.4.1 TÉCNICAS E FERRAMENTAS UTILIZADAS.....	36
3.4.1.1 EASYPAD WAPTOR.....	36
3.4.1.2 M3GATE	37
3.4.1.3 PERSONAL WEB SERVER	38
3.4.1.4 SUN ONE ACTIVE SERVER PAGES 4.0	39
3.4.1.5 INTERNET EXPLORER 6.0	40
3.4.2 FUNCIONAMENTO DO SISTEMA.....	41
3.4.3 APRESENTAÇÃO DO SISTEMA	44
4 CONCLUSÕES	49
4.1 EXTENSÕES	50
REFERÊNCIAS	51
APÊNDICE	54

1 INTRODUÇÃO

Com a instituição do código de divisão e organização judiciária (SANTA CATARINA, 1979) em 1979, bem como do regimento interno (SANTA CATARINA, 1982), que vigorou a partir de 25 de outubro de 1982, isto em adaptação a lei orgânica da magistratura nacional, o Tribunal de Justiça em conjunto com a Corregedoria-Geral, definiram e implementaram diversos métodos para propiciar uma melhor administração da justiça. Dentre eles, pode-se citar a divisão do estado em comarca e distritos.

Com o passar dos anos, o aumento e o acúmulo de processos, isto é, o acréscimo processual, foi um fator relevante para se repensar os métodos adotados. Diversos controles tais como as cargas, as movimentações e as localizações processuais eram registradas manualmente em livros, contribuindo ainda mais para a morosidade da justiça, levando-se em conta todos os passos e trâmites estabelecidos no código de processo civil e código de processo penal.

Em 07 de dezembro de 1995, com o advento da informática, no intuito de automatizar as funções administrativas e cartorárias do Tribunal de Justiça e de suas comarcas, fora criado o Centro de Pesquisa e Aplicação de Informática – CEPAI (SANTA CATARINA, 1995). A principal ação do CEPAI ocorreu em 1996, quando da implantação do Sistema de Automação do Judiciário (SAJ), sistema desenvolvido pela empresa Softplan. O sistema possui duas versões: SAJ-SG, para uso no Tribunal de Justiça e SAJ-PG, para uso nas comarcas.

No início, tal providência foi aclamada, apesar da lentidão no cadastramento dos processos e recursos que já estavam tramitando. A adaptação dos serventuários da justiça e dos advogados militantes nas comarcas foi um processo natural. Foram automatizados também, sistemas de consulta processual, instalados nos cartórios e nas salas da Ordem dos Advogados do Brasil (OAB) de cada comarca.

Em pouco tempo, todos os processos já haviam sido cadastrados, as dúvidas inerentes ao sistema haviam sido sanadas e o tempo de tramitação dos processos diminuía. Com isso, vários efeitos colaterais começaram a surgir. Os processos eram movimentados rapidamente, o que forçava os advogados a virem constantemente ao fórum para consultá-los, uma vez que

certas informações processuais não podem ser prestadas por telefone (conforme dispõe o código de normas da Corregedoria-Geral de Justiça - TJSC, 2002, art. 453-457).

Tal situação foi logo detectada pelo CEPAI que não hesitou em providenciar soluções para o problema: a primeira iniciativa tomada foi o desenvolvimento de uma página internet de consulta processual, que busca as informações em cada comarca, acessando o servidor de dados do qual se desejava obter as informações processuais (TJSC, 2003a); a segunda, foi o desenvolvimento de um sistema chamado TJSCPush (TJSC, 2002) que envia, por e-mail ou por mensagens no celular, informações dos processos de interesse do usuário cadastrado neste serviço.

Ambos os serviços supra mencionados não suprem totalmente as necessidades dos interessados, uma vez que o primeiro serviço fica restrito aos usuários da internet, e que o segundo, apesar de interagir mais com o usuário, somente alerta àquele sobre alguma modificação na localização ou movimentação processual, porém não traz o conteúdo desta, submetendo o usuário a ter que buscar informações mais precisas no *site*. Outro problema deste segundo serviço, é que muitas vezes o usuário é avisado sobre as alterações processuais com atraso, o que dificulta o acompanhamento do processo.

A Comissão de Gestão de Informatização – CGINFO (SANTA CATARINA, 2000), foi um órgão que foi instituído em 11 de fevereiro de 2000, com o objetivo de auxiliar na manutenção¹ do parque informatizado do TJSC, ouvindo os interessados neste procedimento, sendo, portanto, incumbido de várias funções, dentre elas a de definir as políticas e diretrizes institucionais para a informatização, planejar a implantação de ferramentas para promover a gestão da qualidade e acompanhar o processo de informatização. Todos os problemas acima relatados são de conhecimento da CGINFO.

Apesar dos vários problemas existentes, há de salientar que em pesquisa recentemente divulgada pelo *site* Consultor Jurídico (TJSC, 2003b), o serviço de divulgação do TJSC foi considerado o 2º melhor do Brasil, comparado aos demais tribunais de justiça, obtendo 72,7% de aprovação entre os internautas que participaram da enquete naquele *site*. O primeiro lugar

¹ De manter; aquilo que se mantém ou sustenta; sustento; manutenção; mantimento.

ficou com o trabalho de divulgação do Tribunal de Justiça do Estado do Rio Grande do Sul, com 77,9% de aprovação.

Com base nas informações acima, concebendo as tecnologias já desenvolvidas pelo Tribunal de Justiça, este trabalho consiste no desenvolvimento de um sistema para consulta processual baseado na tecnologia *Wireless Application Protocol* (WAP), para auxiliar na necessidade dos interessados quanto à obtenção das informações processuais.

Para o desenvolvimento do referido sistema associou-se à tecnologia WAP a tecnologia *Active Server Pages* (ASP) para acesso ao banco de dados ORACLE utilizado no TJSC e para a criação de páginas dinâmicas na linguagem *Wireless Markup Language* (WML).

1.1 OBJETIVOS DO TRABALHO

O principal objetivo deste trabalho de estágio supervisionado é o desenvolvimento de um sistema de consulta processual baseado na tecnologia WAP, para que os interessados pelas informações processuais, estas oriundas dos autos que tramitam no Tribunal de Justiça e nas suas comarcas, possam acessá-las quando da sua necessidade, em qualquer lugar, desde que disponham tão somente de um celular ou qualquer outro dispositivo dotado de um micro-navegador.

Os objetivos específicos do trabalho são:

- a) identificação dos requisitos de tecnologia;
- b) identificação das informações que devem ser disponibilizadas;
- c) identificação dos usuários;
- d) liberação das informações processuais por três tipos de consulta: pelo nome da parte, pelo nome do advogado ou pelo número do processo.

1.2 ESTRUTURA DO TRABALHO

O texto a seguir está organizado da seguinte forma.

O capítulo 2 fala sobre a tecnologia WAP, apresentando sua estrutura e aspectos principais. Há também aspectos sobre as linguagens WML e WMLScript, utilizadas para a construção de páginas WAP.

O capítulo 3 trata do desenvolvimento do sistema, envolvendo os aspectos de especificação e implementação, fornecendo informações sobre a sua funcionalidade e sobre as ferramentas e técnicas utilizadas. Além disso, referencia aspectos que envolvem a tramitação processual.

Por fim, o capítulo 4 faz uma análise conclusiva indicando as dificuldades encontradas e sugestões para trabalhos futuros envolvendo a tecnologia WAP.

2 TECNOLOGIA WAP

Segundo Demétrio (2000), WAP, sigla de *Wireless Application Protocol* (Protocolo de Aplicação sem fio), é uma especificação para o conjunto de protocolos de comunicação com o intuito de normalizar a forma como os dispositivos sem fio acessam a internet.

Rocha (2003) afirma que o protocolo WAP permite o acesso móvel a informações e serviços disponíveis na internet, com uma navegação eficaz. As informações são comprimidas para que possam circular mais rapidamente pela rede, apesar da lenta conexão oferecida pelos aparelhos dotados de micro-navegador para esta tecnologia (telefones celulares, *paggers*, *Palmtops*, *smartphones*, etc.), algo entre 14,4 e 56,6 Kbps.

O grande êxito desta tecnologia tem sido o fato de colocar vários serviços, antes disponíveis somente em um micro de mesa, no bolso do usuário, fazendo-o dispor de uma série de possibilidades (DEMÉTRIO, 2000).

WAP Brasil (2003), assinala que outro aspecto a ser considerado é que o WAP pode ser utilizado em qualquer sistema operacional, incluindo plataformas populares de *handhelds* como Palm OS, Windows CE. Isso ocorre porque o protocolo é independente de padrões de comunicação, em vez de ser baseado nas plataformas existentes. Dessa forma, qualquer plataforma que é capaz de implementar os padrões de comunicação é compatível com WAP.

A especificação do protocolo WAP foi o resultado do esforço de várias empresas que deu origem ao WAP Fórum, órgão criado em junho de 1997. Inicialmente este órgão foi formado pelas indústrias Ericsson, Nokia, Motorola e Phone.com, porém em meados de 2000, já contava com mais de 500 membros, os quais representavam, naquela ocasião, cerca de 90% do mercado de aparelhos móveis. Isto se deu pelo fato de o WAP Fórum ser um fórum aberto a qualquer participante da indústria relacionada à tecnologia WAP.

O principal objetivo do WAP Fórum é agregar empresas de vários segmentos da indústria de redes sem fio para garantir interoperabilidade de produto e alavancar o mercado de redes sem fio.

Outros objetivos do WAP Fórum podem ser definidos como sendo (WIRELESS APPLICATION PROTOCOL FORUM LTDA, 1999):

- a) disponibilizar conteúdo da internet e/ou outros serviços de dados avançados a telefones celulares ou outros terminais sem fio;
- b) criar uma especificação global de protocolos para redes sem fio que irá interoperar entre diferentes tecnologias de redes sem fio (CDMA, GSM, TDMA, etc.);
- c) incentivar a criação de conteúdo e aplicações que serão disponibilizados através de uma variedade de redes sem fio e de tipos de terminais;
- d) adotar, sempre que possível, padrões de tecnologias existentes e estendê-los somente quando necessário.

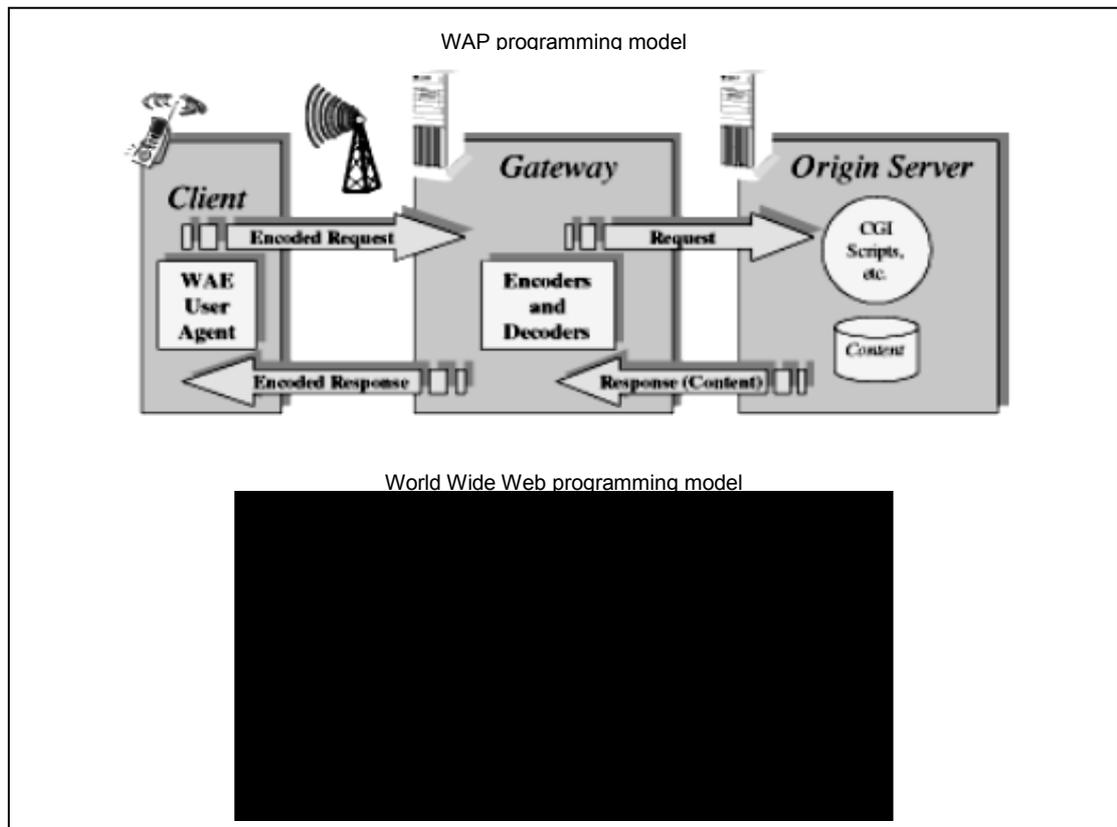
Além disso, o WAP Fórum mantém relações com outros órgãos padronizadores internacionais com o objetivo de assegurar a convergência de futuras especificações. Dentre eles estão o *World Wide Web Consortium (W3C)*, *Telecommunication Industry Association (TIA)*, *Cellular Telecommunication Industry Association (CTIA)* e outros (WIRELESS APPLICATION PROTOCOL FORUM LTDA, 2003).

2.1 MODELO DE REDE WAP

O modelo de programação WAP é bastante parecido com o modelo WWW, o que propicia vários benefícios à comunidade de desenvolvimento, dentre eles a familiarização com o modelo existente, a confiabilidade de uma arquitetura já testada e a possibilidade de utilização de ferramentas comuns.

Entretanto, otimizações e extensões foram realizadas de forma a adaptar este modelo à realidade dos ambientes móveis, conforme se pode ver na figura 1.

Figura 1 – Comparação entre os modelos de rede WWW e WAP



Fonte: adaptado de Wireless Application Protocol Fórum, 1999.

O conteúdo e as aplicações WAP foram especificados em um conjunto de formatos baseados nos formatos WWW. Os dados são transportados utilizando um conjunto de protocolos de comunicação, estes também baseados nos protocolos WWW. Um micro-navegador é o terminal que coordena a interface com o usuário, sendo análogo aos navegadores de internet (DORMAN, 2001).

Também foram definidos, pelo WAP Fórum, um conjunto de componentes que permitem a comunicação entre terminais móveis e o servidor de rede, incluindo (WIRELESS APPLICATION PROTOCOL FORUM LTDA, 1999):

- a) modelo de nomenclatura – o padrão WWW de *Uniform Resource Locators* (URLs) foi utilizado para identificar o conteúdo WAP nos servidores de origem;
- b) tipos diferentes de conteúdo – a todos os conteúdos WAP são fornecidos tipos consistentes com os tipos do WWW. Isso permite aos agentes WAP dos usuários processar corretamente o conteúdo baseado em seu tipo;

c) formatos padrão de conteúdo – formatos de conteúdo WAP são baseados na tecnologia WWW e incluem marcadores de display, informação de calendário, objetos de cartão eletrônico de negócios, imagens e linguagens script;

d) protocolos de comunicação padrão – os protocolos de comunicação WAP permitem a comunicação de pedidos feitos pelo navegador de um terminal móvel ao servidor internet da rede.

