

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

**PROTÓTIPO DE SISTEMA DE APOIO A GESTÃO PARA CORRETORA DE
SEGUROS**

**RELATÓRIO DE ESTÁGIO SUPERVISIONADO DE CURSO SUBMETIDO À
UNIVERSIDADE REGIONAL DE BLUMENAU PARA A OBTENÇÃO DOS
CRÉDITOS DE DISCIPLINA COM O NOME EQUIVALENTE NO CURSO DE
CIÊNCIAS DA COMPUTAÇÃO - BACHARELADO**

VALDA STANGE

BLUMENAU, DEZEMBRO/2000

2000/2-55

PROTÓTIPO DE SISTEMA DE APOIO A GESTÃO PARA CORRETORA DE SEGUROS

VALDA STANGE

ESTE RELATÓRIO DE ESTÁGIO SUPERVISIONADO DE CURSO FOI JULGADO ADEQUADO PARA OBTENÇÃO DOS CRÉDITOS DA DISCIPLINA DO TRABALHO DE CONCLUSÃO DE CURSO, OBRIGATÓRIA PARA A OBTENÇÃO DO TÍTULO DE:

BACHAREL EM CIÊNCIAS DA COMPUTAÇÃO

Prof. Wilson Pedro Carli - Orientador na Furb

Prof. José Roque Voltolini da Silva - Coordenador do Estágio

Walcir Augusto Werhle - Supervisor na Empresa

BANCA EXAMINADORA

Prof. Wilson Pedro Carli

Prof. Maurício Capobianco Lopes

Prof. Dalton Solano dos Reis

DEDICATÓRIA

Aos meus pais por terem acreditado na possibilidade de realização deste. Às minhas irmãs por me incentivarem ao término do curso. Aos meus colegas de trabalho e do curso que me animaram e me incentivaram a persistir. Ao Sr. Rafael e Sra. Olinda, por terem me acolhido nesta cidade no início do curso, e também, à Carlos Stange, por ter apostado na concretização deste sonho que até então era meu.

AGRADECIMENTOS

Agradeço a todas as pessoas que colaboraram para a conclusão deste trabalho, onde muitos contribuíram direta ou indiretamente, em especial o professor e orientador Wilson Pedro Carli, a Valcir Augusto Wherle meu supervisor na empresa onde realizei o estágio.

SUMÁRIO

	LISTA DE QUADROS	vii
	LISTA DE FIGURAS	viii
	LISTA DE SIGLAS	ix
	RESUMO	x
	ABSTRACT	xi
1	INTRODUÇÃO	1
1.1	Objetivos	4
1.2	Organização	4
2	SEGUROS	6
2.1	Sociedades seguradoras	7
2.2	Termos técnicos	8
2.3	Corretores de seguros	10
3	INTERNET INFORMATION SERVER (IIS).....	12
3.1	Integração com Windows NT Server	12
3.2	Publicação e Compartilhamento de informações	13
3.3	Criação e execução de aplicativos da Web	13
3.4	Gerenciamento de sites da Web.....	14
3.5	Vantagens de performance e usabilidade sobre outros servidores	14
4	A LINGUAGEM PHP	16
4.1	Estrutura de um programa PHP	17
4.2	Resultado na Web	19
4.3	Comentários no PHP	20
4.4	Variáveis no PHP	21
4.5	Operações matemáticas	25
4.6	Operações com strings	26

4.7	Formulários HTML	27
4.8	Funções do PHP	28
5	FERRAMENTAS	33
5.1	SQL Server	33
5.2	Conectando um Banco de Dados	34
6	PROTÓTIPO	37
6.1	Levantamento de informações	37
6.2	Especificação	38
6.3	Características	38
6.3.1	Diagrama de contexto.....	38
6.4	Modelo entidade relacionamento (MER)	41
7	IMPLEMENTAÇÃO DO PROTÓTIPO	43
8	CONCLUSÃO	62
8.1	Limitações e problemas	63
8.2	Sugestões	63
	GLOSSÁRIO	64
	ANEXO 1 - ENTRADA DA CHAVE DE LOCALIZAÇÃO.....	67
	ANEXO 2 - RESULTADO DA PESQUISA POR CPF/CNPJ.....	68
	ANEXO 3 - DETALHAMENTO DA APÓLICE.....	69
	REFERÊNCIAS BIBLIOGRÁFICAS	70

LISTA DE QUADROS

Quadro 1:	Estrutura de programa PHP	18
Quadro 2:	Resultado do programa PHP - Exemplo 1.....	19
Quadro 3:	Resultado do programa PHP - Exemplo 2.....	20
Quadro 4:	Resultado do programa PHP - Exemplo 3.....	20
Quadro 5:	Comentários no PHP - Exemplo 1.....	21
Quadro 6:	Comentários no PHP - Exemplo 2.....	21
Quadro 7:	Variáveis no PHP.....	21
Quadro 8:	Uso do modificador <i>static</i> no PHP - Exemplo 1.....	22
Quadro 9:	Uso do modificador <i>static</i> no PHP - Exemplo 2.....	22
Quadro 10:	Operações matemáticas.....	25
Quadro 11:	Sintaxe do operador <i>strlen</i>	26
Quadro 12:	Sintaxe do operador <i>substr</i>	26
Quadro 13:	Operações com <i>strings</i>	26
Quadro 14:	Formulários HTML.....	27
Quadro 15:	Sintaxe básica para definir uma função.....	28
Quadro 16:	Funções do PHP - Exemplo 1.....	29
Quadro 17:	Funções do PHP - Exemplo 2.....	30
Quadro 18:	Funções do PHP - Exemplo 3.....	30
Quadro 19:	Funções do PHP - Exemplo 4.....	30
Quadro 20:	Funções do PHP - Exemplo 5.....	31
Quadro 21:	Funções do PHP - Exemplo 6.....	31
Quadro 22:	Funções do PHP - Exemplo 7.....	32

LISTA DE FIGURAS

Figura 1:	Diagrama de Contexto.....	38
Figura 2:	Diagrama de Fluxo de Dados Nível 0 (cliente).....	39
Figura 3:	Diagrama de Fluxo de Dados Nível 0 (corretor).....	40
Figura 4:	Diagrama de Fluxo de Dados Nível 0 (gerência).....	40
Figura 5:	Modelo Entidade Relacionamento (MER).....	41
Figura 6:	Identificação.....	43
Figura 7:	Entrada da chave de localização.....	44
Figura 8:	Resultado da pesquisa por CPF/CNPJ.....	44
Figura 9:	Detalhamento da apólice.....	45
Figura 10:	Consultas gerenciais.....	46
Figura 11:	Porcentagem de vendas do corretor - tela 1.....	47
Figura 12:	Porcentagem de vendas do corretor - tela 2.....	48
Figura 13:	Ramos de seguros de maior risco.....	49
Figura 14:	Porcentagem de seguros renovados.....	50
Figura 15:	Porcentagem de seguros cancelados.....	51
Figura 16:	Porcentagem de vendas de cada seguradora.....	52
Figura 17:	Porcentagem de apólices.....	53
Figura 18:	Porcentagem de cada ramo vendido para cada seguradora.....	54
Figura 19:	Porcentagem de cada ramo vendido pelo corretor - Tela 1.....	55
Figura 20:	Porcentagem de cada ramo vendido pelo corretor - Tela 2.....	56
Figura 21:	Porcentagem de renovação da corretora.....	57
Figura 22:	Porcentagem de renovação por seguradora.....	58
Figura 23:	Porcentagem de cancelamentos por seguradora	59
Figura 24:	Porcentagem total de cancelamentos por seguradora.....	60
Figura 25:	Porcentagem de vendas por seguradora.....	61

LISTA DE SIGLAS

ASCII	<i>American Standard Code for Information Interchange</i>
CGI	<i>Common Gateway Interface</i>
DTA	<i>Data Transformation Services</i>
FTP	<i>File Transfers Protocol</i>
HTML	<i>HyperText Markup Language</i>
HTTP	<i>HyperText Transfer Protocol</i>
IIS	<i>Internet Information Server</i>
NMTP	<i>National Software Testing Laboratories</i>
NSTL	<i>National Software Testing Laboratories</i>
OLTP	<i>Online Transaction Processing</i>
PHP	<i>Personal Home Page Tools</i>
SNMP	<i>Simple Network Management Protocol</i>
URL	<i>Uniform Resource Locator</i>
WWW	<i>World Wide Web</i>

RESUMO

O trabalho consiste na especificação e implementação do protótipo de um sistema de informações gerenciais com acesso a banco de dados relacionais via *Web*, voltado a oferecer serviços de consulta a contratos de seguros e fornecer dados para auxiliar na tomada de decisões gerenciais e estratégicas.

ABSTRACT

This work concerns in the specification and implementation of a managerial information system prototype with access to the data base by *Web*. The prototype is designed to offer consulting services to insurance contracts and data to aid managerial and strategical decisions making.

1. INTRODUÇÃO

A inexistência de companhias de seguros na antiguidade significava a ausência de uma empresa especializada em gerenciar riscos. Neste passado distante, remediava-se os sinistros depois que eles ocorriam, bem diferente do que é hoje.

Segundo [SEG2000], no século XII d. c., apareceu o contrato de dinheiro e risco marítimo. Por força das suas cláusulas, um financiador emprestava ao navegador o dinheiro correspondente ao valor da embarcação e das mercadorias transportadas. Se não houvesse acidente, o navegador devolvia o dinheiro ao financiador acrescido de juros. Em caso de acidente, o dinheiro não era devolvido. Durante muito tempo, a atividade seguradora esteve vinculada a operações bancárias. Mesmo o primeiro contrato de seguro firmado em 1347, em Gênova, na Itália, ainda não contava com a figura do seguro como gerência de risco.

Atualmente a organização e funcionamento do contrato de seguro que estabelece-se entre segurado e seguradora é regido pelo Sistema Nacional de Seguros Privados (SNSP), e, deve ser firmado somente por órgãos competentes, os quais são denominados corretores de seguros. O corretor de seguro é a representação oficial e autorizada pela Superintendência de Seguros Privados (SUSEP), a intermediar este contrato, entre segurado e companhia seguradora, prestando serviços de avaliação e atendimento ao segurado, quanto às coberturas contratadas, gerência de riscos e representação do mesmo junto a companhia seguradora.

E hoje, como pode-se fazer quase tudo no mundo virtual (comprar, vender, trocar informações, visualizar relatórios), a possibilidade de gerenciar informações correlatas a estes contratos a distância através de acesso a banco de dados via *Web*, está se tornando uma realidade. Uma das tecnologias atualmente disponíveis são as de programação via *Internet*, que fazem com que o *Hiper Text Markup Language* (HTML), antes inflexível, transforme-se em algo mutante e totalmente dinâmico. O *Personal Home Page Tools* (PHP), que é uma tecnologia de *scripts* que rodam no servidor, é *Server-side*, ou seja, sua interpretação é feita integralmente pelo servidor, sendo apenas a resposta (*output*) enviada ao cliente.

A *Internet*, para o corretor não é apenas um canal de comunicação, mas estar presente na rede com o máximo de informações a respeito de sua profissão e considerar seu *WEB SITE*

como uma empresa aberta 24 horas por dia, 7 dias por semana, pois através do *site*, o cliente pode se comunicar a qualquer hora, fazer sugestões, esclarecer dúvidas. O mais importante é que, seja na corretora real ou na virtual, os corretores estejam preparados para exercer o papel de consultor e profissional insubstituível na relação com os consumidores, que estão cada vez mais exigentes.

Um *site* freqüentado é uma calçada movimentada (e com um baixo custo). A corretora deve, então, movimentar seu espaço na rede com o máximo possível de informações a respeito de sua atividade e mercado. Outros dados a serem considerados é que a maioria das análises registra que, tanto em vendas pela rede quanto em serviços e publicidade digital, os números brasileiros deverão continuar dobrando.

Os corretores têm de somar conhecimentos, melhorar a competitividade, pensar em termos de negócio como forma básica de dominar o presente e, ao mesmo tempo, raciocinar em torno de objetivos maiores, entre eles, como antecipar o futuro. Isso tudo implica em alguns pontos básicos:

- a) utilização inteligente da tecnologia disponível no mercado para potencializar as relações com os distintos tipos de cliente;
- b) ficar atento ao fato de que os clientes estão mais informados, o que torna imprescindível entender o que está em sua mente, o que se passa entre os estímulos externos e as decisões de compra;
- c) maior consciência da qualidade implica na aceitação do que o cliente valoriza como bom e não naquilo que imagina-se ser melhor;
- d) criatividade em sentido amplo;
- e) pensar estrategicamente, ou seja ter uma visão de longo prazo para o objetivo comercial que se persegue;
- f) aceitar as mudanças, o que implica no desafio de reconhecer que há novos componentes culturais que estão mudando o comportamento de todos os envolvidos no processo, desde o lado da oferta como da demanda .

Na *Internet*, os dados são checados no banco de dados, imediatamente e retornados ao usuário. Este por sua vez, quer encontrar algo que só a *Internet* possa oferecer, como por exemplo a comodidade de ter acesso a sua apólice, ou seja seu contrato de seguros, exatamente na hora em que desejar sem depender do atendimento de ninguém. Os recursos devem ser convenientes ao usuário, facilitar a sua vida, ou seja, devem prestar-lhe um serviço, e não lhe dar um serviço. Segurança, agilidade e facilidade são alguns dos benefícios que são oferecidos através do acesso a banco de dados via *Web*.

Face a estas necessidades, a corretora de seguros para não perder seu espaço, resolveu investir neste mercado, mesmo sendo difícil de avaliar com precisão o alcance desta alternativa, que é a possibilidade de o cliente/usuário entrar no banco de dados da empresa e consultar informações que dizem respeito ao seu contrato de seguros, ainda com a opção de vislumbrar que corretor o atende, quais suas parcelas, quais seus vencimentos, juntamente com estatísticas de sinistros correspondentes a cada item segurado, possibilitar à gerência o acesso a informações de cunho estratégico para o negócio, e ainda disponibilizar ao corretor o acesso a sua carteira de clientes, permitindo ao mesmo o acompanhamento de suas metas e análise de resultados.

Sendo funcionária da corretora ADDmakler.Herco Cor. e Adm de Seguros Ltda, conhecendo-se a área de seguros e estando concluindo o Curso de Ciências da Computação, e por ter acompanhado o desenvolvimento de um software específico de controle operacional para a corretora, e diante do anseio e necessidade de evolução na área gerencial e tecnológica em serviços, e tratando-se também de uma inovação cultural, e estratégica para a empresa, obteve-se a autorização da mesma para efetuar consultas ao seu banco de dados e efetuação de testes.

Isto possibilitou a realização do estágio, juntamente ao acompanhamento, supervisão e apoio da empresa Simple Informática, desenvolvedora de software, e que presta serviços a corretora, na qual foi realizado o mesmo. Esta prestação de serviço via *Internet* visa também o relacionamento com o cliente, e quando fala-se nessa estratégia, a tecnologia deve ter a função viabilizadora do projeto. Esta evolução requer uma gestão empresarial que ajuste os processos internos da empresa, integrando-os ainda mais através do computador, disponibilizando informações de forma transparente para tomadas de decisões rápidas e

seguras. Sendo assim, os dados então, se tornarão facilmente acessíveis, tanto interna, como externamente. Isso traz a diminuição do ciclo operacional e a fidelização dos clientes, que sentem mais eficiência e credibilidade no processo.

1.1 OBJETIVOS

O objetivo principal deste trabalho é demonstrar a utilização da linguagem PHP, com acesso via *Web*, e consulta a um banco de dados, através de um aplicativo para uma corretora de seguros.

Os objetivos secundários do trabalho são:

- a) propiciar consultas operacionais;
- b) demonstração com gráficos estatísticos;
- c) auxiliar na tomada de decisão estratégica para corretora de seguros;
- d) demonstração da linguagem PHP, através da construção de um protótipo utilizando o banco de dados *Sql Server*.

1.2 ORGANIZAÇÃO

No primeiro capítulo deste trabalho tem-se a introdução, apresentando os objetivos e a organização do trabalho.

No segundo capítulo do trabalho, estão descritas as informações sobre seguros, sociedades seguradoras, termos técnicos, tipos de seguros e corretor de seguros.

No terceiro capítulo, são apresentadas as características da ferramenta IIS, utilizada no protótipo para integração com *Windows NT Server* para criação e execução de aplicativos da *Web*, publicação e compartilhamento de informações, gerenciamento de *sites* da *Web* e vantagens de performance e usabilidade sobre outros servidores.

O quarto capítulo é sobre a linguagem PHP, sua estrutura, suas funções, suas variáveis, operações e resultado na *Web*.

No quinto capítulo são analisadas as ferramentas utilizadas no desenvolvimento do protótipo, como a conexão a um banco de dados, neste caso *Sql Server*.

No sexto capítulo é descrito o protótipo elaborado com base no estudo realizado.

No sétimo capítulo descreve-se sugestões, limitações e principais conclusões do trabalho.

2. SEGUROS

O Sistema Nacional de Seguros Privados – SNSP – foi instituído pelo Governo Federal, através do Decreto-Lei Nº 73, de 21 de novembro de 1966. Anteriormente ao Decreto-Lei Nº 73, o seguro no Brasil era regulado pelo Decreto-Lei Nº 2.063, de 7 de março de 1940. Este decreto não foi totalmente revogado, vigorando ainda para muitos assuntos relativos a seguros.

O seguro é uma operação que se materializa com a realização de um contrato de natureza jurídica e bilateral, tendo em vista que dele emanam responsabilidades, direitos e obrigações para ambas as partes contratantes [ELE1999].

Qualquer que seja sua função, o seguro apresenta três características básicas: previdência, incerteza e mutualismo. Previdência significa proteção contra danos e perdas que possam atingir, no futuro, as próprias pessoas ou às suas propriedades ou bens. A incerteza significa dúvida quanto à ocorrência e/ou época do evento que provoca danos e perdas. Nos seguros de bens ou coisas, a incerteza abrange dois aspectos: a ocorrência (vai acontecer?) e a época (quando?). Nos seguros de vida, a incerteza refere-se somente à época (quando?) .