O protocolo WAP utiliza a tecnologia de *proxy* para conectar os dispositivos sem fio à internet. O *proxy* WAP é tipicamente responsável pelas seguintes funcionalidades:

a) *gateway* de protocolo – traduz pedidos da pilha de protocolo WAP (WSP, WTP, WTLS e WDP) para a pilha de protocolos WWW (http e TCP/IP);

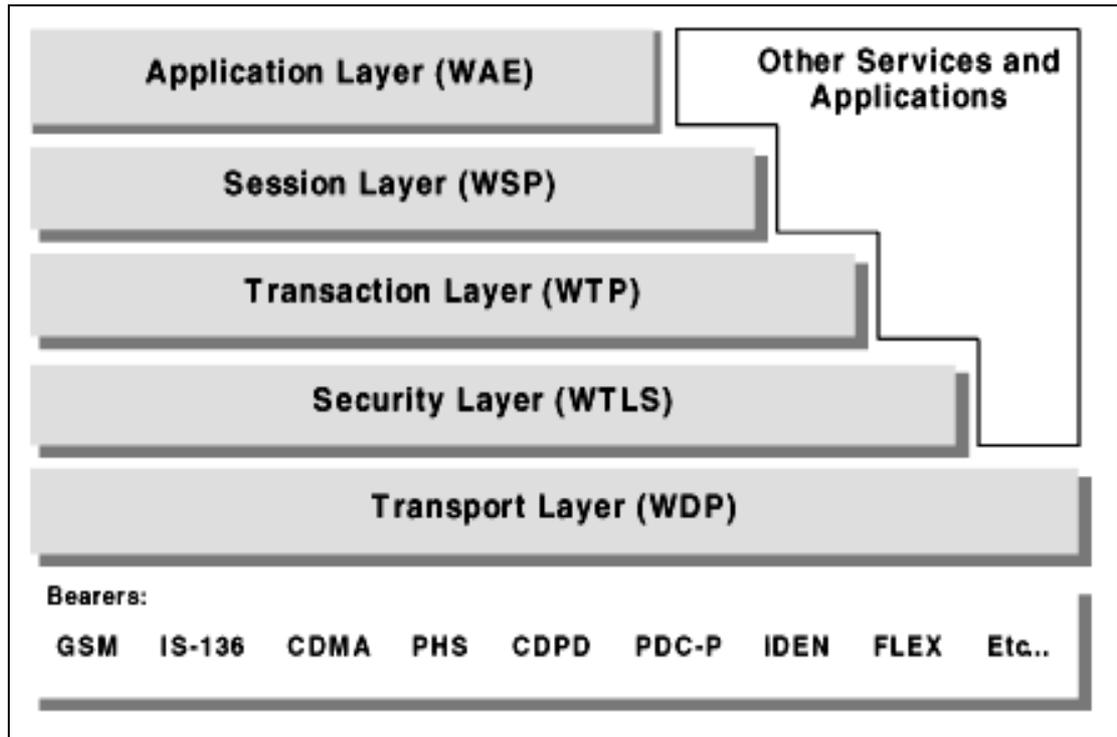
b) codificadores e decodificadores – traduzem o conteúdo WAP para formatos compactos para reduzir o tamanho dos dados na rede.

Essa infra-estrutura garante que usuários de um terminal móvel WAP possam navegar em uma vasta variedade de conteúdos e aplicações, e que os desenvolvedores de aplicações possam construir serviços que rodem numa larga base de terminais móveis. O *proxy* WAP permite que aplicações e conteúdos possam ser hospedados em um *webserver* padrão e com isso possam utilizar tecnologias WWW tais como CGI, ASP e outros tipos de linguagens *script*.

2.2 ARQUITETURA WAP

A arquitetura WAP é formada por um ambiente escalável e extensivo para o desenvolvimento de aplicações para dispositivos de comunicação móveis. Isto se realiza por meio do protocolo WAP, que é composto por várias camadas de subprotocolos (figura 2), em que cada camada pode ser acessada pela camada superior e por outros serviços e aplicações, e ainda, aplicações externas podem acessar diretamente as camadas de sessão, transação, segurança e transporte (DEMÉTRIO, 2000).

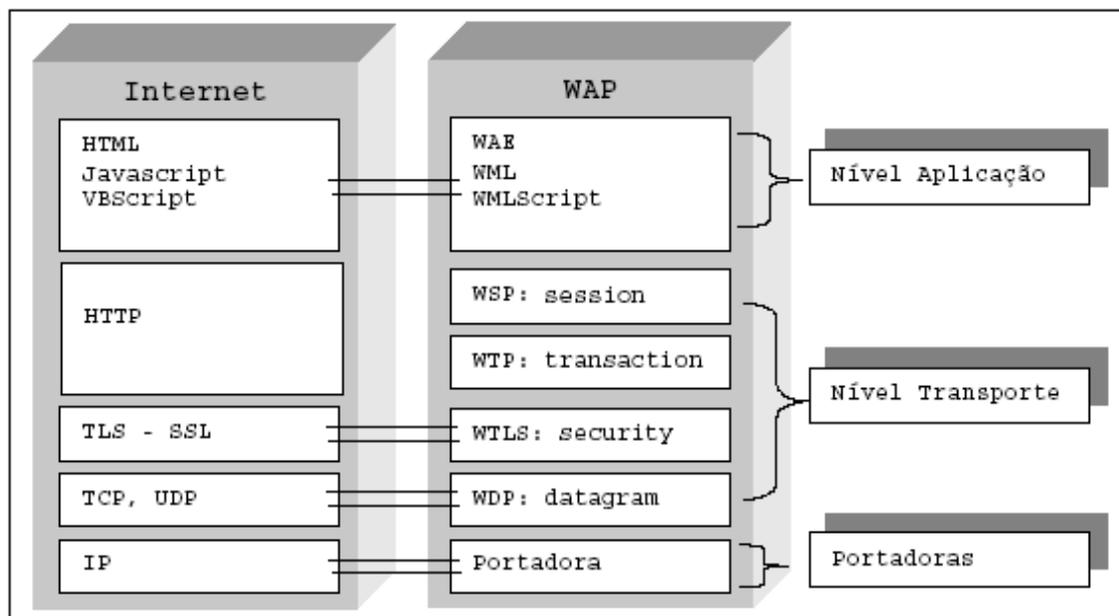
Figura 2 – Pilha da arquitetura WAP



Fonte: Wireless Application Protocol Fórum, 1999.

A pilha de protocolos WAP é análoga à usada hoje na internet, como se pode ver na figura 3. Os protocolos cobrem tanto o nível de aplicação (WML e WMLScript) como o nível de transporte (WSP e WTP, WTLS e WDP). Existem vários tipos de tecnologias portadoras (*bearers*) como SMS, CDMA, TDMA, GSM, etc., mas em qualquer uma delas o desenvolvimento de aplicações se dará de uma mesma forma.

Figura 3 – Analogia entre os protocolos internet e WAP



Fonte: Wireless Application Protocol Fórum, 2003.

2.2.1 WIRELESS APPLICATION ENVIRONMENT (WAE)

WAE é a camada de aplicação de propósito geral baseada na combinação das tecnologias WWW e de telefonia móvel. Seu principal objetivo é estabelecer uma interação entre os operadores e os provedores de serviços, para permitir a construção de aplicações de uma forma eficiente, independente do dispositivo em que vão ser executadas. WAE inclui um micro-navegador com as seguintes funções (DEMÉTRIO, 2000):

- a) *Wireless Markup Language* (WML), uma linguagem leve, otimizada, semelhante a HTML, porém específica para uso em dispositivos móveis;
- b) WMLScript, uma linguagem similar ao JavaScript que otimiza a WML;
- c) serviços de telefonia e interfaces de programação *Wireless Telephony Application* (WTA);
- d) suporte a imagens, calendários, agendas, etc.

WAE não especifica formalmente nenhum agente-usuário, ou seja, suas características e capacidades são deixadas para os implementadores. WAE somente define serviços e

formatos fundamentais que são necessários para assegurar a interoperabilidade entre as diferentes implementações.

2.2.2 WIRELESS SESSION PROTOCOL (WSP)

WSP permite a aplicação de uma interface para o estabelecimento de sessões. Para isso foram criados dois tipos de protocolos diferentes (DEMÉTRIO, 2000). O primeiro para a conexão mediante o serviço de transações *Wireless Transaction Protocol* (WTP) e o outro permite acessar diretamente a camada *Wireless Datagram Protocol* (WDP), sem a necessidade de estabelecer uma conexão direta com as outras camadas, o que melhora o rendimento de aplicações que não necessitam de confirmação de envio de dados, além de permitir a troca eficiente de dados entre as aplicações.

A diferença básica é que nos serviços orientados à conexão são disponibilizadas facilidades de gerenciamento e segurança na transmissão entre o cliente (*browser WAP*) e o servidor (*gateway WAP*). Se por algum motivo a transmissão se torna impossível, a mesma pode ser interrompida e retomada mais tarde partindo do ponto de parada anterior. Existe um rígido controle entre os pacotes enviados e recebidos e esse controle obviamente gera um tráfego um pouco maior do que nos serviços sem conexão. Já no caso destes últimos, não existe nenhum controle de recebimento de pacotes enviados pelo dispositivo, ou seja, os serviços não são confirmados. Nesse caso a sessão de transmissão não é considerada segura e sua principal vantagem é que gera menos tráfego de rede (CANI, 2000).

O núcleo WSP é um formato binário utilizado pelo http, o que permite a inclusão de cabeçalhos e informações. Dessa forma, os métodos utilizados pelo HTTP 1.1 são suportados garantindo a compatibilidade com o mesmo. Convém salientar que WSP, por si só, não interpreta a informação do cabeçalho nas requisições e respostas, e que seu ciclo de vida não está ligado ao transporte, uma vez que a sessão pode ser suspensa quando estiver ociosa, para fins de liberação de recursos de rede ou economia de bateria do dispositivo (DORMAN, 2001).

2.2.3 WIRELESS TRANSACTION PROTOCOL (WTP)

É um protocolo simplificado, idealizado para situações de baixa largura de banda, caso específico das comunicações sem fio (DEMÉTRIO, 2000).

A camada WTP da pilha de protocolos WAP é a responsável por controlar o envio e recepção das mensagens, oferecendo suporte de transações, o que acrescenta confiança ao serviço de datagramas provido pela camada WDP, uma vez que libera a camada superior a retransmissões e reconhecimentos necessários quando esse serviço é utilizado (DIAS, 2000).

Este protocolo possui várias características, dentre as quais podemos citar (DORMAN, 2001):

- a) o alcance da confiabilidade que é feito através de identificadores únicos de transação, reconhecimento, remoção de duplicação e retransmissões;
- b) não possui fase de criação e destruição de conexão explícita, uma vez que este procedimento de abertura e fechamento de conexão impõe uma excessiva sobrecarga na rede de comunicação;
- c) possui segurança opcional, isto é, o usuário do WTP pode confirmar cada mensagem enviada;
- d) faculta ao último reconhecimento da transação a contenção de informação fora da banda relacionada à transação, como medida de desempenho;
- e) usa a concatenação para transportar múltiplas unidades de dados do protocolo em uma única unidade de dados de serviço, que será transportada pelo WDP;
- f) baseia-se na mensagem, isto é, a unidade básica de intercâmbio é uma mensagem inteira e não uma cadeia de bytes;
- g) usa mecanismos para minimizar o número de transações sendo retornadas como resultado de pacotes duplicados;
- h) possibilita o cancelamento de transações atuais, incluindo o descarte de dados não enviados tanto do cliente quanto do servidor. O cancelamento pode ser disparado pelo usuário ou automático;
- i) dá segurança através de mensagens de execução confiáveis. Tanto o sucesso quanto a falha são reportados, isto é, se uma execução não pode ser manipulada pelo seu destinatário, então uma mensagem de cancelamento será retornada para o remetente ao invés do resultado;
- j) permite transações assíncronas, isto é, o destinatário envia o resultado assim que os dados tornam-se disponíveis.

2.2.4 WIRELESS TRANSPORT LAYER SECURITY(WTLS)

O protocolo de segurança na arquitetura WAP provê para as camadas superiores um serviço de interface de transporte seguro, preservando o serviço de interface de transporte abaixo dele. A camada WTLS é modular, isto é, o seu uso não depende do nível de segurança requerido em uma aplicação, ou seja, é uma camada opcional da pilha WAP. Além da segurança, o WTLS fornece uma interface para gerenciamento de conexões seguras (DORMAN, 2001).

O WTLS disponibiliza os seguintes serviços:

- a) integridade de dados – garante que não haverá alteração do conteúdo nos dados enviados;
- b) privacidade – assegura que não haverá como descriptografar a mensagem, a não ser o transmissor ou receptor;
- c) autenticação do cliente – limita o acesso através do servidor de conteúdo, ou seja, apenas quem tem permissão pode acessar determinado conteúdo;
- d) autenticação do servidor – certifica que o servidor acessado é realmente o desejado.

Aplicações são capazes de habilitar ou desabilitar características do WTLS dependendo da suas necessidades de segurança ou das características da rede.

2.2.5 WIRELESS DATAGRAM PROTOCOL (WDP)

É a camada que transporta os dados, responsável pelo envio e recebimento de mensagens pelos vários tipos de redes. Como o protocolo WDP proporciona uma interface entre os protocolos WAE, WSP e WTLS, é capaz de funcionar de forma independente da rede em que se está trabalhando (DIAS, 2000).

Os serviços oferecidos pelo WDP incluem endereçamento de aplicação por número de porta e segmentação opcional, além de reunir a detecção de erros. Estes serviços permitem às aplicações operarem transparentemente sobre diferentes portadoras (*bearers*) disponíveis. O número da porta identifica a entidade mais acima do WDP que pode ser o WTP ou WSP ou até mesmo uma aplicação de correio eletrônico (DORMAN, 2001).

2.2.6 BEARERS (PORTADORAS)

No protocolo WAP a transferência de dados é realizada em diferentes tipos de portadoras, incluindo *Short Message Service* (SMS), com até 160 caracteres alfas-numéricos ou binários e confirmação de entrega, comutação por circuitos ou comutação por pacotes. Esses tipos são implementados pelos diferentes tipos tecnológicos de rede sem fio.

Cada *bearer* oferece um nível de qualidade de serviço diferente segundo sua velocidade de transmissão, taxa de erros e taxa de atrasos. A pilha de protocolos WAP existe, então, para compensar ou tolerar essas diferenças.

2.3 LINGUAGENS PARA PROGRAMAÇÃO WAP

Foram criadas duas linguagens de programação específicas para WAP. A primeira é a *Wireless Markup Language* (WML), baseada na linguagem XML, constituída por um conjunto de funcionalidades herdadas do HTML. A outra foi desenvolvida para criação de páginas dinâmicas, a WMLScript, bastante parecida com outras linguagens de *script* como o JavaScript e o VBScript.

2.3.1 WIRELESS MARKUP LANGUAGE (WML)

A geração de conteúdo para internet está fundamentalmente, baseada na construção de páginas WEB através da linguagem HTML. Na tecnologia WAP não poderia ser diferente, porém, utiliza-se uma linguagem com sintaxe semelhante chamada WML, baseada no XML. As páginas WAP podem ser exibidas no micro navegador incorporado aos dispositivos móveis.

Ao contrário da estrutura plana de documentos HTML, documentos WML são divididos em um conjunto de unidades bem definidas para interação com o usuário. Como parte deste contexto, criam-se os conceitos de *deck* e *card*.

Um simples documento WML é conhecido como um *deck*, representando uma aplicação ou serviço. Um *deck* contém um ou vários *cards*, e seu limite fica compreendido entre as *tags* `<wml>` e `</wml>`.

Um *card* é definido como uma unidade básica de *display* no WAP. Pode ser visto como uma tela de informação, que pode conter texto, imagens, *hiperlinks* e campos de entrada. Em um *deck*, compreende-se por *card* todo código escrito entre as tags `<card>` e `</card>`.

No quadro 1 pode-se visualizar a diferença entre os conceitos de *deck* e *card* em um documento WML.

Quadro 1 – Diferença entre *deck* e *card*

```
<?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1//EN"
"http://www.wapforum.org/DTD/wml_1.1.xml">

<wml> <!-- início do deck -->

  <card id="splash" ontimer="#aviso"> <!-- início 1º card -->
  <timer value="10"/>
  <p align="center">
    
  </p>
</card> <!-- fim do 1º card -->

  <card id="aviso" title="TJSC WAP" newcontext="true"> <!--início do 2º card -->
  <p align="center">
    <b>Consulta de Processos 1&#186; Grau</b><br/>
    <small>Alertamos que os resultados aqui obtidos s&#227;o de car&#225;ter
    informativo, n&#227;o substituindo avisos ou intimaç&#245;es publicados
    no Di&#225;rio da Justi&#231;a<br/>
    <a href="pesquisa.wml#index">&#91;OK&#93;</a><br/></small>
  </p>
</card> <!-- fim do 2º card -->

</wml> <!-- fim do deck -->
```

Um *deck* é similar a uma página HTML, já que é identificado por uma URL. Assim sendo, é uma unidade de conteúdo de transmissão. Seu conteúdo é limitado a 1.400 bytes, correspondente ao arquivo binário codificado obtido a partir do código-fonte WML e não de seu tamanho original.

O uso de caracteres especiais na linguagem WML deve ter como base a equivalência contida na tabela Latin-1 (ISO 8859-1).

Em WML pode-se utilizar imagens de até 150x150 *pixels*. Estas imagens estão no formato *Wireless Bitmap* (WBMP), desenvolvido especialmente para esta linguagem.