Mutualismo é a reunião de um grupo de pessoas, com interesses seguráveis comuns, que concorrem para a formação de uma massa econômica, com a finalidade de suprir, em determinado momento, necessidades eventuais de algumas daquelas pessoas.

Considera-se, portanto, uma operação de seguro como sendo a celebração de um contrato jurídico, onde uma das partes (o segurador) se obriga para com a outra (segurado), mediante o recebimento de uma importância estipulada (o prêmio), a indenizá-la (o Sinistro), resultante de um evento futuro, possível e incerto (o Risco), indicados no contrato. O objetivo do seguro deve ser visto, sempre, como de “repor” o dano causado pela ocorrência do sinistro.

O contrato de seguro é firmado através de uma apólice ou bilhete de seguro, emitido pela seguradora ou segurado, na qual indicará: o ramo do seguro, os riscos assumidos, o valor segurado, o objeto do seguro, o valor do prêmio, o prazo de vigência e outros elementos que forem acordados entre as partes. São objetos de contrato de seguros: bens, coisas, pessoas, responsabilidades e direitos que uma pessoa (o segurado) deseja garantir contra prejuízos que,

no futuro, possam se verificar. A obrigação do segurado é pagar o prêmio e a do segurador indenizar, observadas as condições gerais e particulares do contrato de seguro.

O seguro classifica-se em dois grupos: seguros sociais e seguros privados. Seguros Sociais são os que se destinam à proteção das classes trabalhadoras, sendo sua característica básica a obrigatoriedade. Esses seguros são operados pelo Estado e incluem assistência médica, aposentadoria, pensão e acidentes de trabalho.

Seguros Privados são aqueles operados por empresas privadas de seguro, podendo ou não ser obrigatórios. Podem apresentar, ainda, características sociais, como, por exemplo, o seguro obrigatório de Danos Causados por Veículos Automotores de Via Terrestre – DPVAT [ELE1999].

Com a finalidade de identificar as diferentes formas de seguros que atendem as diversas atividades econômicas e sociais, foi estabelecida uma classificação distinta, de acordo com o risco contra o qual o segurado deseja cobertura. Entende-se por cobertura a garantia de reembolso ao segurado dos prejuízos decorrentes da efetivação do sinistro previsto no contrato de seguro.

2.1 SOCIEDADES SEGURADORAS

As sociedades seguradoras são empresas que operam na aceitação dos riscos de seguro, respondendo, junto ao segurado, pelas obrigações assumidas. A atuação dessas empresas é normatizada e delimitada. Estão sujeitas às normas, instruções e fiscalização da SUSEP (Superintendência de Seguros Privados) e do IRB (Instituto de Resseguros do Brasil), bem como se obrigam a lhes fornecer dados e informações relacionados a quaisquer aspectos de sua atividade [ELE1999].

É atribuição das sociedades seguradoras administrar eficientemente os seguros que lhes são confiados. Os Bens que são colocados como garantia das reservas técnicas de uma sociedade seguradora devem ser registrados na SUSEP. Estes bens não podem ser alienados ou gravados sem prévia autorização da SUSEP.

A autorização para funcionamento das sociedades seguradoras será fornecida através de portaria do Ministério da Fazenda, mediante requerimento firmado pelos incorporadores, dirigido ao CNSP e apresentado através da SUSEP.

2.2 TERMOS TÉCNICOS

A apólice de seguro é o documento que o segurador emite após a aceitação da cobertura do risco proposto pelo segurado, com base nos dados registrados na proposta. A apólice deve conter alguns elementos considerados indispensáveis, como [SEG2000]:

- a) nome;
- b) endereço;
- c) especificação do risco;
- d) bem segurado;
- e) valor da importância segurada;
- f) valor do prêmio.

Além dos elementos citados acima, são incluídos todos aqueles julgados como necessários para distinguir os direitos e obrigações das partes contratantes – o segurado e o segurador. Entre estes podemos citar:

- a) averbação: é o documento que o segurado utiliza para informar à seguradora sobre verbas e objetos a garantir a apólice aberta. As averbações são utilizadas para segurados que tenham expressivo movimento nos seus negócios, pois permite maior agilidade na contratação do seguro. O segurado obriga-se a entregar o formulário de averbação à seguradora mediante protocolo ou remetê-lo sob registro postal [ELE1999].
- b) cosseguro: é a operação onde mais de uma seguradora participa diretamente de um mesmo risco em uma mesma apólice. Na operação de cosseguro cada

segurador é responsável por uma cota ou parte do montante total do seguro. O prêmio pago é dividido na proporção da cota de cada segurador.

- c) endosso: é o instrumento que o contrato de seguro utiliza quando, eventualmente, é necessário fazer alguma modificação na apólice.
- d) importância Segurada: é o valor estabelecido pelo segurado que deverá corresponder ao valor do bem segurado. O valor da importância é estabelecido de modo distinto entre o seguro de coisas e o seguro de pessoas.
- e) No Seguro de Coisas é expressamente proibido por lei segurar um objeto com valor superior ao real. Nos Seguros de Pessoas e Responsabilidades, a determinação da importância segurada é livre entre as partes contratantes.
- f) indenização: é a contra-prestação do segurador. Ou seja, a importância que a companhia seguradora deverá pagar ao segurado no caso da efetivação de um risco coberto no contrato de seguro. O Código Civil estabelece no seu artigo 1.458 que o “segurador é obrigado a pagar em dinheiro o prejuízo resultante do risco assumido e, conforme as circunstâncias, o valor total da coisa segurada”.
- g) prêmio: é a soma em dinheiro, paga pelo segurado à seguradora, para que esta assuma a responsabilidade de um determinado risco. O cálculo de um prêmio é feito com base no prazo do seguro, da importância segurada e da exposição ao risco.
- h) proposta: é o documento onde o segurado registra sua vontade em segurar determinado objeto ou bem, nos mais diferentes riscos.
- i) pulverização do risco: significa distribuir ou dividir as responsabilidades do risco assumido pelo segurador, através do cosseguro e do resseguro com a finalidade de garantir a homogeneidade qualitativa e quantitativa das carteiras de seguro.
- j) resseguro: é a operação de que se vale um ou mais seguradores para transferir à resseguradora o excesso de responsabilidades que ultrapassa o limite de sua capacidade econômica de indenizar.

- k) risco: o termo risco, no contrato de seguro, significa o evento incerto, de data incerta, que independe da vontade das partes (segurado e segurador) e contra o qual é feito o seguro.
- l) segurado: é uma pessoa física ou jurídica, em nome de quem se faz o seguro e que possui interesse econômico no bem exposto ao risco.
- m) estipulante e beneficiário: é a pessoa física ou jurídica que contrata um seguro a favor do segurado. Em alguns casos, a pessoa do estipulante é distinta do segurado, podendo ser o representante ou mandatário do segurado.
- n) beneficiário é a pessoa física ou jurídica designada pelo segurado para receber as indenizações devidas pelo segurador. Existem casos em que o segurado indica um beneficiário. Tal situação é típica nos seguros de vida, em que o risco coberto é a morte do próprio segurado.
- o) segurador: é a empresa legalmente constituída para assumir e gerir riscos. È, portanto assume a responsabilidade de determinados riscos, devidamente especificados no contrato de seguros. O Segurador é quem paga a indenização ao segurado ou aos seus beneficiários, no caso da ocorrência do sinistro.
- p) sinistro: é o evento previsto no contrato de seguro, quando realizado, causando danos ao segurado ou a seus beneficiários. Quando um risco se efetiva, isto quer dizer que ocorreu um sinistro.

2.3 CORRETORES DE SEGUROS

Segundo [ELE1999], os Corretores de Seguros, pessoa física ou jurídica, são os profissionais legalmente autorizados a intermediar o contrato de seguro entre a seguradora e o segurado.

Cabe ao corretor de seguros intermediar os seguros pretendidos, bem como orientar e esclarecer o segurado sobre as coberturas necessárias à sua atividade. As comissões de corretagem de seguros só poderão ser pagas aos corretores devidamente habilitados.

O exercício da profissão de corretor de seguros depende de prévia habilitação e registro, sendo que a habilitação se dá através de exame realizado pela Fundação Escola Nacional de Seguros – FUNENSEG.

O corretor não pode aceitar ou exercer empregos públicos. Além disso, é responsável, civilmente, perante os segurados e as sociedades seguradoras pelos prejuízos que a eles causar por omissão, imperícia ou negligência, no exercício de sua profissão, e está sujeito às normas, instruções e fiscalização da SUSEP. Os impedimentos relacionados acima, atingem também os sócios e diretores de corretoras.

3. INTERNET INFORMATION SERVER (IIS)

As empresas estão procurando tecnologias da *Web* baseadas em padrões a fim de trocar informações de forma mais eficaz com clientes, parceiros e funcionários, em todo o mundo. Além disso, elas estão tentando implementar soluções que tirem vantagem dos investimentos existentes em conhecimento e em tecnologia. O *Microsoft Internet Information Server* (IIS) é projetado de forma a atender as necessidades de uma ampla série de usuários, desde grupos de trabalho e de departamentos, em uma *intranet* corporativa, até Provedores de Serviços da *Internet* que abrigam *sites* da *Web* acessados por milhões de visitantes por dia. O IIS é um recurso da *Web* do sistema operacional *Microsoft Windows NT Server*, que permite compartilhar informações, criar e distribuir aplicativos comerciais, abrigar e gerenciar *sites* [IIS2000].