2.3.2 WIRELESS MARKUP LANGUAGE SCRIPT (WMLSCRIPT)

A especificação WAP define uma linguagem de *script* chamado WMLScript, que vem suprir a deficiência de processamento do WML, tais como a falta de acesso aos periféricos dos dispositivos, a impossibilidade da declaração de variáveis e de construções de lógicas algorítmicas, tornando-a mais robusta (DEMÉTRIO, 2000).

Segundo Dias (2000), WMLScript é uma linguagem leve e procedural, que realça as facilidades de navegação e apresentação do WML, além de possuir capacidade de suporte ao comportamento da interface do usuário de modo mais avançado, adicionando inteligência ao cliente, provendo um mecanismo conveniente de acesso aos dispositivos e seus periféricos, reduzindo a necessidade de idas e voltas ao servidor.

O conteúdo WMLScript deve ser desenvolvido em um arquivo distinto (com extensão wmls), ou seja, não pode estar junto do conteúdo de um documento WML. Para fazer uso da lógica WMLScript desenvolvida é necessária a existência, no documento WML, de comandos de chamamento, como no quadro 2.

Quadro 2 – Uso do WMLScript

Pesquisa.wml

```
<?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1//EN"
"http://www.wapforum.org/DTD/wml_1.1.xml">
<wml>
  <card id="index" title="TJSC WAP" newcontext="true">
    ...
    <a href="scripts.wmls#verificadados()">&#91;Pesquisar&#93;</a> <!-- chamada da
lógica WMLScript -->
    ...
  </card>
</wml>
```

Scripts.wmls

```
extern function verificadados()
{
  var valorpp = (WMLBrowser.getVar("valorpesq"));
  var tipopp = (WMLBrowser.getVar("tipopesq"));
  if (valorpp != "")
  {
    var urlconsulta = "http://servidordadosibirama/consultas.asp?tipopp=" + tipopp
+ "&valorpp=" + valorpp
    WMLBrowser.go(urlconsulta);
  }
  else
  {
    WMLBrowser.go("erro.wml");
  }
};
```

2.4 USO DA TECNOLOGIA ASP PARA GERAÇÃO DE PÁGINAS WML

Active Server Pages (ASP) é um dos recursos da programação que viabiliza a criação de páginas dinâmicas, ou seja, páginas com conteúdo variável (WEISSINGER, 1999).

É um ambiente de programação interpretado por *scripts* no servidor, onde se pode criar páginas dinâmicas interativas e de alta performance. Quando o cliente solicita a página o servidor interpreta a requisição e devolve o resultado.

Para utilizar ASP precisa-se de um servidor que interprete esta tecnologia. Na criação de páginas dinâmicas em WML, além desta providência, é necessário alterar o cabeçalho padrão utilizado naquele tipo de arquivo para o utilizado nos arquivos WML, limitando o conteúdo do código programático ASP entre as *tags* `<%` e `%>`, como no quadro 3.

Quadro 3 – Geração de páginas WML utilizando ASP

Partes.asp

```
<% Response.ContentType = "text/vnd.wap.wml" %><?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1//EN"
"http://www.wapforum.org/DTD/wml_1.1.xml"> <!-- cabeçalho padrão para documentos WML -->
<wml>
<card id="consultas" title="RESULTADOS - TJSC WAP">
<p align="center">
<%
' variáveis utilizadas
Dim conn
Dim rsdados
Dim tipopp
Dim valorpp
Dim contador
' caracteres de conexão ao banco de dados
set conn=server.createobject("adodb.connection")
conn.open "Provider=MSDAORA.1;Data Source=wap;User id=wap;Password=wap;Persist Security
Info=True"
set rsdados=server.createobject("adodb.recordset")
response.write "<b>Processos com a parte selecionada:</b><br/>"
' executa a seleção no banco de dados
set rsdados=conn.execute("select distinct nrproc from procparte where segredojustica = 0
and cdparte = "&Request.QueryString("codigo")&" order by nrproc")
while not rsdados.eof
' exibe os resultados obtidos na forma de link's
response.write"<a
href=""http://servidordadosibirama/consultas.asp?tipopp=1&#38;valorpp="
rsdados.Fields("nrproc") & "">" & "0" & rsdados.Fields("nrproc") & "</a><br/>"
rsdados.movenext
wend
' fecha a conexão com o servidor de dados
rsdados.close
set rsdados=nothing
conn.close
set conn=nothing
%>
</p>
</card>
</wml>
```

3 DESENVOLVIMENTO DO SISTEMA

Para o desenvolvimento do sistema em questão, foram levadas em consideração as necessidades do TJSC, os recursos tecnológicos disponíveis para o desenvolvimento, bem como as limitações destes recursos. A seguir é feita uma breve introdução do que trata o sistema judiciário e posteriormente é descrito o sistema.

3.1 O SISTEMA JUDICIÁRIO

Com o passar dos anos, a velocidade da informação fez com que as pessoas viessem a conhecer seu direitos. De uma maneira geral, os indivíduos lesados buscam nos tribunais o remédio jurídico para seus problemas.

A justiça comum, onde tramita a maioria dos processos, é o órgão a quem compete o julgamento das causas cíveis e criminais da população civil.

Muitas pessoas julgam o poder judiciário um órgão moroso e assoberbado de tarefas. Isto se deve o fato de existir dispositivos para a normalização da tramitação, descritos nos códigos do processo civil e criminal brasileiros.

Pode-se resumir a tramitação processual nos seguintes passos (TJSC, 2002):

O processo chega ao fórum por intermédio da delegacia de polícia (inquéritos, boletins de ocorrência, etc.), ou pelos advogados representando o interesse de seus clientes. Existem também casos em que juízes deprecam atos a juízes de outras comarcas pelo fato de serem atos de competência exclusiva, estes processos chamam-se cartas precatórias.

O processo é cadastrado e distribuído a uma das varas existentes na comarca, a quem compete sua tramitação, e ao juiz que responde por ela, seu julgamento.

Em se tratando de processos de interesse particular (cobranças, execuções, etc.) são cobradas custas processuais para bancar as despesas relativas a tramitação processual.

Existem casos em que as pessoas que são acionadas ou até mesmo as que necessitam requerer algo via processo são dispensadas de pagar as custas processuais, pois é garantido pela constituição o direito à justiça gratuita.

Nas causas em que o interesse processual versa estreitamente sobre as partes envolvidas, cabendo o conceito de segredo de justiça, que é um dispositivo utilizado para resguardar o interesse e o nome das partes envolvidas. São exemplos de processos em segredo de justiça, todos aqueles que envolvem menores, dissoluções de sociedade de fato (divórcios, separações, etc.) e processos cíveis os quais se requer e se tem deferido pelo magistrado o processamento sigiloso em detrimento da frustração de algum ato processual.

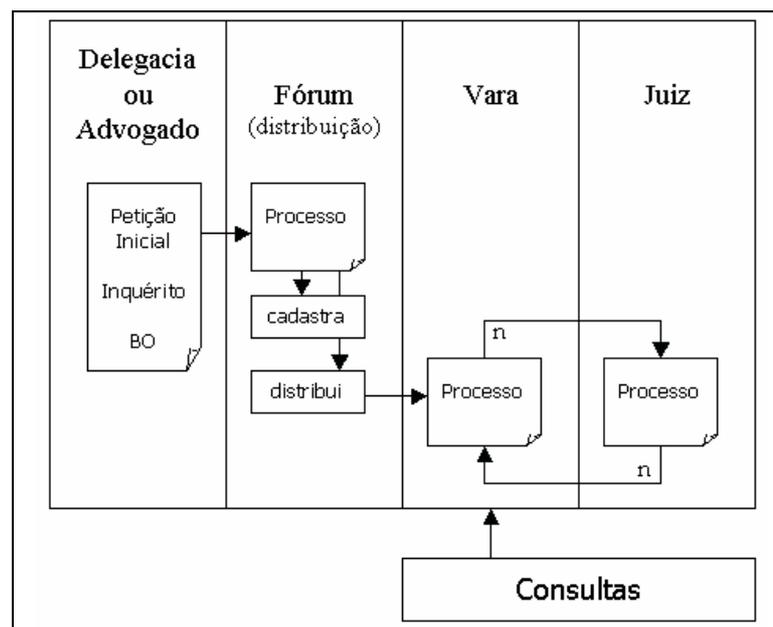
Dentre os atos procedidos nos autos, temos as movimentações processuais, que são utilizados para a elaboração de mapa estatístico para o TJSC acompanhar o acréscimo processual de cada comarca. Além disso, são utilizadas as localizações processuais que servem para saber a posição física exata do processo no fórum.

No cumprimento das determinações judiciais são procedidos diversos atos, como a expedição de mandados, ofícios, designações de audiências, dentre outros.

Outro ato importante na tramitação processual é a intimação dos advogados para certos procedimentos atinentes aos autos.

Em cada uma destas fases, os advogados precisam estar informados do andamento processual, como ilustrado na figura 4.

Figura 4 – Tramitação processual



Levando-se em conta a impossibilidade da informação do andamento dos processos via telefone, sabe-se que antes do desenvolvimento deste trabalho, os advogados tinham acesso às informações processuais de três maneiras: pelo *site* na internet, pelo terminal de consulta existente na sala da OAB de cada comarca, ou no balcão dos cartórios judiciais.

Pretende-se com este trabalho, disponibilizar aos advogados ou a qualquer interessado pelo andamento dos processos, o acesso móvel a estas informações.

3.2 CARACTERÍSTICAS DO PROBLEMA

O sistema deve extrair do servidor de dados as informações processuais decorrentes do tipo de pesquisa e dos dados informados para a mesma, não podendo em hipótese alguma, prejudicar o segredo de justiça de certos casos.

Outro ponto é a utilização de uma tecnologia alternativa para que o servidor de dados Sun Solaris, utilizado no TJSC, possa rodar aplicativos ASP.

3.3 ESPECIFICAÇÃO

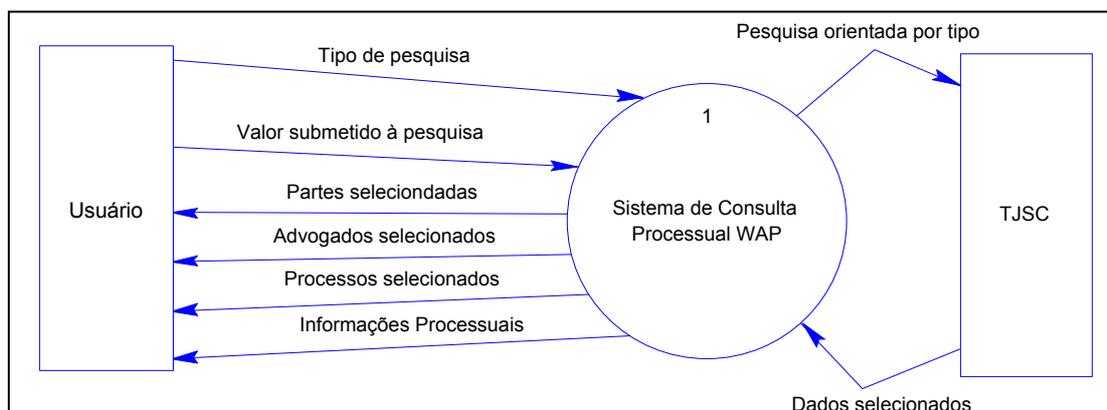
A metodologia utilizada no desenvolvimento da especificação do sistema foi a análise estrutura.

Para a especificação do sistema foi utilizada a ferramenta Power Designer, na qual foram criados o diagrama de contexto, os diagramas de fluxo de dados (DFD) e o diagrama entidade-relacionamento (DER) lógico.

3.3.1 DIAGRAMA DE CONTEXTO

Neste diagrama (figura 5) apresenta-se a interação do usuário como o sistema de uma maneira total, representando quais informações são fornecidas e quais retornam ao usuário.

Figura 5 – Diagrama de contexto



Pode-se resumir esta interação como sendo:

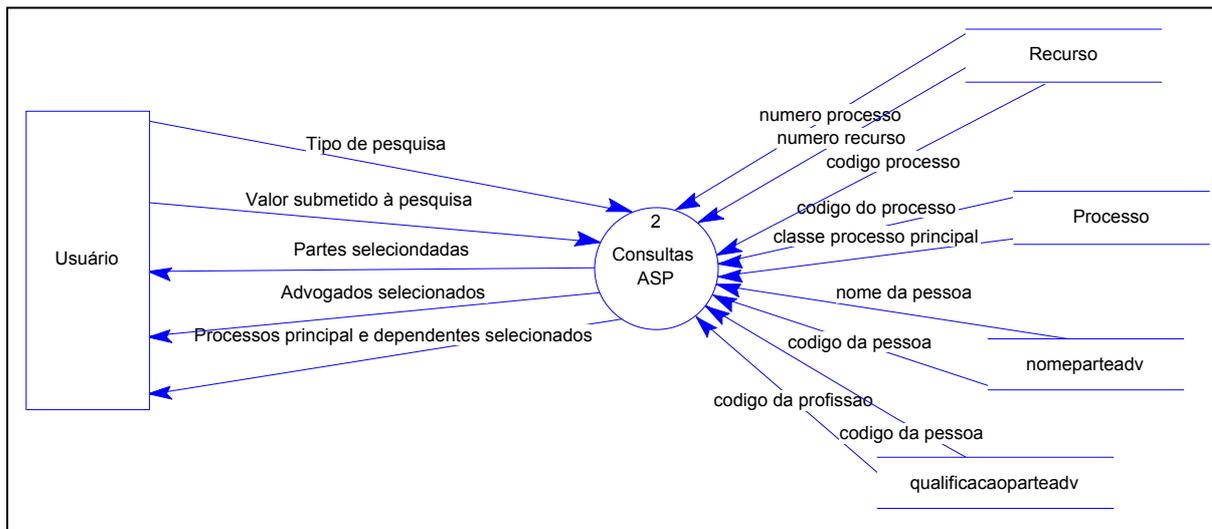
- a) o usuário informa o tipo de pesquisa desejada e os dados que serão submetidos, podendo optar por três tipos de pesquisa: por número de processo, pelo nome da parte ou pelo nome do advogado;
- b) o sistema valida os dados informados, submetendo-os ao processamento. Caso haja inconsistência nos dados que o usuário forneceu, o sistema mostra uma mensagem de erro.
- c) numa pesquisa em que os dados informados são consistentes, o sistema vai buscar na servidor do TJSC os dados que serão manipulados a fim de gerar uma resposta ao usuário. Esta resposta dependerá do tipo de pesquisa selecionada.
- d) para a obtenção das informações processuais, o usuário interage por várias telas, estas geradas a partir dos dados fornecidos para a pesquisa na tela de consulta.

3.3.2 DIAGRAMAS DE FLUXO DE DADOS (DFD)

Nestes diagramas apresentam-se as principais funcionalidades do sistema. Estas funcionalidades são relativas aos arquivos ASP que foram utilizados para o acesso a base de dados e para geração das páginas dinâmicas WML, uma vez que somente nestes arquivos há fluxo de dados.

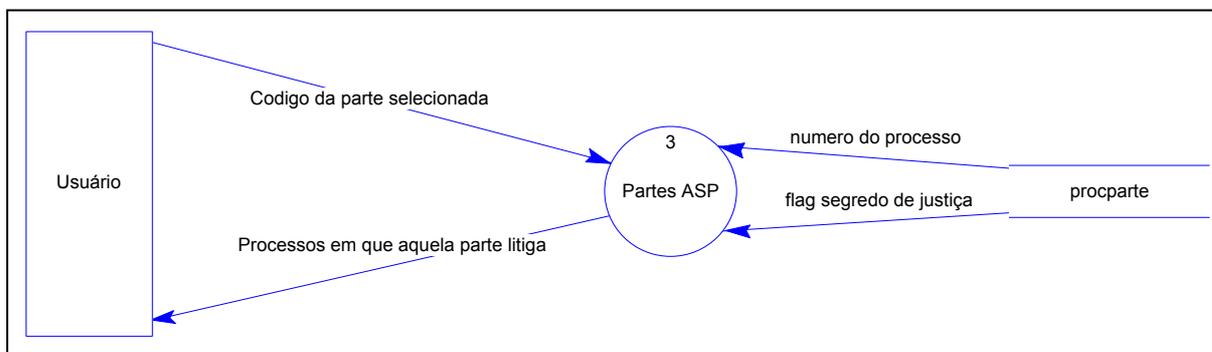
No arquivo consultas.asp (figura 6) o usuário informa o tipo de pesquisa desejada e os dados que serão submetidos. Caso o usuário tenha solicitado a pesquisa por número, obtêm como resposta a classe e o número do processo principal e todos os dependentes. Já se tiver pesquisado pelo nome da parte ou do advogado, obtêm uma listagem de todos os nomes que contenha o nome informado.