3.1 INTEGRAÇÃO COM WINDOWS NT SERVER

O *Windows NT Server* é um servidor que permite o gerenciamento, para o compartilhamento de informações e a execução de aplicativos. O *Windows NT Server* pode conter serviços integrados da *Web* de modo a incluir esses recursos na *intranet*.

O IIS proporciona um bom desempenho de servidor da *Web* em *Windows NT Server* e oferece uma relação preço-desempenho considerável, se comparada com outros sistemas. Os serviços de aplicativos do *Windows NT Server* fornecem uma plataforma expansível para a criação e distribuição de aplicativos da *Web* [IIS2000].

O IIS traz a segurança do *Windows NT Server* para o *site* da *Web*, para proteger as informações com a facilidade de um único diretório de usuário e a capacidade de fazer *login* em uma rede.

O *Windows NT Server* com *Internet Information Server* é uma solução que fornece um conjunto integrado de ferramentas para executar e gerenciar a rede, a *Web* e os serviços de aplicativo.

3.2 PUBLICAÇÃO E COMPARTILHAMENTO DE INFORMAÇÕES

Segundo [MIC2000] recursos de publicação na *Web*, ferramentas personalizáveis e novas tecnologias de assistentes exclusivos do *Internet Information Server*, tornam o *Windows NT Server* com IIS uma maneira de publicar informações e compartilhá-las seguramente em *intranets* corporativas e na *Internet*. Com ele é possível:

- a) criar páginas da *Web* de qualidade e publicar *sites* inteiros, sem saber HTML, incluindo novos assistentes e modelos;
- b) publicar informações na *Web* usando um navegador da *Web*, o assistente de publicação na *Web* ou FTP;
- c) compartilhar arquivos e dados em servidores *Windows NT*, *Novell NetWare* e UNIX, e em 55 bancos de dados, inclusive bancos de dados *Microsoft Sql Server*, *Oracle* e *Sybase*;
- d) pesquisar o conteúdo em tipos de documento HTML e *Microsoft Office*.

3.3 CRIAÇÃO E EXECUÇÃO DE APLICATIVOS DA WEB

Os aplicativos da *Web* podem simplificar processos comerciais, tais como relatório de despesas, gerenciamento de lucros, compras, gerenciamento de estoque, *marketing* e vendas. O IIS fornece o que se precisa para distribuir aplicativos da *Web* em *Windows NT Server*, para atender a estas necessidades comerciais [MIC2000].

O IIS utiliza um recurso conhecido como isolamento de processo para proteger aplicativos e *sites* da *Web* contra falhas causadas por componentes de comportamento falho ou aplicativos da *Web* no servidor, executando-os em espaços de memória separados.

Os desenvolvedores podem focalizar a solução comercial e não as complexidades tipicamente associadas ao desenvolvimento de aplicativo de servidor [IIS2000].

Desenvolvedores familiarizados com aplicativos de área de trabalho, escritos em qualquer linguagem, podem escrever e depurar componentes e aplicativos executados no servidor.

3.4 GERENCIAMENTO DE SITES DA WEB

As ferramentas de gerenciamento do IIS ajudam a configurar *sites* da *Web*, gerenciar conteúdo e analisar padrões de uso para desenvolver o *site*. Ferramentas de gerenciamento personalizáveis, opções de administração flexíveis e ferramentas de análise, tornam o *Windows NT Server* com IIS o servidor da *Web* fácil de gerenciar. Novos recursos de configuração permitem definir propriedades independentemente para cada *site* da *Web*, diretório virtual ou arquivo no servidor [MIC2000].

Com o ISS é possível automatizar tarefas de gerenciamento comuns em vários servidores, tais como a criação de novos *sites* da *Web* e de diretórios, executando *scripts* de administração a partir da linha de comando; e ainda analisar arquivos de registro para determinar os padrões de uso e analisar o conteúdo do *site* da *Web* com ferramentas de análise integradas [IIS2000].

3.5 VANTAGENS DE PERFORMANCE E USABILIDADE SOBRE OUTROS SERVIDORES

De acordo com a edição de abril de 2000, da publicação *Software Digest*, o *Internet Information Server 3.0*, da *Microsoft* é o servidor mais recomendado do mercado pela sua rapidez e integração com o *Windows NT*, na administração e gerenciamento. A publicação divulgou os resultados de testes comparativos realizados pelo NSTL (*National Software Testing Laboratories*) entre o *Internet Information Server 3.0* e os servidores *Enterprise*, da *Netscape* e *Domino*, da *Lotus*. O produto da *Microsoft* obteve a média geral 8.1 contra 7.1 da *Netscape* e 6.0 do *Lotus Domino*. A NSTL conduziu os testes dos servidores *Web* avaliando três características, com diferentes valores: versatilidade (40%), performance (40%) e usabilidade (20%) [MIC2000].

O *Internet Information Server 3.0* vem com *add-ons* tais como o *Index Server*, *Crystal Report Writer*, *NetShow*, *FrontPage* e a nova tecnologia *Active Server Pages*. Como todos os produtos avaliados, o IIS também prioriza as áreas de funcionalidade para aplicações distribuídas de uma maneira transparente, otimizada e com um excelente custo benefício. Para as organizações que procuram aplicações de servidor para melhorias na infra-estrutura da tecnologia (especialmente grupos de colaboração), é necessário adicionar uma base de dados e sistema de mensagem, servidor de *cluster* e possivelmente *add-ons* de processamento transacionais para realizar a tarefa corretamente.

O *Internet Information Server* oferece considerável performance em todos os oito níveis de requisição de cliente e o tempo de resposta para o acesso a um *site* foi menor entre alguns produtos testados. A versão UNIX do *Netscape Enterprise Server* obteve resultados de performance semelhantes ao IIS. O servidor UNIX, utilizado para testar o *Netscape* usou quatro processadores *Alpha* de 400 Mhz, já o servidor *Windows NT* utilizado para testar o IIS precisou de quatro processadores *Pentium* 200 de Mhz. Quando os dois sistemas rodaram na mesma configuração, o servidor da *Netscape* apresentou performance inferior [IIS2000].

Depois que os técnicos se familiarizaram com os programas via documentação, foram instalados os *softwares* e selecionados os programas de acordo com a facilidade de uso na criação e administração de três tipos de *sites Web*: um *site* informativo de uma empresa; um *site* com conteúdo multimídia e outro utilizado como um ambiente colaborativo para negócios. Toda a documentação dos produtos foi bem organizada e observou-se que o acesso ao *help-online* é mais rico nos servidores da *Microsoft* e da *Lotus*. O *Domino* utiliza um banco de dados do *Notes* para armazenar arquivos *help* assim como o padrão de sistema *Windows*.

Em suma, o desempenho do IIS 3.0 com novas tecnologias como ASP (*Active Server Pages*) nativo, e PHP oferecem uma interface familiar em relação ao *Visual Basic* e permite, ainda, a inclusão de código diretamente em páginas HTML sem a necessidade de executar qualquer tipo de compilação.

4. A LINGUAGEM PHP

Personal Home Page Tools - PHP é uma linguagem que permite criar *sites WEB* dinâmicos, possibilitando uma interação com o usuário através de formulários, parâmetros da URL e *links* [VIV2000].

O uso de páginas estáticas para o controle/apresentação de informações ao usuário se mostra quase que impraticável, a medida que a necessidade de constantes atualizações se fazem necessárias no dia-a-dia da página. Por meio de linguagens de *script* (PHP), é possível criar essas páginas com o seu conteúdo dinâmico de acordo com as condições apresentadas ao usuário [WAP2000].

Uma linguagem *Script*, pode ser entendida como uma página que é interpretada pelo servidor, devolvendo um formato HTML de acordo com a requisição de um *browser* cliente. Esse *script* nada mais é do que um direcionador de saída que irá exibir ao cliente um formato HTML ou silimar em função do roteiro seguido pelo mesmo.

A diferença de PHP com relação a linguagens semelhantes a *Javascript* é que, o código PHP é executado no servidor, sendo enviado para o cliente apenas HTML puro. Desta maneira é possível interagir com bancos de dados e aplicações existentes no servidor, com a vantagem de não expor o código fonte para o cliente. Isso pode ser útil quando o programa está lidando com senhas ou qualquer tipo de informação confidencial.

O que diferencia PHP de um *script CGI*, escrito em *C* ou *Perl*, é que o código PHP fica embutido no próprio HTML, enquanto no outro caso é necessário que o *script CGI* gere todo o código HTML, ou leia de um outro arquivo. Basicamente, qualquer coisa que pode ser feita por algum programa CGI pode ser feita também com PHP, como coletar dados de um formulário, gerar páginas dinamicamente ou enviar e receber *cookies* [VIV2000].