Figura 6 – DFD consultas.asp



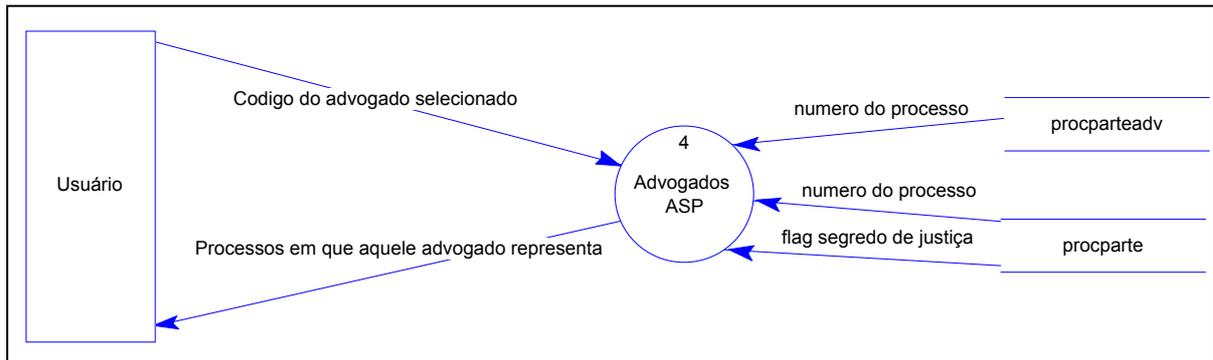
No arquivo partes.asp (figura 7) o usuário seleciona uma das partes listada na etapa anterior (consultas.asp). Implicitamente o sistema passa o código daquela como parâmetro de entrada. Como resultado o usuário recebe uma listagem de todos os processos em que aquela parte está envolvida com um link para o arquivo consultas.asp, desde que não esteja em segredo de justiça.

Figura 7 – DFD partes.asp



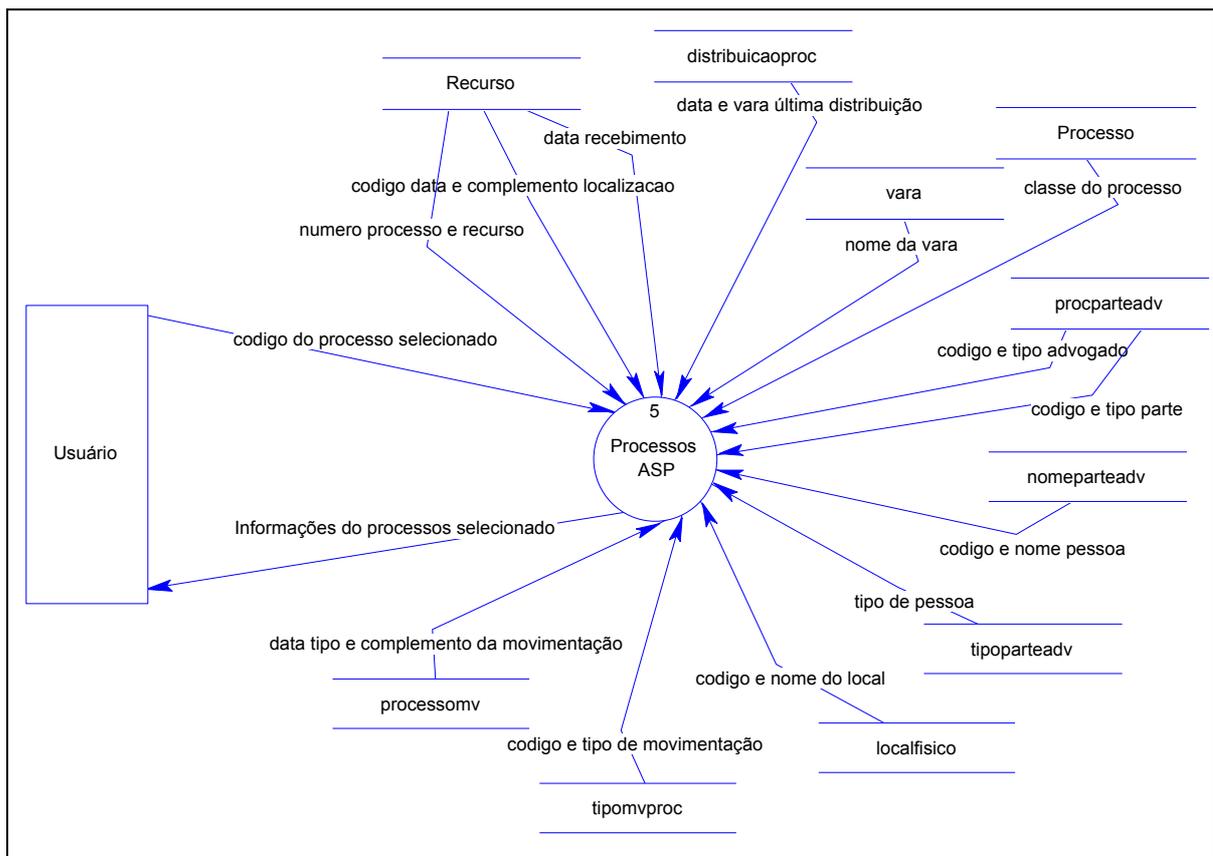
No arquivo advogados.asp (figura 8) o usuário seleciona um dos advogados listado na etapa anterior (consultas.asp). Implicitamente o sistema passa o código daquele como parâmetro de entrada. Como resultado o usuário recebe uma listagem de todos os processos em que aquele advogado representa alguma das partes com um link para o arquivo consultas.asp, desde que esta não esteja em segredo de justiça.

Figura 8 – DFD advogados.asp



No arquivo processos.asp (figura 9) o usuário seleciona um dos processos listado na etapa anterior (consultas.asp). Implicitamente o sistema está passando o código daquele como parâmetro de entrada. Como resultado o usuário recebe as informações processuais daqueles autos.

Figura 9 – DFD processos.asp



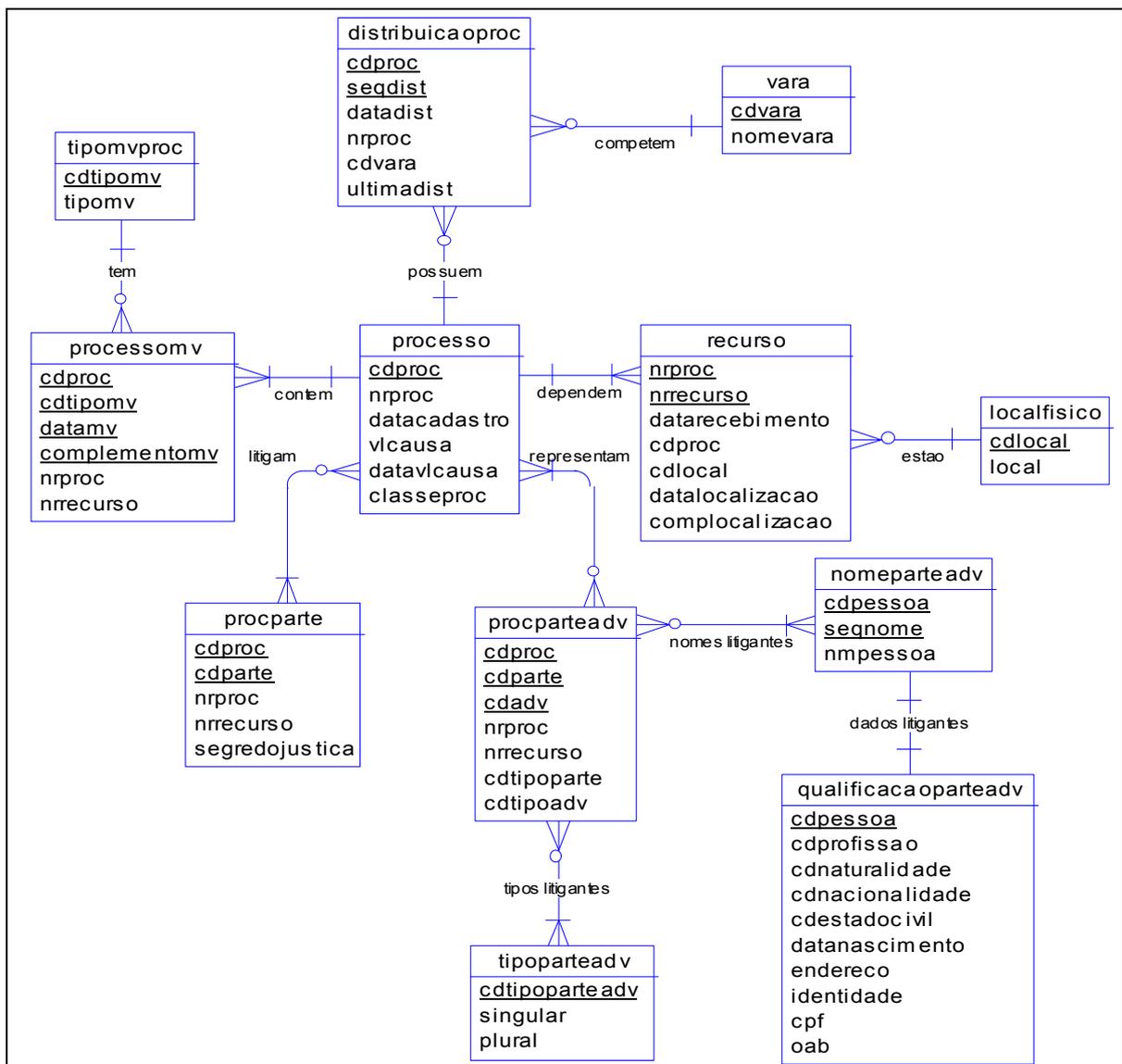
3.3.3 DIAGRAMA ENTIDADE-RELACIONAMENTO (DER)

Neste diagrama apresenta-se a interação entre as entidades contidas na base de dados ORACLE (ORACLE, 2003) do TJSC.

Esta especificação mostra tão somente como são constituídas as tabelas desta base de dados, bem como seus inter-relacionamentos. Não foi promovida nenhuma alteração, já que o sistema limita-se apenas a consultá-la.

No nível lógico (figura 10) apresentam-se todos os relacionamentos entre as tabelas que constituem a base de dados.

Figura 10 – DER lógico



As tabelas consultadas pelo sistema são as seguintes:

- a) processo – armazena os dados relativos ao cadastro dos processos, bem como as classes as quais eles pertencem;
- b) distribuicaoproc – armazena os dados relativos a distribuição dos processos. O processo pode ser cadastrado, porém não distribuído. É o caso em que há falta de preparo (não pagamento das custas iniciais relativas ao processo);
- c) vara – armazena as varas da comarca, a quem competem a tramitação dos processos;
- d) recurso – são todos os passos evolutivos de uma tramitação processual, os quais dependentes dos autos principais. Recursos podem ser apelações, embargos de declaração, execuções de sentenças, etc. No modelo em tela, primeiro recurso é o próprio processo;
- e) procparte – armazena as partes que litigam em determinado processo, as quais podem estar sobre segredo de justiça;
- f) procparteadv – armazena a representação processual, ou seja, as partes e respectivos advogados. No modelo em tela, os advogados e seus clientes são tipificados para representar os pólos ativo e passivo do processo;
- g) tipoparteadv - armazena a tipificação da partes (autor, réu, executado, etc.);
- h) nomeparteadv – armazena o nome das pessoas que têm ou tinham processos tramitando ou aquelas que já serviram de testemunha para algum caso;
- i) qualificacaoparteadv – armazena dados pessoais da pessoas que têm ou tinham processos tramitando ou aquelas que já serviram de testemunha para algum caso;
- j) processomv – armazena todas as movimentações que o processo já tiveram para um rápida e eficiente análise, realizada para fins de elaboração de mapa estatístico;
- k) tipomvproc – armazena os tipos de movimentação processual existente;
- l) localfisico – armazena os nomes dos locais onde se pode encontrar os processos. Esta tabela se relaciona com a tabela recurso que pode conter ou não, uma localização diferente para cada recurso que está tramitando.

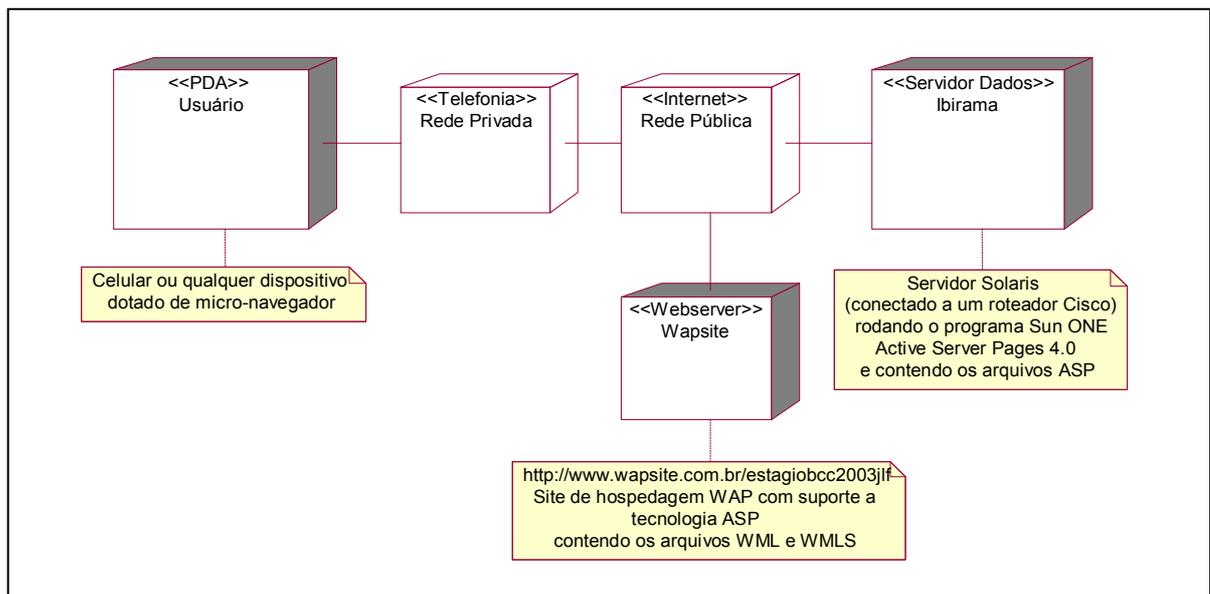
3.4 IMPLEMENTAÇÃO

O sistema foi implementado nas instalações da comarca de Ibirama -SC e toda infraestrutura necessária para a realização deste trabalho, com exceção do *WAPserver*, foi disponibilizado pela Diretoria de Informática do TJSC.

Para suprir a necessidade de disponibilizar as páginas WAP aos usuários externos à intranet do TJSC foi utilizado o *site* de hospedagem WAPsite (WAPsite, 2003).

O diagrama de *deployment* apresentado na figura 11 (com estereótipos desenvolvidos especialmente para o caso em tela) demonstra a topologia do sistema, possibilitando a visualização da distribuição física de seus componentes.

Figura 11 – Infra-estrutura associada à implementação



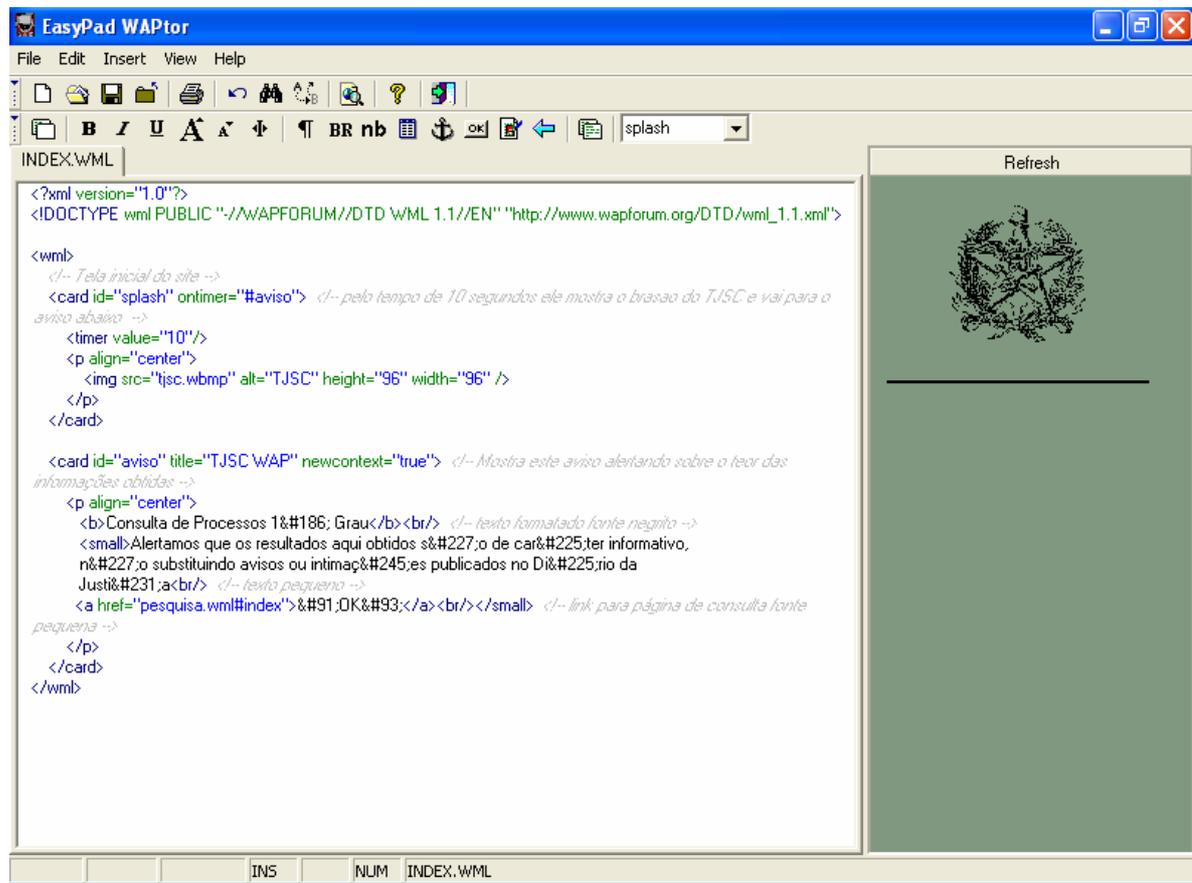
3.4.1 TÉCNICAS E FERRAMENTAS UTILIZADAS

Para a implementação do sistema foram utilizadas as ferramentas EasyPad Waptor, M3Gate, Personal Web Server, Sun ONE Active Server Pages 4.0, Internet Explorer 6.0, além das linguagens WML, WMLScript e ASP já descritas neste trabalho.

3.4.1.1 EASYPAD WAPTOR

EasyPad Waptor (figura 12) é nome de um editor WML que permite a visualização do resultado assim que o código está sendo criado, possibilitando assim um *feedback* ao desenvolvedor.

Figura 12 – EasyPad Waptor



Neste editor há duas janelas. Uma de edição onde é escrito o código-fonte e a outra de visualização, onde se apresenta o resultado da implementação.

No caso desta implementação o EasyPad Waptor foi utilizado para edição dos códigos-fonte WML e ASP.

3.4.1.2 M3GATE

O M3Gate é um simulador WAP da empresa Numeric Algorithm Laboratories, que permite testar e depurar o código WML das páginas implementadas.

Além disso, o M3Gate intercepta chamadas do protocolo http do navegador de internet padrão (Netscape ou Internet Explorer) quando se trata de páginas WML. Neste caso é possível a utilização de *browsers* tradicionais para a visualização das páginas WML, como apresentado na figura 13.

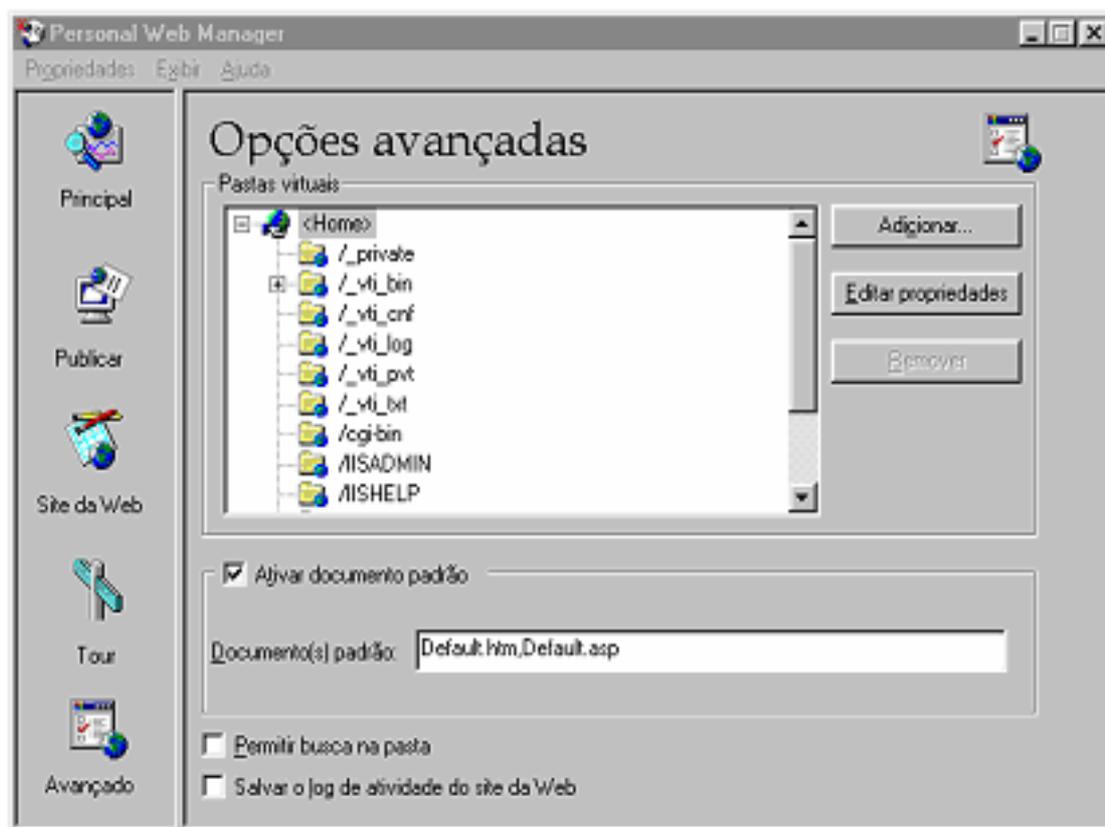
Figura 13 – M3Gate



3.4.1.3 PERSONAL WEB SERVER

O Personal Web Server (figura 14), mais conhecido como PWS, é o *webserver* encontrado nos sistemas operacionais Windows 95/98 para a publicação de páginas de internet pessoais. Este servidor permite trabalhar com a tecnologia ASP.

Figura 14 – Personal Web Server



Todas as máquinas da rede interna do TJSC têm acesso ao servidor de dados por meio de um *alias* para o uso do sistema de automação (SAJ/PG). Devido à necessidade da implementação do sistema, o PWS foi instalado numa destas máquinas com sistema operacional Windows, a fim de possibilitar a interpretação da tecnologia ASP enquanto o Sun ONE Active Server Pages não era instalado no servidor.

3.4.1.4 SUN ONE ACTIVE SERVER PAGES 4.0

O Sun ONE Active Server Pages é um *gateway* que possibilita aos sistemas operacionais da família UNIX, como o Sun Solaris, interpretar e rodar a tecnologia ASP.

Inicialmente este produto foi chamado de Chili!Soft ASP, pois pertencia à empresa Chili!Soft que foi comprada pela Sun em 2002. A partir de então o produto veio a integrar a linha Sun ONE, que é destinada aos sistemas operacionais da família UNIX, proporcionando a estes rodar aplicativos desenvolvidos para outros sistemas operacionais.

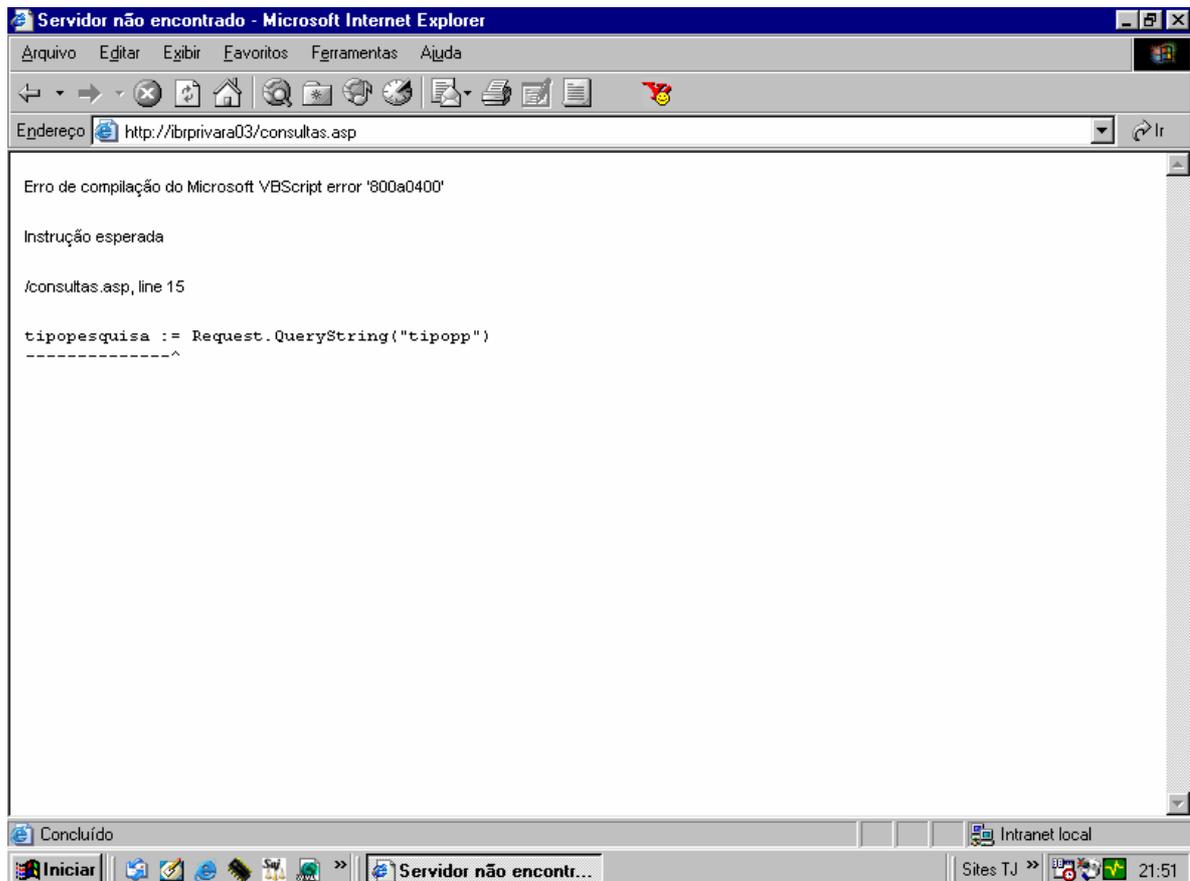
Como havia sido comentado, devido à instalação do PWS durante a fase de implementação, somente uma máquina da rede interna do TJSC tinha acesso às informações geradas pelo sistema

Neste trabalho, Sun ONE Active Server Pages veio possibilitar a chamada externa dos arquivos ASP, que agora estão residentes no servidor de dados Sun Solaris, uma vez que somente esta máquina da comarca de Ibirama por ser acessada da Internet.

3.4.1.5 INTERNET EXPLORER 6.0

O Internet Explorer 6.0, navegador de internet da Microsoft, foi utilizado para os testes e validação dos arquivos ASP implementados, e também para se testar a interceptação do M3Gate em chamadas http destinadas à abertura de páginas WML. A figura 15 mostra um erro de compilação durante a fase de desenvolvimento.

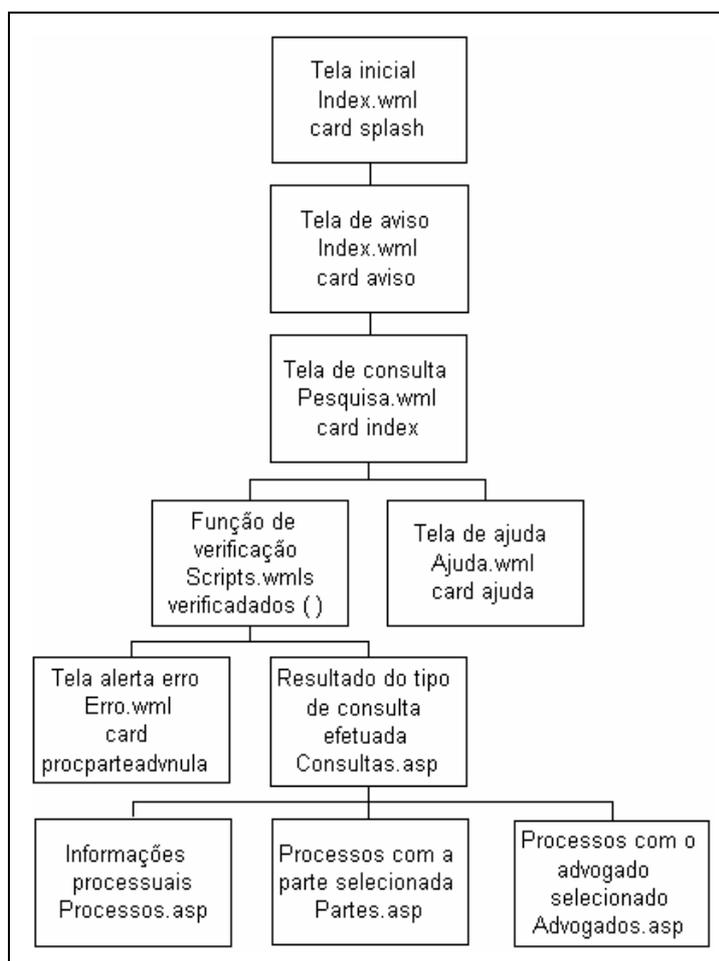
Figura 15 – Internet Explorer 6.0



3.4.2 FUNCIONAMENTO DOSISTEMA

A figura 16 apresenta os módulos implementados na forma de um macro-fluxo.

Figura 16 – Macro-fluxo do sistema



Os códigos-fonte da implementação foram listados no apêndice e as funcionalidades de cada módulo são apresentadas abaixo:

Tela inicial - é mostrada quando o usuário acessa o *site*. Contém um arquivo wbmp com o brasão símbolo do TJSC. Está implementada no *card splash* do arquivo index.wml.

Tela de aviso – aparece 1 segundo após a tela inicial, fornecendo um alerta aos usuários do *site* quanto o teor das informações obtidas. Está implementada no *card aviso* do arquivo index.wml.

Tela de consulta – possibilita ao usuário a escolha do tipo de pesquisa e a edição do conteúdo que será submetido para a mesma. Está implementada no *card* index do arquivo pesquisa.wml.

Tela de ajuda – contém instruções para o uso do *site*. Pode ser acessada a partir da tela de consulta. Está implementada no *card* ajuda do arquivo ajuda.wml.

Função de verificação – realizada para verificar se o conteúdo submetido à pesquisa não é vazio. Está implementada no arquivo scripts.wmls.

Tela alerta erro – é mostrada ao usuário caso o conteúdo submetido à pesquisa for vazio. Está implementada no *card* proparteadvnula do arquivo erro.wml.

Resultado do tipo de consulta efetuada – é gerado a partir da execução do arquivo consultas.asp quando os dados fornecidos na tela de consulta são submetidos. A partir deste resultado o usuário terá acesso às partes, advogados ou processos identificados com o valor submetido à pesquisa.