O PHP também tem como uma das características mais importantes o suporte a um grande número de bancos de dados, como *dBase*, *Interbase*, *mSQL*, *mySQL*, *Oracle*, *Sybase*, *PostgreSQL* e vários outros. Construir uma página baseada em um banco de dados torna-se uma tarefa extremamente simples com PHP [CON2000].

Além disso, PHP tem suporte a outros serviços através de protocolos como IMAP, SNMP, NMTP, POP3 e, logicamente, HTTP. Ainda é possível abrir *sockets* e interagir com outros protocolos [WAP2000].

A linguagem PHP foi concebida durante o outono de 1994 por Rasmus Lerdorf. As primeiras versões não foram disponibilizadas, tendo sido utilizadas em sua *home-page* apenas para que ele pudesse ter informações sobre as visitas que estavam sendo feitas. A primeira versão utilizada por outras pessoas foi disponibilizada em 1995, e ficou conhecida como “*Personal Home Page Tools*” (ferramentas para página pessoal). Era composta por um sistema bastante simples que interpretava algumas *macros* e alguns utilitários que rodavam “por trás” das *home-pages*: um livro de visitas, um contador e algumas outras coisas.

Em meados de 1995 o interpretador foi reescrito, e ganhou o nome de PHP/FI, o “FI” veio de um outro pacote escrito por Rasmus que interpretava dados de formulários HTML (*Form Interpreter*). Ele combinou os *scripts* do pacote *Personal Home Page Tools* com o FI e adicionou suporte a mSQL, nascendo assim, o PHP/FI.

Por volta de 1997 houve uma mudança no desenvolvimento do PHP. Ele deixou de ser um projeto de Rasmus com contribuições de outras pessoas para ter uma equipe de desenvolvimento mais organizada. O interpretador foi escrito por Zeev Suraski e Andi Gutmans, sendo que o novo interpretador foi a base para a versão 3.

4.1 ESTRUTURA DE UM PROGRAMA PHP

Basicamente um programa PHP possui a seguinte estrutura [SOA2000]:

- a) **Comandos HTML** - comandos que serão enviados ao *browser* do cliente sem qualquer formatação pelo PHP. Nesta parte pode-se colocar a inicialização do HTML (<html>, <head>, <body>...) e quaisquer outras informações úteis à melhor apresentação da página, tais como: programas *JavaScript* e outros;
- b) **Código PHP** - começando sempre por <?php, pode-se colocar um ou vários *scripts* PHP numa página, concatenando *scripts* PHP com *tags* HTML. Os *scripts*

4.2 RESULTADO NA WEB

A partir do momento que é escrito o programa em PHP, ele já está pronto para ser exibido na *WEB*. É claro que programas que não tenham nenhuma linha “*echo*” ou “*printf*” não exibirão nada no *browser*. Assim, para que o programa exiba alguma informação dinâmica, deverá utilizar comando “*echo*” ou “*printf*” dentro do código PHP (Quadro 2).

Quadro 2: Resultado do programa PHP -

Exemplo 1

```
<html>
<body>
<?php
// este programa não mostra nada
$b = 20.45;
$h =1.546;
$nome = "meu nome é Valda";
$nome.= " e eu nasci em Santa Catarina";
$dia = date ("d/m/Y",time ());
$numero = 123.45;
$area = $b * $h / 2;
?>
</body>
</html>
```

Fonte: [BAR2000]

Este programa vai mostrar no *browser*, simplesmente uma tela em branco, pois não existe qualquer linha com comando de saída. O *script* foi processado e o PHP armazenou em memória os valores das variáveis declaradas, mas esses valores foram perdidos quando o resultado foi enviado ao *browser* do usuário. Se o usuário pedir para visualizar o código fonte no *browser*, terá apenas: `<html> <body> </body> </html>`

Mas alterando um pouco o programa anterior, inserindo um comando de exibição após a definição das variáveis (Quadro 3). Obtêm-se uma página com as informações apresentadas no Quadro 4.

Quadro 3: Resultado do programa PHP - Exemplo 2

```

<html>
<body>
<?php
// este programa agora mostra as informações
$b = 20.45;
$h =1.546;
$nome = "meu nome é Valda";
$nome.= " eu nasci em Santa Catarina";
$dia = date ("d/m/Y",time ());
$numero = 123.45;
$area = $b * $h / 2;
echo "$nome<br>Hoje é dia $dia<br> a variavel \$numero tem o
valor de $numero<br>A area de um triangulo com base = $b e
altura = $h é $area<br>";
?>
</body>
</html>

```

Fonte: [BAR2000]

Quadro 4: Resultado do programa PHP - Exemplo 3

```

Meu nome é Valda eu nasci em Santa Catarina
Hoje é dia 03/12/2000
a variável $numero tem valor de 123.45
A area de um triangulo com base = 20.45 e altura 1.546 é
15.8079

```

Fonte: [BAR2000]

Isto leva a concluir que, a menos que não se precise mostrar nada ao usuário, e isto será útil em algumas situações (como atualização de uma base de dados, por exemplo), é imprescindível que se utilize os comandos de saída do PHP.

4.3 COMENTÁRIOS NO PHP

Todo programa deve possuir comentários, visando o entendimento do código em consultas posteriores. No PHP, existem três tipos de marcadores de comentário, que são: // # e /* */ para comentário de uma linha, conforme os Quadros 5 e 6.

Quadro 5: Comentários no PHP - Exemplo 1

```
// atribui o nome à variável  
$nome "Valda";  
$email valda@furb.com.br; # atribui o E-mail à variável
```

Fonte: [SOA2000]

Quadro 6 - Comentários no PHP - Exemplo 2

```
/*  
Nas linhas abaixo, atribuiremos os valores  
Do nome e do e-mail às respectivas variáveis  
*/  
$nome = "Valda";  
$email = "valda@furb.com.br";
```

Fonte: [SOA2000]

4.4 VARIÁVEIS NO PHP

O PHP trata suas variáveis (ou constantes), que podem ser variáveis escalares ou não-escalares. As variáveis escalares são aquelas que podem ser retrabalhadas, ou “divididas em pedaços menores”, enquanto as não escalares são as *arrays* (matrizes) e os objetos.

A identificação de uma variável, independente do seu tipo é pelo sinal \$ colocado como primeiro caractere, como no Quadro 7.

Quadro 7 - Variáveis no PHP

```
$nome = "Valda";  
$matricula = 9843825;  
$conex= ocilogon("scott", "tiger", "ORA8");
```

Fonte: [SOA2000]

A primeira variável é do tipo *string*; a segunda, inteiro (ambas escalares); e a terceira é uma variável que possui a referência de um objeto de conexão com um banco de dados Oracle.

Como a linguagem C, as variáveis \$nome e \$Nome são consideradas diferentes, pois o PHP as trata como sensíveis ao caso.

O modificador *static*, conforme Quadro 8 é uma variável estática e visível num escopo local, mas ela é inicializada apenas uma vez e seu valor não é perdido quando a execução do *script* deixa esse escopo.

Quadro 8 - Uso do modificador *static* no PHP - Exemplo 1

```
function Teste () {  
    $a = 0;  
    echo $a;  
    $a++;  
}
```

Fonte: [BAR2000]

O último comando da função é inútil, pois assim que for encerrada a execução da função a variável \$a perde seu valor. Já no exemplo seguinte, a cada chamada da função a variável \$a terá seu valor impresso e será incrementada, conforme Quadro9.

Quadro 9 - Uso do modificador *static* no PHP - Exemplo 2

```
function Teste () {  
    static $a = 0;  
    echo $a;  
    $a++;  
}
```

Fonte: [BAR2000]

O modificador *static* é muito utilizado em funções recursivas, já que o valor de algumas variáveis precisa ser mantido. Ele funciona da seguinte forma: o valor das variáveis declaradas como estáticas é mantido ao terminar a execução da função. Na próxima execução da função, ao encontrar novamente a declaração com *static*, o valor da variável é recuperado.

As variáveis cujos nomes também são variáveis no PHP tem um recurso conhecido como variáveis variáveis. Sua utilização é feita através do duplo cifrão (\$\$).

- `$a = "teste";`
- `$$a= "Valda";`

O que equivale ao seguinte:

- `$a = "teste";`
- `$teste = "Valda";`

As variáveis enviadas pelo navegador, para interagir com a navegação feita pelo usuário, são necessárias para que o PHP possa enviar e receber informações para o *software* de navegação. A maneira de enviar informações, geralmente é através de um comando de impressão, como o *echo*. Para receber informações vindas do navegador através de um *link* ou um formulário HTML o PHP utiliza as informações enviadas através da URL. Por exemplo: se o *script* PHP está localizado em "http://localhost/testephp1" e o usuário o chama com a url "http://localhost/teste.php1?valda=teste", automaticamente o PHP criará uma variável como nome \$valda contendo a *string* "teste". Nota-se que o conteúdo da variável está no formato *urlencode*. Os formulários HTML já enviam informações automaticamente nesse formato, e o PHP decodifica sem necessitar de tratamento pelo programador.

O formato *urlencode* é obtido substituindo os espaços pelo caracter "+" e todos os outros caracteres não alfa-numéricos (com exceção de "_") pelo caracter "%" seguido do código ASCII em hexadecimal.