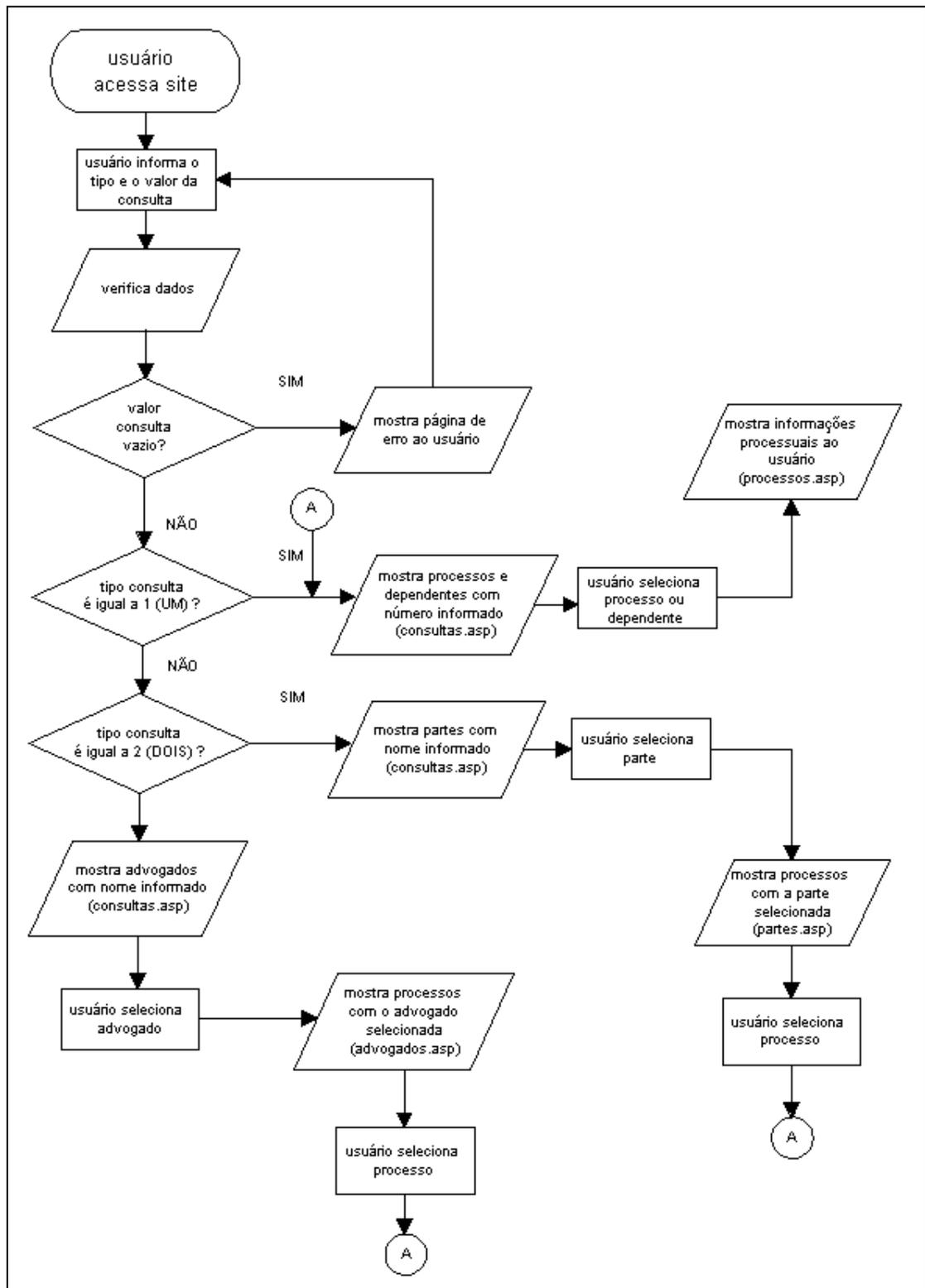
Informações processuais – são geradas a partir da execução do arquivo processos.asp quando o usuário seleciona um dos processos listados no resultado obtido do arquivo consultas.asp. A partir desta seleção o usuário tem acesso às informações processuais dos autos selecionados.

Processos com a parte selecionada – são listados a partir da execução do arquivo partes.asp quando o usuário seleciona uma das partes mostradas pelo resultado obtido do arquivo consultas.asp. A partir desta seleção o usuário tem acesso aos processos em que aquela parte está envolvida desde que esta não esteja sobre segredo de justiça.

Processos com o advogado selecionado – são listados a partir da execução do arquivo advogados.asp quando o usuário seleciona um dos advogados mostrados pelo resultado obtido do arquivo consultas.asp. A partir desta seleção o usuário tem acesso aos processos em que aquele advogado representa alguma parte envolvida desde que esta não esteja sobre segredo de justiça.

O fluxograma a seguir (figura 17) demonstra de maneira simplificada o funcionamento do sistema, a fim de possibilitar um melhor entendimento.

Figura 17 – Fluxograma do sistema



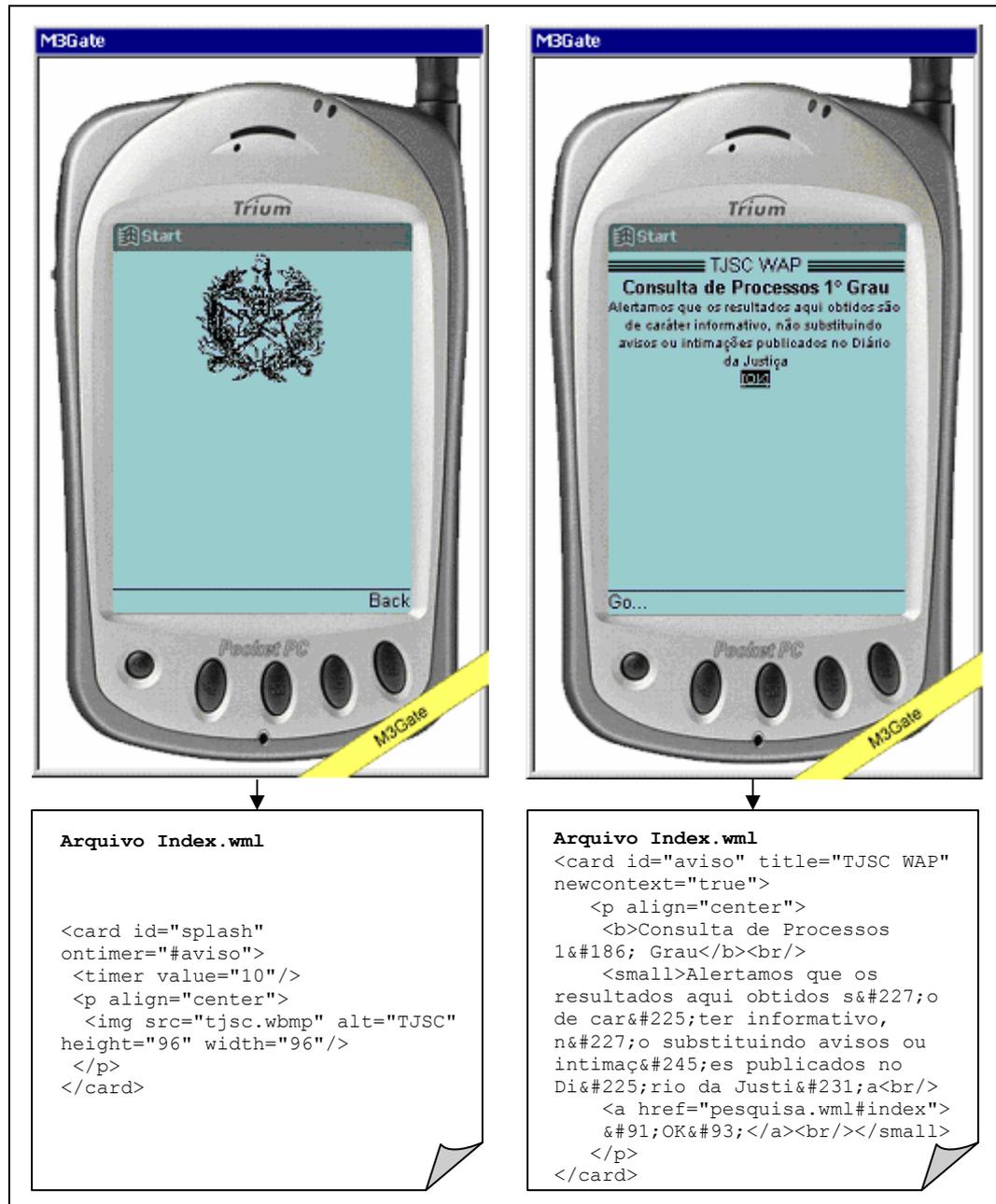
Como demonstrado no fluxograma acima, os caminhos da pesquisa a ser realizada são definidos pelo usuário, quando da escolha do seu tipo.

3.4.3 APRESENTAÇÃO DO SISTEMA

Neste item apresenta-se o sistema desenvolvido numa seqüência de utilização passo a passo.

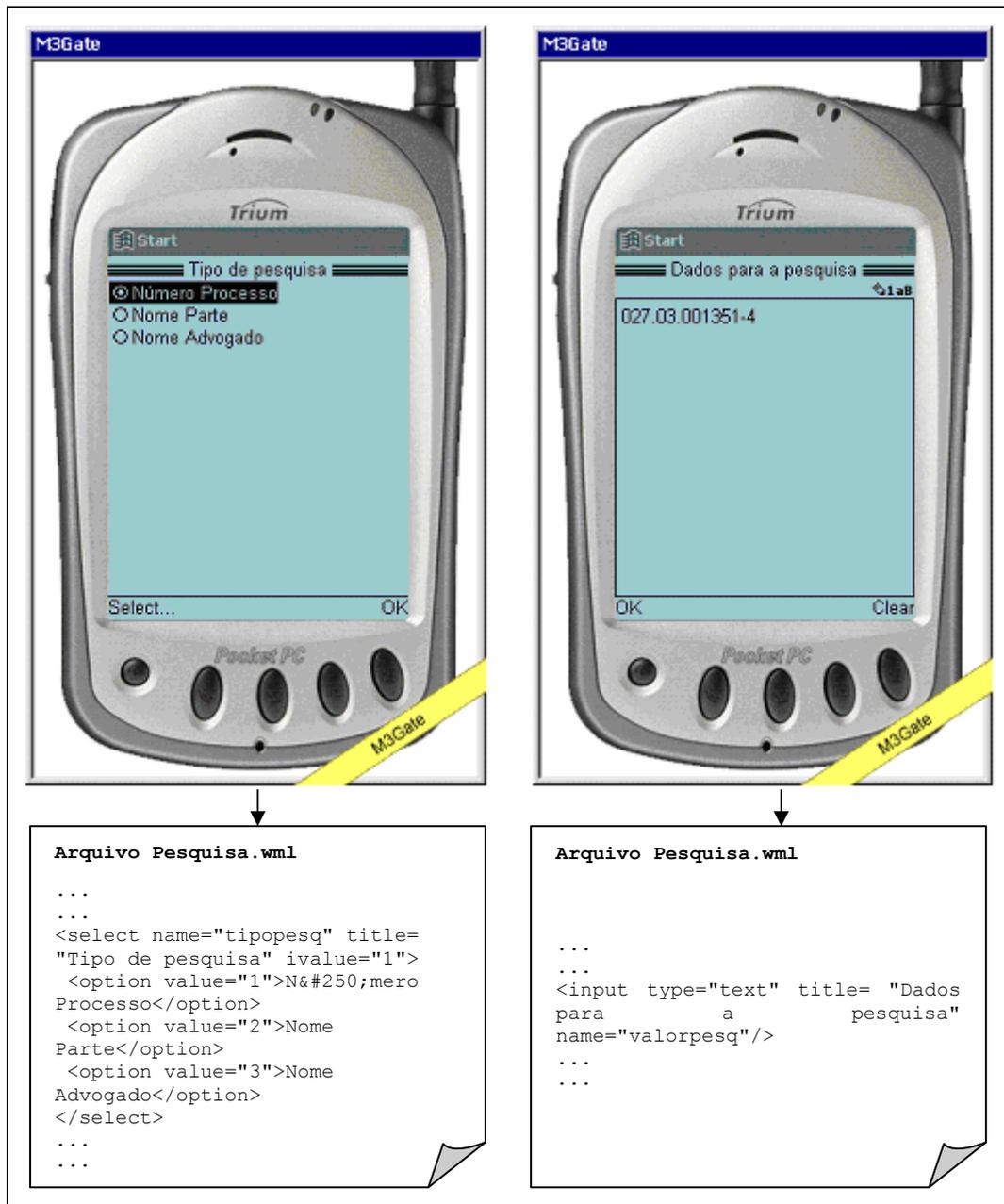
A seguir são apresentadas as telas do sistema e uma breve descrição destas.

Figura 18 – Tela inicial e tela de aviso



Já na tela de consulta (figura 19), o usuário encontrará duas informações a serem preenchidas. A primeira é utilizada para a escolha do tipo de pesquisa. Já a segunda é utilizada para editar o conteúdo que será submetido para a mesma. Nesta tela também existem *links* para limpar a tela, reiniciar o *site* e para acessar a ajuda.

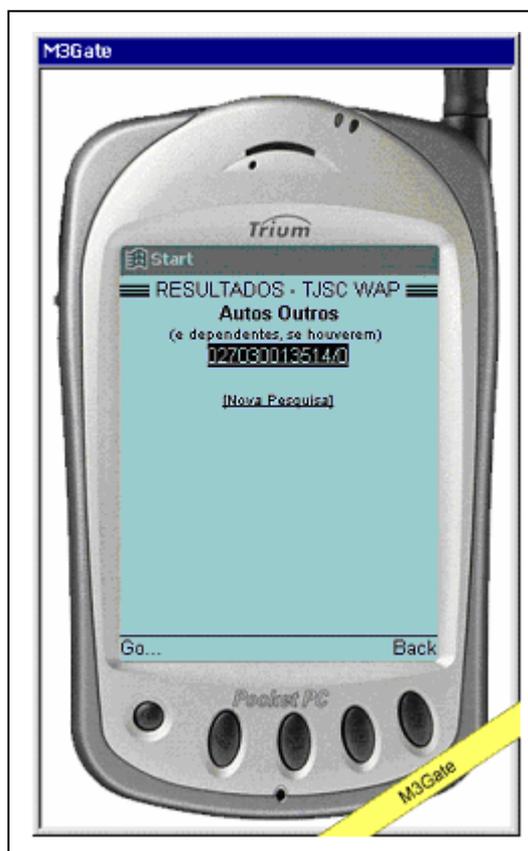
Figura 20 – Preenchimento dos campos para pesquisa



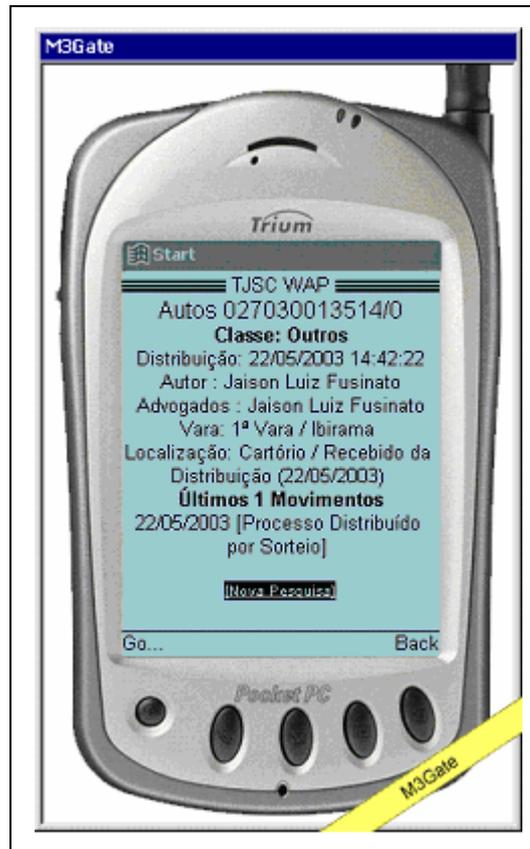
Ao clicar nos campos contidos na página de consulta, o usuário se depara com outras telas, que permitem a escolha do tipo de pesquisa e a edição do texto (conteúdo) que será

submetido à pesquisa. No exemplo contido na figura 20, a pesquisa escolhida foi por número de processo, passando como parâmetro o numero 027.03.001351-4.

Figura 21 – Resultado da consulta



Ao clicar no *link* pesquisar, o usuário submete a pesquisa ao processamento, recebendo como resposta uma nova página, em que encontrará a classe do processo principal solicitado, seu número e todos os seus dependentes (figura 21). Esta tela origina-se da execução do arquivo consultas.asp, que pode gerar diferentes respostas ao usuário, uma vez que é orientada pelo tipo de pesquisa selecionado.

Figura 22 – Obtenção das informações processuais

Ao clicar no número do processo desejado o usuário recebe como resposta as informações processuais daquele processo (figura 22). Esta tela é gerada a partir da execução do arquivo processos.asp. As informações fornecidas são o número e a classe do processo, data da distribuição ou do recebimento (no caso de recursos), as partes envolvidas e seus respectivos advogados, a localização do processo e até as três últimas movimentações.