Por exemplo: o texto "Testando 1 2 3 !!" em *urlencode* fica "Testando+1+2+3+%21%21"

O PHP possui duas funções para tratar com texto em *urlencode*. As sintaxes são:

```
string urlencode(string texto);
```

```
string urldecode(string texto);
```

Essas funções servem respectivamente para codificar ou decodificar um texto passado como argumento.

O PHP possui diversas variáveis de ambiente, como a `$PHP_SELF`, por exemplo, que contém o nome e o *path* do próprio arquivo. Algumas outras contém informações sobre o navegador do usuário, o servidor http, a versão do PHP e diversas informações. Para ter uma listagem de todas as variáveis e constantes de ambiente, e seus respectivos conteúdos, deve-se utilizar a função `phpinfo ()`.

A função que verifica o tipo de uma variável é usada por causa da tipagem dinâmica utilizada pelo PHP. Nem sempre é possível saber qual o tipo de uma variável em determinado instante se não contar com a ajuda de algumas funções que ajudam a verificar isso. A verificação pode ser feita de duas maneiras:

- a) **Função que retorna o tipo da variável:** Esta função é a `gettype`, onde sua assinatura é a seguinte:

`string gettype(mixed var)`. A palavra “mixed” indica que a variável `var` pode ser de diversos tipos.

A função `gettype` pode retornar as seguintes *strings*: “integer”, “double”, “string”, “array”, “object” e “unknown type”;

- b) **Funções que testam o tipo da variável:** São as funções `is_int`, `is_integer`, `is_real`, `is_long`, `is_float`, `is_string`, `is_array` e `is_object`. Todas têm o mesmo formato: `int is_integer (mixed var)`. Todas essas funções retornam *true* se a variável for daquele tipo, e *false* em caso contrário.

A destruição de uma variável, se torna possível desalocando uma variável se ela não for usada posteriormente através da função `unset`, que tem a seguinte assinatura: `int unset(mixed var)`.

A função destrói a variável, ou seja, libera a memória ocupada por ela, fazendo com que ela deixe de existir. Se mais na frente for feita uma chamada á variável, será criada uma nova variável de mesmo nome e de conteúdo vazio, a não ser que a chamada seja pela função `isset`. Se a operação for bem sucedida, retorna *true*.

Para verificar se uma variável possui um valor, existem dois tipos de teste que podem ser feitos para verificar se uma variável está setada: com a função `isset` e com a função `empty`.

A função `isset`: `int isset (mixed var)`, retorna *true* se a variável estiver setada (ainda que com uma *string* vazia ou o valor zero), e *false* em caso contrário.

A função `empty`: `int empty (mixed var)`. Retorna *true* se a variável não contiver um valor (não estiver setada) ou possuir valor 0 (zero) ou uma *string* vazia. Caso contrário, retorna *false*.

4.5 OPERAÇÕES MATEMÁTICAS

As operações no PHP seguem o padrão das outras linguagens (+, -, ~, /, `sin()`, `cos()`).

Um exemplo está demonstrado no Quadro 10, para calcular o valor líquido de um preço, depois de aplicar 10% de desconto sobre o preço bruto.

Quadro 10 - Operações matemáticas

```
$valorbruto = 10;  
$desconto = 10 *$valorbruto / 100;  
$valorliquido = $valorbruto - $desconto;
```

Fonte: [BAR2000]

4.6 OPERAÇÕES COM STRINGS

Operações com *strings* são uma das características mais desenvolvidas do PHP. Entre as mais importantes estão:

- `strlen()`, que permite saber quantos caracteres possui a *string* (Quadro 11);
- `substr()`, que devolve uma *substring* da string informada (Quadro 12);
- `strpos()`, para saber se determinado caractere (ou *substring*) está contida em uma *string* (Quadro 13).

Quadro 11 – Sintaxe do operador strlen

```
echo "A palavra 'Internet' possui" .strlen("Internet") "
caracteres";
```

Fonte: [BAR2000]

Quadro 12 – Sintaxe do operador substr

```
echo substr("abcde", 2 , 2); // Esta linha irá exibir os
caracteres "cd";
```

Fonte: [BAR2000]

Quadro 13 - Operações com strings

```
if strpos ($email, "@"){
echo("Seu e-mail parece estar correto!\n");
}else{
echo ("O e-mail está inválido\n");
}
```

Fonte: [BAR2000]

Verifica-se que se o caractere "@" está contida em uma variável *\$email*. Se estiver, exibe a primeira mensagem. Do contrário, exibe a segunda.

4.7 FORMULÁRIOS HTML

O *Hyper Text Markup Language* (HTML) é uma linguagem de formatação usada para construir páginas na Internet. Ao clicar num botão “*Submit*” em um formulário HTML as informações dos campos serão enviadas ao servidor especificado para que possa ser produzida uma resposta. O PHP trata esses valores como variáveis, cujo nome é o nome do campo definido no formulário. O exemplo do Quadro 14 mostra isso.

Quadro 14 - Formulários HTML

```
<html>
<head><title>O PHP</title></head>
<body>
<?php
if ($texto != " ")
echo "Você digitou \ "$texto\ "<br><br>";
?>
<form method = post action= "<? echo $PATH_INFO; ?>">
<input type ="text" name = "texto" value = "" size = 10>
<br>
<input type = "submit" name = "sub" value = "Enviar!">
</ form>
</body>
</html>
```

Fonte: [BAR2000]

Ao salvar o arquivo acima e carregá-lo no *browser*, o usuário verá apenas um formulário que contém um espaço para digitar o texto. Ao digitar um texto qualquer e submeter o formulário, a resposta, que é o mesmo arquivo PHP (indicado pela constante \$PATH_INFO, que retorna o nome do arquivo).

Isso ocorre porque o código PHP testa o conteúdo da variável \$texto. Inicialmente ele é uma *string* vazia, e por isso nada é impresso na primeira parte. Quando algum texto é digitado no formulário e submetido, o PHP passa a tratá-lo como uma variável. Como no formulário o campo possui o nome “texto”, a variável com seu conteúdo será \$texto. Assim, no próximo teste o valor da variável será diferente de uma *string* vazia, e o PHP imprime um texto antes do formulário.

4.8 FUNÇÕES DO PHP

A sintaxe básica para definir uma função pode ser vista no quadro 15.

Quadro 15 – Sintaxe básica para definir uma função

```
function nome_da_função([arg1, arg2, arg3]) {  
Comandos;  
....;  
[return <valor de retorno>];  
}
```

Fonte: [SOA2000]

As funções no PHP não diferem muito das outras linguagens. Algumas características das funções [DIA2000]:

- a) devem ser declaradas antes de serem usadas;
- b) podem receber parâmetros por valor ou por referência;
- c) podem ter quantidade variável de parâmetros;
- d) os parâmetros podem ser declarados com um valor *default*;
- e) uma vez definida, uma função não poderá ser “redefinida”.

Qualquer código PHP válido pode estar contido no interior de uma função, conforme Quadro 16. Como a checagem de tipos em PHP é dinâmica, o tipo de retorno não deve ser declarado, sendo necessário que o programador esteja atento para que a função retorne o tipo desejado. É recomendável que esteja tudo bem documentado para facilitar a leitura e compreensão do código. Para efeito de documentação, utiliza-se o seguinte formato de declaração de função:

- `tipo function nome_da_função(tipo arg1, tipo arg2, ..);`

Este formato só deve ser utilizado na documentação do *script*, pois o PHP não aceita a declaração de tipos. Isso significa que em muitos casos o programador deve estar atento ao

tipos dos valores passados como parâmetros, pois se não for passado o tipo esperado não é emitido nenhum alerta pelo interpretador PHP, já que este não testa os tipos.

- a) **valor de retorno:** toda função pode opcionalmente retornar um valor, ou simplesmente executar os comandos e não retornar valor algum. Não é possível que uma função retorne mais de um valor, mas é permitido fazer com que uma função retorne um valor composto, como listas ou *arrays*.
- b) **argumentos:** é possível passar argumentos para uma função. Eles devem ser declarados logo após o nome da função, entre parênteses, e tornam-se variáveis pertencentes ao escopo local da função. A declaração do tipo de cada argumento também é utilizada apenas para efeito de documentação.

Quadro 16 - Funções do PHP - Exemplo 1

```
function imprime ($texto) {  
    echo $texto;  
}  
imprime ("Valda");
```

Fonte: [SOA2000]

A passagem de parâmetros por referência normalmente, em PHP é feita por valor, ou seja, se o conteúdo da variável for alterado, essa alteração não afeta a variável original.

Como a passagem de parâmetros é por valor, a função `mais5` é inútil, já que após a execução sair da função o valor anterior da variável é recuperado. Se a passagem fosse feita por referência, a variável `$a` teria 8 como valor (Quadro 17). O que ocorre normalmente é que ao ser chamada uma função, o interpretador salva todo o escopo atual, ou seja, os conteúdos das variáveis. Se uma dessas variáveis for passada como valor, seu conteúdo fica preservado, pois a função irá trabalhar na verdade com uma cópia da variável. Porém, se a passagem de parâmetros for feita por referência, toda alteração que a função realizar no valor passado como parâmetro afetará a variável que o contém [BAR2000].