4 CONCLUSÕES

Ao concluir este trabalho, percebeu-se que a sociedade de uma maneira geral, busca incessantemente a rapidez e a qualidade da informação.

O estudo e uso da tecnologia WAP associada com ASP se mostrou eficiente, apesar das limitações encontradas no desenvolvimento deste trabalho.

Apesar de ser considerada nova, a tecnologia WAP está se difundindo e logo tomará lugar de destaque no mercado, tendo em vista a união das empresas produtoras desta tecnologia, com objetivo de expandir este serviço.

Para a criação de soluções WAP como neste trabalho, exige-se além do estudo e entendimento do assunto, habilidade e imaginação, pois vários aspectos são comuns a um profissional da computação. Como exemplo, pode-se citar o fato de que todos os trabalhos envolvendo WAP desenvolvidos na FURB até aqui sugeriram a utilização do WMLScript para a validação dos dados submetidos pelo usuário, sendo que neste trabalho este recurso foi utilizado.

Pode-se afirmar que os objetivos deste trabalho foram alcançados. O sistema proposto foi desenvolvido e está pronto para ser adaptado às regras da Corregedoria e da Diretoria de Informática do TJSC para sua implantação definitiva, já que houve, por parte daqueles órgãos e por parte dos advogados da comarca de Ibirama-SC, aprovação unânime quanto à proposta de sua utilização para auxiliar o serviço de divulgação do TJSC.

Em relação às ferramentas utilizadas, pode-se afirmar que todas elas foram eficientes e de extrema importância no desenvolvimento deste trabalho.

Além disso, o presente trabalho propiciou o crescimento deste acadêmico nos aspectos profissionais, acadêmicos e até mesmo sociais, já que além do conhecimento e experiência obtidos, foram necessárias interações com várias pessoas a fim de sanar dúvidas, obter opiniões e pareceres.

As limitações citadas anteriormente são basicamente o tamanho máximo de 1400 bytes para as páginas WML e a dificuldade de se encontrar um *gateway* ASP para sistemas não Windows, neste caso, o Sun Solaris.

Apesar da boa performance do sistema, detectou-se ao utilizar o emulador M3Gate que, dependendo do *skin* escolhido, a visualização do *site* fica prejudicada em face do pequeno tamanho da tela de alguns dispositivos WAP. Esta constatação foi confirmada ao acessar o *site* de um celular Nokia da operadora TIM Celular.

4.1 EXTENSÕES

Como sugestões para trabalhos futuros, tem-se as seguintes possibilidades:

- a) buscar algum método ou técnica para suprir a deficiência do tamanho da tela dos dispositivos WAP, possibilitando assim a visualização correta do *site* em vários tipos de aparelhos;
- b) buscar soluções para a limitação do tamanho das páginas WML, tais como paginação na tecnologia ASP;
- c) buscar outras tecnologias para a criação de páginas dinâmicas, independente da plataforma utilizada.

REFERÊNCIAS

CANI, Giovani Endrigo. **Protótipo de software para acesso à informações baseado na tecnologia WAP (wireless application protocol)**. 2000. 88 f. Trabalho de Conclusão de Curso (Bacharelado em Ciências da Computação) - Centro de Ciências Exatas e Naturais, Universidade Regional de Blumenau, Blumenau.

DEMÉTRIO, Rinaldo. **A tecnologia W@P: aprenda a criar sites para celulares com a linguagem WML**. São Paulo: Érica, 2000.

DIAS, Adilson de Souza. **WAP: a internet sem fios**. Rio de Janeiro: Ciência Moderna, 2000.

DORMAN, Andy. **Wireless communication: o guia essencial da comunicação sem fio**. Tradução Fábio Freitas. Rio de Janeiro: Campus, 2001.

ORACLE CORPORATION. **Site com informações técnicas e documentação do banco de dados ORACLE**, Redwood Shores, CA, Estados Unidos da América, [2003?]. Disponível em: <<http://otn.oracle.com/products/oracle9i/content.html>>. Acesso em: 21 mar. 2003.

ROCHA, Leandro M. **Aprenda WAP**. São Paulo, SP, [2003?]. Disponível em: <<http://www.apredawap.com.br>>. Acesso em: 26 fev. 2003.

SANTA CATARINA (Estado). Lei Nº 5.624, de 9 de novembro de 1979. Dispõe sobre a adaptação do Código de Divisão e Organização Judiciárias do Estado de Santa Catarina à Lei Orgânica da Magistratura Nacional e dá outras providências. **Diário Oficial [do] Estado de Santa Catarina**, Florianópolis, SC, 12 nov. 1979. Disponível em: <<http://www.tj.sc.gov.br/institucional/normas/cdoj/indexdocdojsc.htm>>. Acesso em: 25 fev. 2003.

SANTA CATARINA (Estado). Tribunal de Justiça do Estado de Santa Catarina. Ato Regimental Nº 1, de 14 de novembro de 1982. Aprova o Regimento Interno do Tribunal de Justiça. **Diário da Justiça [do] Estado de Santa Catarina**, Florianópolis, SC, 25 nov. 1982. Disponível em: <<http://www.tj.sc.gov.br/institucional/normas/regimento/reginterno.htm>>. Acesso em: 25 fev. 2003.

SANTA CATARINA (Estado). Tribunal de Justiça do Estado de Santa Catarina. Resolução Nº 007/95 GP, de 5 de dezembro de 1995. Cria o Centro de Pesquisa e Aplicação de Informática - CEPAL. **Diário da Justiça [do] Estado de Santa Catarina**, Florianópolis, SC, 7 dez. 1995. Disponível em: <<http://presidencia.tj.sc.gov.br/documentos/resolucao/1995/gp/res0795gp.doc>>. Acesso em: 25 fev. 2003.

SANTA CATARINA (Estado). Tribunal de Justiça do Estado de Santa Catarina. Resolução Nº 04/00 GP, de 11 de fevereiro de 2000. Institui a Comissão de Gestão de Informatização - CGINFO. **Diário da Justiça [do] Estado de Santa Catarina**, Florianópolis, SC, 11 fev. 2000. Disponível em: <<http://presidencia.tj.sc.gov.br/documentos/resolucao/2000/gp/res0400gp.doc>>. Acesso em: 25 fev. 2003.

TJSC. **Código de normas da Corregedoria-Geral de Justiça**, Florianópolis, SC, 2002. Disponível em: <<http://www.tj.sc.gov.br/corregedoria/paginas/consultas/liberada/cnfj.rtf>>. Acesso em: 26 fev. 2003.

TJSC. **Serviço TJSCPush**, Florianópolis, SC, [2002?]. Disponível em: <<http://www.tj.sc.gov.br/consultas/tjscpush/index.htm>>. Acesso em: 26 fev. 2003.

TJSC. **Site de consulta processual**, Florianópolis, SC, [2003a?]. Disponível em: <<http://www.tj.sc.gov.br/consultas/comarcas.htm>>. Acesso em: 26 fev. 2003.

TJSC. **Notícias TJSC**, Florianópolis, SC, 2003b. Disponível em: <<http://www.tj.sc.gov.br/jsp/mostranoticia.jsp?CDNOTICIA=7400>>. Acesso em: 30 maio 2003.

WAP BRASIL. **Wap Brasil**, São Paulo, SP, [2003?]. Disponível em: <<http://www.wapbr.org>>. Acesso em: 25 fev. 2003.

WAPSITE. **Wapsite**, São Paulo, SP, [2003?]. Disponível em: <<http://www.wapsite.com.br>>. Acesso em: 26 fev. 2003.

WEISSINGER, A. Keyton. **ASP: guia completo**. Rio de Janeiro: Ciência Moderna, 1999.

WIRELESS APPLICATION PROTOCOL FORUM LTDA. **Official wireless application protocol.** New York: Wiley Computer Publishing, 1999.

WIRELESS APPLICATION PROTOCOL FORUM LTDA. **Official wireless application protocol.** New York, [2003?]. Disponível em: <<http://www.wapforum.org>>. Acesso em: 26 fev. 2003.


```

{
    var valorpp = (WMLBrowser.getVar("valorpesq")); // obtem da página que chamou o valor da
variável valorpesq
    // e atribui este valor à variável valorpp
    var tipopp = (WMLBrowser.getVar("tipopesq")); // obtem da página que chamou o valor da
variável tipopesq
    // e atribui este valor à variável tipopp
    if (valorpp != "") // verifica se o valor que foi submetido à pesquisa não é vaziao
    {
        // se não for vaziao, atribui o caminho do arquivo consultas.asp junto com os valores das
variáveis acima
        // a fim de montar um comando post àquele arquivo ASP
        var urlconsulta = "http://servidordadosibirama/consultas.asp?tipopp=" + tipopp +
"&valorpp=" + valorpp
        // submete os valores que estão contidos naquele comando ao servidor de dados da comarca de
Ibirama
        // onde se encontram os arquivos ASP's para obtenção das informações processuais
        WMLBrowser.go(urlconsulta);
    }
    else
    {
        // se for vaziao, mostra ao usuário a mensagem de erro contida no arquivo erro.wml
        WMLBrowser.go("erro.wml");
    }
};

```

Consultas.asp

```

<% Response.ContentType = "text/vnd.wap.wml" %><?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1//EN" "http://www.wapforum.org/DTD/wml_1.1.xml">
<wml>
<card id="consultas" title="RESULTADOS - TJSC WAP">
<p align="center">
<%
' variáveis utilizadas
Dim conn
Dim rsdados
Dim codigo
Dim tipopesquisa
Dim valorpesq
Dim letras
Dim novovalorpesq
Dim valorformatado
' atribui à variável tipopesquisa o valor da variável tipopp passada como parâmetro
tipopesquisa = Request.QueryString("tipopp")
' caracteres de conexão ao banco de dados
set conn=server.createobject("adodb.connection")
conn.open "Provider=MSDAORA.1;Data Source=wap;User id=wap;Password=wap;Persist Security
Info=True"
set rsdados=server.createobject("adodb.recordset")
' faz uma seleção especifica ao tipo de pesquisa
Select Case tipopesquisa
Case 1
' se o tipo de pesquisa for 1 (por número do processo) entra nesta opção
' atribui à variável valorpesq o valor obtido da variável valorpp passada como parâmetro
valorpesq = Request.QueryString("valorpp")
' formata o valor obtido, pois o mesmo pode estar no formato 027.XX.XXXXXX-X/XXX
novovalorpesq = ""
For i=1 to len(valorpesq)
    letras = mid(valorpesq, i, 1)
    Select Case letras
        ' percorre a string da variável valorpesq retirando pontos, barras, hífens e vírgulas
        Case "." , "/" , "-" , ","
            novovalorpesq = novovalorpesq
        Case else
            novovalorpesq = novovalorpesq & letras
    End Select
Next
valorformatado = ""
For i=1 to len(novovalorpesq)
    letras = mid(novovalorpesq, i, 1)
    ' percorre o novovalorpesq que foi a formatação anterior e retira a parte /XXX
    If i > 12 then

```

```

        valorformatado = valorformatado
    else
        valorformatado = valorformatado & letras
    End if
Next
' neste passo estamos prontos para submeter o valor ao select
' seleciona todos os processos que existem com aquele número, inclusive os incidentes
processuais
set rsdados=conn.execute("select nrproc, cdproc, nrrecurso from recurso where nrproc =
"&valorformatado&" order by nrproc")
if rsdados.eof then
    ' caso não exista(m) processo(s) com aquele número mostra esta mensagem
    response.write "<b>N&#227;o h&#225; processos com os par&#226;metros
informados&#33;&#33;&#33;</b><br/><small>Verifique se o número do processo foi digitado
corretamente.</small>"
else
    ' fecha o recordset rsdados para utilizá-lo em outro select
    rsdados.close
    set rsdados=nothing
    ' se existir processos com aquele número, mostra o nome do processo principal
    set rsdados=conn.execute("select classeproc from processo where nrproc =
"&valorformatado&")
    response.write "<b>Autos "& rsdados.Fields("classeproc") & "</b><br/><small>(e
dependentes, se houverem) </small><br/>"
    ' fecha o recordset rsdados para utilizá-lo em outro select
    rsdados.close
    set rsdados=nothing
    ' mostra também todos os números e dependentes daquele processo
    set rsdados=conn.execute("select nrproc, cdproc, nrrecurso from recurso where nrproc =
"&valorformatado&" order by nrproc")
    while not rsdados.eof
        ' mostra um link para o arquivo processos.asp passando como parâmetro o código do
processo selecionado
        response.write "<a href=""http://servidordadosibirama/processos.asp?codigo=" &
rsdados.Fields("cdproc") & "">" & rsdados.Fields("nrproc") & "/" &
rsdados.Fields("nrrecurso") & "</a><br/>"
        rsdados.movenext
    wend
end if
Case else
    ' caso o tipo de pesquisa seja diferente de 1 então entra neste laço
    ' se o tipo de pesquisa for 3 entra neste laço
    if tipospesquisa = 3 then
        ' seleciona todos advogados que tem nome contendo a string (contida em valorpp) passada
como parâmetro
        ' e cujo código da profissão (tabela qualificação) seja igual 8 (advogado)
        set rsdados=conn.execute("select nmpessoa, cdpessoa from nomeparteadv where seqnome = 0
and nmpessoa like '%"&Request.QueryString("valorpp")&"%' and cdpessoa in (select cdpessoa from
qualificacaoparteadv where cdprofissao = 8) order by nmpessoa")
        if rsdados.eof then
            ' caso não existam advogados com aquele nome, mostra esta mensagem
            response.write "<b>N&#227;o h&#225; advogados com o nome
informado&#33;&#33;&#33;</b><br/><small>Alertamos que para este tipo de pesquisa não se deve
utilizar acentos. Verifique também o uso correto das letras maiúsculas/minúsculas.</small>"
        else
            ' se existirem advogados selecionadas mostra todos
            response.write "<b>Advogados(as) selecionados(as):</b><br/>"
            while not rsdados.eof
                ' mostra um link para o arquivo advogados.asp passando o código do advogado selecionado como
parâmetro
                response.write "<a href=""http://servidordadosibirama/advogados.asp?codigo=" &
rsdados.Fields("cdpessoa") & "">" & rsdados.Fields("nmpessoa") & "</a><br/>"
                rsdados.movenext
            wend
        end if
    else
        ' se o tipo de pesquisa não for 3 (será 2) então entra neste laço
        ' seleciona todas as partes que tem nome contendo a string (contida em valorpp) passada
como parâmetro
        set rsdados=conn.execute("select nmpessoa, cdpessoa from nomeparteadv where seqnome = 0
and nmpessoa like '%"&Request.QueryString("valorpp")&"%' order by nmpessoa")
        if rsdados.eof then
            ' caso não existam partes com aquele nome, mostra esta mensagem

```

```

        response.write      "<b>N&#227;o      h&#225;      partes      com      o      nome
informado&#33;&#33;&#33;</b><br/><small>Alertamos que para este tipo de pesquisa não se deve
utilizar acentos. Verifique também o uso correto das letras maiúsculas/minúsculas.</small>"
    else
        ' se existirem partes selecionadas mostra todas
        response.write "<b>Partes selecionadas:</b><br/>"
        while not rsdados.eof
            ' mostra um link para o arquivo partes.asp passando o código da parte selecionada como
parâmetro
            response.write      "<a      href=""http://servidordadosibirama/partes.asp?codigo="      &
rsdados.Fields("cdpessoa") & "">" & rsdados.fields("nmpessoa") & "</a><br/>"
            rsdados.movenext
        wend
    end if
end if
End select
' fecha a conexão com o banco de dados
rsdados.close
set rsdados=nothing
conn.close
set conn=nothing
' e abaixo mostra um link para a página de consulta
%>
</p>
<p align="center">
    <small>
    <br/>
    <a href="pesquisa.wml#index">&#91;Nova Pesquisa&#93;</a><br/></small>
</p>
</card>
</wml>