Quadro 17 - Funções do PHP - Exemplo 2

```
function mais5 ($numero) {
```

```

$numero += 5;
}
$a = 3;
mais5 ($a); // $a continua valendo 3

```

Fonte: [SOA2000]

Há duas maneiras de fazer com que uma função tenha parâmetros passados por referência:

- a) indicando isso na declaração da função, o que faz com que a passagem de parâmetros sempre seja assim;
- b) também na própria chamada da função. Nos dois casos utiliza-se o modificador “&” (Quadro 18).

Quadro 18 - Funções do PHP - Exemplo 3

```

function mais5(&$num1, $num2) {
$num1 += 5;
$num2 += 5;
}
$a = $b = 1;
mais5($a, $b); /* Neste caso, só $num1 terá seu valor alterado,
pois a passagem por referência está definida na declaração da
função. */
mais5($a, &$b); /* Aqui as duas variáveis terão seus valores
alterados. */

```

Fonte: [SOA2000]

Em PHP é possível ter valores *default* para argumentos de funções, ou seja, valores que serão assumidos em caso de nada ser passado no lugar do argumento. Quando algum parâmetro é declarado desta maneira, a passagem do mesmo na chamada da função torna-se opcional (Quadro 19).

Quadro 19 - Funções do PHP - Exemplo 4

```

function teste($valda = "testando"){
echo $valda;
}
teste(); // imprime "testando"
teste("outro teste") ; // imprime "outro teste"

```

Fonte: [SOA2000]

É bom lembrar que quando a função tem mais de um parâmetro, o que tem valor *default* deve ser declarado por último.

O contexto é o conjunto de variáveis e seus respectivos valores num determinado ponto do programa. Na chamada de uma função, ao iniciar a execução do bloco que contém a implementação da mesma é criado um novo contexto, contendo as variáveis declaradas dentro do bloco, ou seja, todas as variáveis utilizadas dentro daquele bloco serão eliminadas ao término da execução da função.

O escopo de uma variável em PHP define a porção do programa onde ela pode ser utilizada. Na maioria dos casos todas as variáveis têm escopo global. Entretanto, em funções definidas pelo usuário um escopo local é criado. Uma variável de escopo global não pode ser utilizada no interior de uma função sem que haja uma declaração [BAR2000].

O trecho apresentado no Quadro 20, não produzirá nenhuma saída, pois a variável \$valda é de escopo global, e não pode ser referida num escopo local, mesmo que não haja outra com nome igual que cubra a sua visibilidade. Para que o *script* funcione da forma desejada, a variável global a ser utilizada deve ser declarada. (Quadro 21).

Quadro 20 - Funções do PHP - Exemplo 5

```
$valda = "Testando";  
function Teste() {  
    echo $valda; // Não apresenta nada na Tela  
}  
Teste();
```

Fonte: [SOA2000]

Quadro 21 - Funções do PHP - Exemplo 6

```
$valda = "Testando";  
function Teste() {  
    global $valda;  
    echo $valda;  
}  
Teste();
```

Fonte: [SOA2000]

Uma declaração “global” pode conter várias variáveis, separadas por vírgulas. Uma outra maneira de acessar variáveis de escopo global dentro de uma função é utilizando um *array* pré-definido pelo PHP cujo nome é \$GLOBALS. O índice para a variável referida é o próprio nome da variável, sem o caracter \$. O exemplo dos Quadros 21 e 22 produzem o mesmo resultado.

Quadro 22 - Funções do PHP - Exemplo 7

```
$valda = "Testando";  
function Teste() {  
    echo $GLOBALS ['valda']; // imprime $valda  
    echo $valda; // não imprime nada  
}  
Teste();
```

Fonte: [SOA2000]

5. FERRAMENTAS

5.1 SQL SERVER

O *Structured Query Language* (SQL) é uma linguagem estruturada para manipulação de dados. É padronizada para os bancos de dados relacionais, mas cada gerenciador pode possuir uma extensão própria dessa linguagem [SQL2000].

Como no exemplo de um pedido para um corretor de uma corretora de seguros, cada comando no SQL, é um pedido de busca ou alteração de dados para o gerenciador do banco de dados. Quem vai executar propriamente o comando é o gerenciador.

Neste contexto, o *Sql Server* oferece uma plataforma eficiente e flexível que suporta banco de dados de *terabytes* de informações, adaptando-se perfeitamente a aplicativos existentes. O produto fornece um ambiente de baixo custo para personalizar e desenvolver novas aplicações, criadas exclusivamente para atender às necessidades de uma corporação [GUP1990].

Facilita a criação, o gerenciamento e a distribuição de aplicativos baseados no processamento de transações *Online Transaction Processing* (OLTP).

O *Sql Server* fornece ajuste e administração automatizados ao banco de dados, bem como ferramentas sofisticadas para operações complexas. O programa oferece inovações na facilidade de uso, estabilidade, confiabilidade e desempenho, um modelo de programação rápido e simples para desenvolvedores, novo bloqueio dinâmico no nível de linha, *backup* ativo e gerenciamento multilocal.

Com o processamento analítico *online* (*Online Analytical Processing* - OLAP), é possível fornecer desempenho e eficiência otimizados para a criação de relatório, análise, suporte à decisão e modelagem de dados. Os serviços de transformação de dados (*Data Transformation Services* - DTA), facilitam a criação e automatização da manutenção de *Data Warehouses* permitindo importar, exportar e transformar dados de origem diferentes graficamente [SQL2000].

5.2 CONECTANDO UM BANCO DE DADOS

Os acessos a bancos de dados utilizarão o gerenciador de banco de dados MsSQL. Para interagir com uma base de dados SQL existem três comandos básicos que devem ser utilizados: um que faz a conexão com o servidor de banco de dados, um que seleciona a base de dados a ser utilizada e um terceiro que executa uma cláusula SQL.

A conexão com o servidor de banco de dados MsSQL em PHP é feita através do comando `mssql_connect`, que tem a seguinte sintaxe:

- `int mssql_connect(string servidor, string usuário, string senha)`.

Os parâmetros são bastante simples: o endereço do servidor (*host*), o nome do usuário (*login*) e a senha para a conexão. A função retorna um valor inteiro, que identifica a conexão estabelecida e deverá ser armazenada numa variável para ser utilizado depois.

Uma vez conectado, é preciso selecionar o banco de dados existente no servidor com o qual se deseja trabalhar. Isso é feito através da função `int mssql_select_db`, que possui a seguinte sintaxe:

- `int mssql_select_db(string nome_bd, int id_link)`.

O valor de retorno é 0 (zero) se o comando falhar, e 1 em caso de sucesso. O nome da base de dados a selecionar é o primeiro parâmetro fornecido, seguido pelo identificador da conexão. Se este for omitido, o interpretador PHP tentará utilizar a última conexão estabelecida. Recomenda-se sempre explicitar esse valor, para facilitar a legibilidade do código.

A execução de cláusulas SQL, após estabelecida a conexão e selecionada a base de dados a ser utilizada, quase toda a interação com o servidor MsSQL pode ser feita através de

consultas escritas em SQL (*Structured Query Language*), com o comando `mssql_query`, que utiliza a seguinte sintaxe:

- `int mssql_query(string consulta, int id_link)`.

O valor de retorno é 0 (zero) se falhar ou 1 em caso de sucesso. Sucesso aqui significa que a consulta está sintaticamente correta e foi executada no servidor. Nenhuma informação sobre o resultado é retomada deste comando, ou até mesmo se o resultado é o esperado. No caso da consulta ser um comando *SELECT*, o valor de retorno é um valor interno que identifica o resultado, que poderá ser tratado com a função `mssql_result()` e outras. A *string query* não deve conter ponto-e-vírgula no final do comando, e o identificador da conexão é opcional.

O tratamento de resultados de *query SELECT*, se dá quando ao executar uma *query SQL SELECT* através do comando `mssql_query`, o identificador do resultado deve ser armazenado numa variável que pode ser tratada de diversas formas. Uma das maneiras de fazê-lo é usar o comando `mssql_result`.

O comando `mssql_result` tem a seguinte sintaxe:

- `int mssql_result(int resultado, int i, mixed [campo])`.

Onde resultado é o identificador do resultado, obtido com o retorno da função `mssql_query`, linha especifica a tupla a ser exibida, já que uma *query SELECT* pode retornar diversas tuplas, e campo é o identificador do campo a ser exibido, sendo o tipo descrito como *mixed* pela possibilidade de ser de diversos tipos (neste caso, inteiro ou *string*).

O Método de trabalho com servidores de banco de dados:

- a) conectar ao servidor;
- b) abrir o banco de dados;

- c) enviar os comandos SQL;
- d) fechar o banco de dados;
- e) desconectar ao servidor.

Sendo que as duas últimas geralmente não precisam ser feitas explicitamente, pois quando o *script* terminar, o banco de dados é fechado e a conexão é encerrada automaticamente [DIA2000].