```

Partes.asp

```

<% Response.ContentType = "text/vnd.wap.wml" %><?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1//EN" "http://www.wapforum.org/DTD/wml_1.1.xml">
<wml>
<card id="consultas" title="RESULTADOS - TJSC WAP">
    <p align="center">
        <%
        ' variáveis utilizadas
        Dim conn
        Dim rsdados
        Dim tipopp
        Dim valorpp
        Dim contador
        ' caracteres de conexão ao banco de dados
        set conn=server.createobject("adodb.connection")
        conn.open "Provider=MSDAORA.1;Data Source=wap;User id=wap;Password=wap;Persist Security
Info=True"
        set rsdados=server.createobject("adodb.recordset")
        ' executa a seleção de todos os processos com o código da parte informado como parâmetro
        set rsdados=conn.execute("select distinct nrproc from procparte where cdparte =
"&Request.QueryString("codigo")&""")
        ' testa se existem processos selecionados
        if rsdados.eof then
            ' se não tem nenhum processo com determinada parte, mostra esta mensagem
            response.write "<br/><b>N&#227;o      h&#225;      processos      com      os      par&#226;metros
informados&#33;&#33;&#33;</b>"
        else
            ' senão mostra todos os processos com exceção dos que estão em segredo de justiça, ordenando
pelo número
            response.write "<b>Processos com a parte selecionada:</b><br/>"
            set rsdados=conn.execute("select distinct nrproc from procparte where segredojustica = 0
and cdparte = "&Request.QueryString("codigo")&"" order by nrproc")
            while not rsdados.eof
                ' exhibe os processos selecionados com link para o arquivo consultas.asp localizado no
servidor de dados da comarca de Ibirama para obtenção das informações processuais
                response.write
                <a href=""http://servidordadosibirama/consultas.asp?tipopp=1&#38;valorpp="
                &
rsdados.Fields("nrproc") & "">" & "0" & rsdados.fields("nrproc") & "</a><br/>"
                rsdados.movenext
            wend
        end if
        %>
    </p>
</card>
</wml>

```

```

' seleciona todos os processos que existem em segredo de justiça
set rsdados=conn.execute("select distinct nrproc from procparte where cdparte =
"&Request.QueryString("codigo")&" and segredojustica = 1 ")
if rsdados.eof then
else
' conta quantos processos existem em segredo de justiça
contador= 0
while not rsdados.eof
contador = contador+1
rsdados.movenext
wend
' mostra quantos processos existem em segredo de justiça
response.write "<small>Existe(m) "& contador &" processo(s) em Segredo de
Justiça.</small><br/> "
end if
end if
' fecha a conexão com o servidor de dados
rsdados.close
set rsdados=nothing
conn.close
set conn=nothing
' e abaixo mostra um link para a página de pesquisa
%>
</p>
<p align="center">
<small>
<br/>
<a href="pesquisa.wml#index">&#91;Nova Pesquisa&#93;</a><br/></small>
</p>
</card>
</wml>

```

Advogados.asp

```

<% Response.ContentType = "text/vnd.wap.wml" %><?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1//EN" "http://www.wapforum.org/DTD/wml_1.1.xml">
<wml>
<card id="consultas" title="RESULTADOS - TJSC WAP">
<p align="center">
<%
' variáveis utilizadas
Dim conn
Dim rsdados
Dim tipopp
Dim valorpp
Dim contador
' caracteres de conexão ao banco de dados
set conn=server.createobject("adodb.connection")
conn.open "Provider=MSDAORA.1;Data Source=wap;User id=wap;Password=wap;Persist Security
Info=True"
set rsdados=server.createobject("adodb.recordset")
' executa a seleção de todos os processos com o código do advogado informado como parâmetro
set rsdados=conn.execute("select distinct nrproc from procparteadv where cdadv =
"&Request.QueryString("codigo")&" order by nrproc")
' testa se existem processos selecionados
if rsdados.eof then
' se não tem nenhum processo com determinado advogado, mostra esta mensagem
response.write "<br/><b>N&#227;o h&#225; processos com os par&#226;metros
informados&#33;&#33;&#33;</b>"
else
' senão mostra todos os processos com exceção dos que estão em segredo de justiça,
ordenando pelo número
response.write "<b>Processos com o(a) advogado(a) selecionado(a):</b><br/>"
set rsdados=conn.execute("select nrproc from procparteadv where cdadv =
"&Request.QueryString("codigo")&" and nrproc in (select distinct nrproc from procparte where
segredojustica=0) order by nrproc")
while not rsdados.eof
' exibe os processos selecionados com link para o arquivo consultas.asp localizado no
servidor de dados da comarca de Ibirama para obtenção das informações processuais
response.write "<a
href=""http://servidordadosibirama/consultas.asp?tipopp=1&#38;valorpp="
&
rsdados.Fields("nrproc") & "">" & "0" & rsdados.fields("nrproc") & "</a><br/>"
rsdados.movenext

```

```

wend
' seleciona todos os processos que existem em segredo de justiça
set rsdados=conn.execute("select nrproc from procparteadv where cdadv =
"&Request.QueryString("codigo")&" and nrproc in (select distinct nrproc from procparte where
segredojustica=1) order by nrproc")
if rsdados.eof then
else
' conta quantos processos existem em segredo de justiça
contador= 0
while not rsdados.eof
contador = contador+1
rsdados.movenext
wend
' mostra quantos processos existem em segredo de justiça
response.write "<small>Existe(m) "& contador &" processo(s) em Segredo de
Justiça.</small> <br/> "
end if
end if
' fecha a conexão com o servidor de dados
rsdados.close
set rsdados=nothing
conn.close
set conn=nothing
' e abaixo mostra um link para a página de pesquisa
%>
</p>
<p align="center">
<small>
<br/>
<a href="pesquisa.wml#index">&#91;Nova Pesquisa&#93;</a><br/></small>
</p>
</card>
</wml>

```

Processos.asp

```

<% Response.ContentType = "text/vnd.wap.wml" %><?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1//EN" "http://www.wapforum.org/DTD/wml_1.1.xml">
<wml>
<card id="resultados" title="TJSC WAP">
<p align="center">
<%
' variáveis utilizadas
Dim conn
Dim rsdados
Dim rsauxiliar
Dim nrrecurso
Dim nrprocesso
Dim contador
Dim tipoparte
Dim tipomovimento
' caracteres de conexão ao banco de dados
set conn=server.createobject("adodb.connection")
conn.open "Provider=MSDAORA.1;Data Source=wap;User id=wap;Password=wap;Persist Security
Info=True"
set rsdados=server.createobject("adodb.recordset")
' obtem da tabela recurso o número do processo e do recurso para posterior consulta em
outras tabelas
set rsdados=conn.execute("select * from recurso where cdproc =
"&Request.QueryString("codigo")&"')
' exibe o número do processo / recurso em fontes grandes
response.write "<big>Autos 0" & rsdados.fields("nrproc") & "/" & rsdados.fields("nrrecurso")
& "</big><br/>"
' atribui os atributos da tabela recurso às variáveis abaixo
nrrecurso = rsdados.fields("nrrecurso")
nrprocesso = rsdados.fields("nrproc")
' fecha o recordset rsdados para usá-lo em outro select
rsdados.close
set rsdados=nothing
' obtem da tabela processo a classe do processo e exibe
set rsdados=conn.execute("select * from processo where cdproc =
"&Request.QueryString("codigo")&"')
response.write "<b>Classe: " & rsdados.fields("classeproc") & "</b><br/>"

```

```

' fecha o recordset rsdados para usá-lo em outro select
rsdados.close
set rsdados=nothing
' verifica se o recurso é zero
if nrrecurso = "0" then
' se for zero ele é o processo principal, então terá que obter a data da distribuição para
exibir
set rsdados=conn.execute("select * from distribuicao where nrproc = "&nrprocesso&" and
ultimadist = 1")
response.write "Distribuição: " & rsdados.fields("datadist") & "<br/>"
else
' senão for zero ele é processo dependente ou incidente processual, então terá que obter a
data de recebimento
set rsdados=conn.execute("select * from recurso where cdproc =
"&Request.QueryString("codigo")&""")
response.write "Recebimento: " & rsdados.fields("datarecebimento") & "<br/>"
end if
' fecha o recordset rsdados para usá-lo em outro select
rsdados.close
set rsdados=nothing
' seleciona os tipos de partes existentes no processo afim de usá-las no laço abaixo
set rsdados=conn.execute("select distinct cdtipoparte from procparteadv where nrproc =
"&nrprocesso&" and nrrecurso="&nrrecurso&" order by cdtipoparte")
' entra num laço passando todas as partes, distinguindo-as por tipo a fim de obter seu nome
e advogado
while not rsdados.eof
tipoparte = rsdados.fields("cdtipoparte")
' usa um recordset auxiliar para obter o código das partes
set rsauxiliar=conn.execute("select distinct cdparte from procparteadv where cdtipoparte =
"&tipoparte&" and nrproc = "&nrprocesso&" and nrrecurso="&nrrecurso&""")
' conta quantas partes tem o processo
contador = 0
while not rsauxiliar.eof
contador = contador+1
rsauxiliar.movenext
wend
' fecha o recordset rsauxiliar para usá-lo em outro select
rsauxiliar.close
set rsauxiliar=nothing
' verifica se a quantidade de partes envolvidas com o mesmo tipo (autor, réu, etc) é maior
que 1
if contador > 1 then
' se for maior, busca o nome do tipo de parte no plural e mostra
set rsauxiliar=conn.execute("select plural from tipoparteadv where cdtipoparteadv =
(select distinct cdtipoparte from procparteadv where cdtipoparte = "&tipoparte&" and nrproc =
"&nrprocesso&" and nrrecurso="&nrrecurso&")")
response.write rsauxiliar.fields("plural") & " : "
' fecha o recordset rsauxiliar para usá-lo em outro select
rsauxiliar.close
set rsauxiliar=nothing
' seleciona o nome da parte da tabela nomeparteadv desde que a seqüência seja zero
(própria pessoa)
set rsauxiliar=conn.execute("select nmpessoa from nomeparteadv where seqnome = 0 and
cdpessoa in (select distinct cdparte from procparteadv where cdtipoparte = "&tipoparte&" and
nrproc = "&nrprocesso&" and nrrecurso="&nrrecurso&") order by nmpessoa")
' mostra a primeira parte (ordem alfabética) mais as palavras "e outro(s)"
response.write rsauxiliar.fields("nmpessoa") & " e outro(s)<br/> "
' fecha o recordset rsauxiliar para usá-lo em outro select
rsauxiliar.close
set rsauxiliar=nothing
' busca o nome do tipo de advogado no plural e mostra
set rsauxiliar=conn.execute("select plural from tipoparteadv where cdtipoparteadv =
(select distinct cdtipoadv from procparteadv where cdtipoparte = "&tipoparte&" and nrproc =
"&nrprocesso&" and nrrecurso="&nrrecurso&")")
response.write rsauxiliar.fields("plural") & " : "
' fecha o recordset rsauxiliar para usá-lo em outro select
rsauxiliar.close
set rsauxiliar=nothing
' seleciona os nomes dos advogado da tabela nomeparteadv desde que a seqüência seja zero
(própria pessoa)
set rsauxiliar=conn.execute("select nmpessoa from nomeparteadv where seqnome = 0 and
cdpessoa in (select distinct cdadv from procparteadv where cdtipoparte = "&tipoparte&" and
nrproc = "&nrprocesso&" and nrrecurso="&nrrecurso&") order by nmpessoa")
' mostra todos os advogados daquela parte

```

```

while not rsauxiliar.eof
  response.write rsauxiliar.fields("nmpessoa") & "<br/> "
  rsauxiliar.movenext
wend
' fecha o recordset rsauxiliar para usá-lo em outro select
rsauxiliar.close
set rsauxiliar=nothing
else
' se a quantidade de partes envolvidas com o mesmo tipo (autor, réu, etc) é menor ou
igual a 1
' seleciona o nome to tipo da parte e mostra
set rsauxiliar=conn.execute("select singular from tipoparteadv where cdtipoparteadv =
(select distinct cdtipoparte from procparteadv where cdtipoparte = "&tipoparte&" and nrproc =
"&nrprocesso&" and nrrecurso="&nrrecurso&")")
response.write rsauxiliar.fields("singular") & " : "
' fecha o recordset rsauxiliar para usá-lo em outro select
rsauxiliar.close
set rsauxiliar=nothing
' seleciona o nome da parte da tabela nomeparteadv desde que a seqüência seja zero
(própria pessoa)
set rsauxiliar=conn.execute("select nmpessoa from nomeparteadv where seqnome = 0 and
cdpessoa in (select distinct cdparte from procparteadv where cdtipoparte = "&tipoparte&" and
nrproc = "&nrprocesso&" and nrrecurso="&nrrecurso&") order by nmpessoa")
response.write rsauxiliar.fields("nmpessoa") & "<br/> "
' fecha o recordset rsauxiliar para usá-lo em outro select
rsauxiliar.close
set rsauxiliar=nothing
' seleciona o nome do tipo de advogado daquela parte e mostra
set rsauxiliar=conn.execute("select plural from tipoparteadv where cdtipoparteadv =
(select distinct cdtipoadv from procparteadv where cdtipoparte = "&tipoparte&" and nrproc =
"&nrprocesso&" and nrrecurso="&nrrecurso&")")
response.write rsauxiliar.fields("plural") & " : "
' fecha o recordset rsauxiliar para usá-lo em outro select
rsauxiliar.close
set rsauxiliar=nothing
' seleciona os nomes dos advogado da tabela nomeparteadv desde que a seqüência seja zero
(própria pessoa)
set rsauxiliar=conn.execute("select nmpessoa from nomeparteadv where seqnome = 0 and
cdpessoa in (select distinct cdadv from procparteadv where cdtipoparte = "&tipoparte&" and
nrproc = "&nrprocesso&" and nrrecurso="&nrrecurso&") order by nmpessoa")
' mostra todos os advogados daquela parte
while not rsauxiliar.eof
  response.write rsauxiliar.fields("nmpessoa") & "<br/> "
  rsauxiliar.movenext
wend
' fecha o recordset rsauxiliar para usá-lo em outro select
rsauxiliar.close
set rsauxiliar=nothing
end if
' passa para a próxima parte envolvida no processo, para mostras as mesmas informações
rsdados.movenext
wend
' fecha o recordset rsdados para usá-lo em outro select
rsdados.close
set rsdados=nothing
' seleciona o nome da vara onde se encontra o processo e mostra
set rsdados=conn.execute("select nomevara from vara where cdvara = (select cdvara from
distribuicaooproc where nrproc = "&nrprocesso&" and ultimadist =1)")
response.write "Vara: " & rsdados.fields("nomevara") & " / Ibirama <br/>"
' fecha o recordset rsdados para usá-lo em outro select
rsdados.close
set rsdados=nothing
' seleciona o nome do local físico onde se encontra o processo e mostra
set rsdados=conn.execute("select local from localfisico where cdlocal = (select cdlocal from
recurso where cdproc = '&Request.QueryString("codigo")&')")
response.write "Localização: " & rsdados.fields("local")
' fecha o recordset rsdados para usá-lo em outro select
rsdados.close
set rsdados=nothing
' seleciona a data e o complemento da localização em que o processo foi localizado e mostra
set rsdados=conn.execute("select datalocalizacao, complocalizacao from recurso where cdproc
= '&Request.QueryString("codigo")&' ")
response.write " (" & rsdados.fields("datalocalizacao") & " ) " &
rsdados.fields("complocalizacao") & "<br/>"

```

```

' fecha o recordset rsdados para usá-lo em outro select
rsdados.close
set rsdados=nothing
' seleciona todas as movimentações do processo ordenando em ordem decrescente de data de
movimento
set rsdados=conn.execute("select * from processomv where nrproc = "&nrprocesso&" and
nrrecurso="&nrrecurso&" order by datamv desc")
' conta quantos movimentos tem o processo
contador=0
while not rsdados.eof
  contador = contador +1
  rsdados.movenext
wend
' limita em até três as movimentações a serem mostradas (até 3 últimas)
if contador > 3 then
  contador =3
end if
rsdados.movefirst
response.write "<b>Últimos " & contador & " Movimentos</b><br/>"
For i=1 to contador
  ' mostra a data do movimento
  response.write rsdados.fields("datamv") & " ["
  tipomovimento = rsdados.fields("cdtipomv")
  ' mostra o tipo de movimentação
  set rsauxiliar=conn.execute("select tipomv from tipomvproc where cdtipomv =
'&tipomovimento&'")
  response.write rsauxiliar.fields("tipomv") & "]"
  ' mostra o complemento da movimentação
  response.write rsdados.fields("complementomv") & "<br/> "
  rsdados.movenext
Next
' fecha os recordsets rsauxiliar e rsdados e a conexão com o servidor de dados
rsauxiliar.close
set rsauxiliar=nothing
rsdados.close
set rsdados=nothing
conn.close
set conn=nothing
' e abaixo mostra um link para a página de pesquisa
%>
</p>
<p align="center">
  <small>
    <br/>
    <a href="pesquisa.wml#index">&#91;Nova Pesquisa&#93;</a><br/></small>
  </p>
</card>
</wml>

```