A consulta aos dados contidos em uma tabela é feita usando o comando *Select*. Este comando diz ao banco para separar, de todos os dados contidos em uma tabela, apenas aqueles que se precisa. Para saber quais pessoas, dentre as que preencheram o formulário, tem o primeiro nome Maria, basta executar o comando:

- `Select * (ou All) from dados where nome like Maria.`

No padrão SQL, pode-se informar como se quer a comparação do “parecido”, especificando se o que se quer começa, termina ou contém o texto a ser comparado. Como se quer saber todo mundo cujo primeiro nome é Maria, fica assim no SQL:

- `Select * from dados where nome like "Maria%".`

6. PROTÓTIPO

Como base para o estágio proposto, analisou-se a necessidade atual de uma empresa na região de Blumenau. Esta empresa atua no ramo de corretagem de seguros, gerenciamento de riscos e avaliações.

6.1 LEVANTAMENTO DE INFORMAÇÕES

Observou-se que o controle que a empresa detém atualmente se dá quase que absolutamente, a nível operacional, onde diariamente é alimentado um sistema em plataforma *windows*, com todas as informações sobre seguros, tipos de seguros, seguradoras, corretores, clientes, perfil do cliente, (sistema já existente na corretora) pertinentes ao negócio corretagem de seguros e análises de riscos, gerando um banco de dados robusto em informações de cunho gerencial e que seria de extrema relevância para uma gestão mais eficaz desta empresa.

Perante o fato, e de acordo com a metodologia de uma consultoria, que presta serviços a esta empresa, surgiu a necessidade e oportunidade de ser retirado deste banco de dados, muito mais que informações gerenciais indispensáveis. Diante da nova filosofia de gestão implementada, a gerência e a diretoria, necessitavam de uma ferramenta de apoio para melhor administrar. Verificou-se também a necessidade de assessoria ao corretor com informações atualizadas do contrato de resultados, (contrato este firmado entre a corretora e o corretor no início de cada ano, no qual contém metas e descrição do perfil de vendas) podendo mantê-lo informado do desempenho diário, quando assim necessitar-se. Esta ferramenta possibilita gerenciar estrategicamente sua conduta de vendas e mercado, como também avaliar os seus resultados perante a corretora.

Para agregar valor perante o cliente, e facilitar o fluxo de informações entre cliente e empresa, o protótipo desenvolvido contempla ainda, a prestação de serviços via Internet, no qual terá disponibilizado através de acesso ao banco de dados, informações referentes aos seus

contratos de seguros firmados com esta corretora independentemente da companhia seguradora responsável pela contratação do risco.

6.2 ESPECIFICAÇÃO

O protótipo desenvolvido propõe o gerenciamento e controle estratégico dos dados de corretores e carteiras (tipos) de seguros, atendendo ainda consultas de clientes. A parte cliente do protótipo foi desenvolvida em PHP, onde será acessado via *browser*, através da *Internet*. As páginas PHP acessam o banco de dados *SQL Server*.

Para a utilização do sistema na corretora, será necessário utilizar um computador rodando um *browser*, e estar habilitado a ter acesso ao sistema com identificação e senha cadastrada.

6.3 CARACTERÍSTICAS

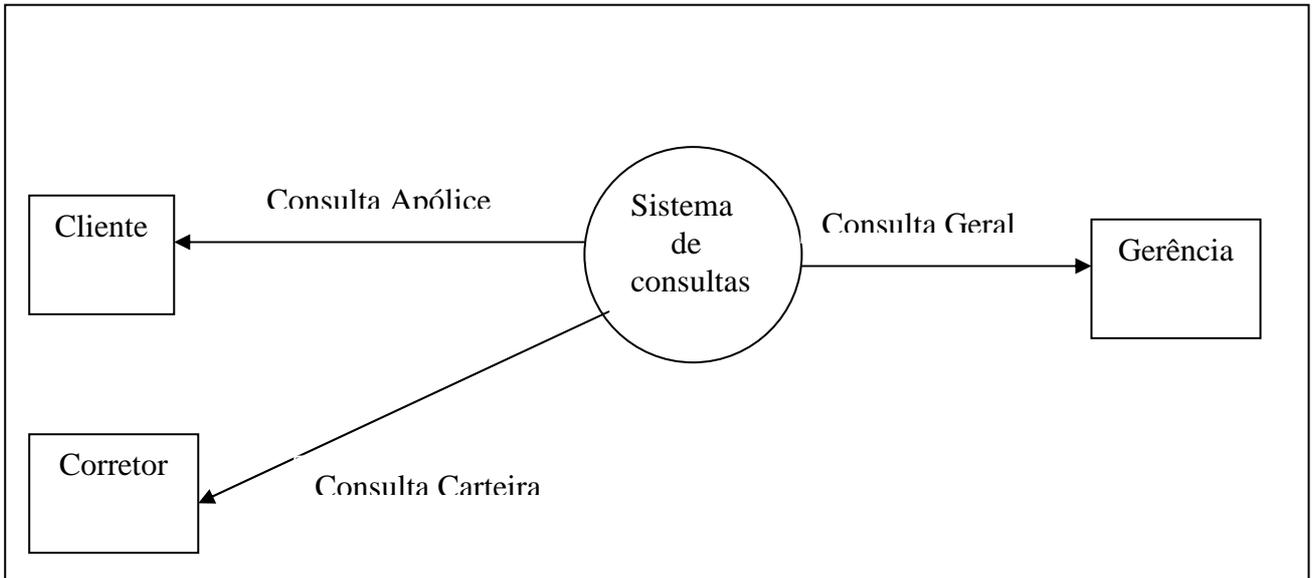
As características esperadas por esse protótipo são:

- a) rapidez no atendimento aos clientes;
- b) controle e gerenciamento a distância do negócio em corretagem de seguros;
- c) disponibilizar ao cliente acesso a consultas dos seus contratos de seguros;
- d) facilidade no uso deste sendo uma forma simples e direta.

6.3.1 DIAGRAMA DE CONTEXTO

O modelo obtido é exposto na figura 1 a seguir.

Figura 1 – Diagrama de Contexto



Com base no Diagrama de Contexto mostrado anteriormente, pôde-se construir o Diagrama de Fluxos de Dados (DFD) nível zero conforme figuras 2,3 e 4.

Figura 2 – Diagrama de Fluxo de Dados nível 0 (cliente)

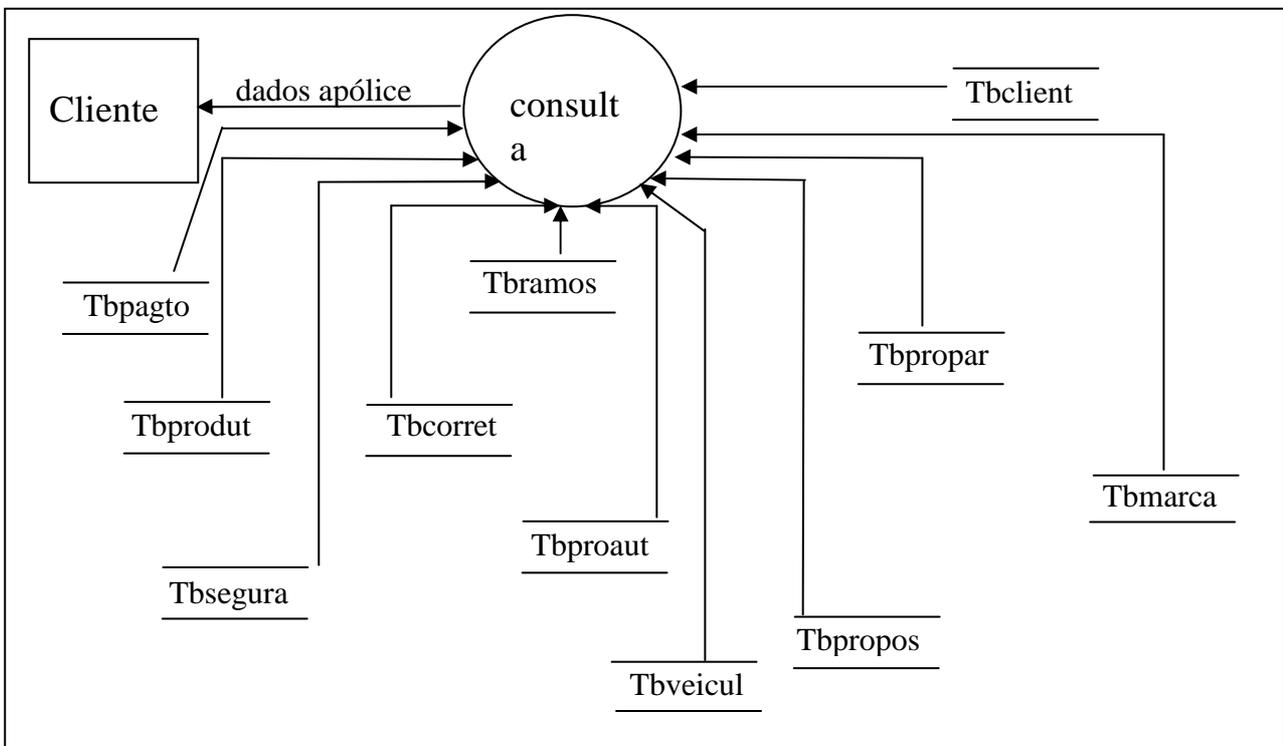


Figura 3 – Diagrama de Fluxo de Dados nível 0 (corretor)

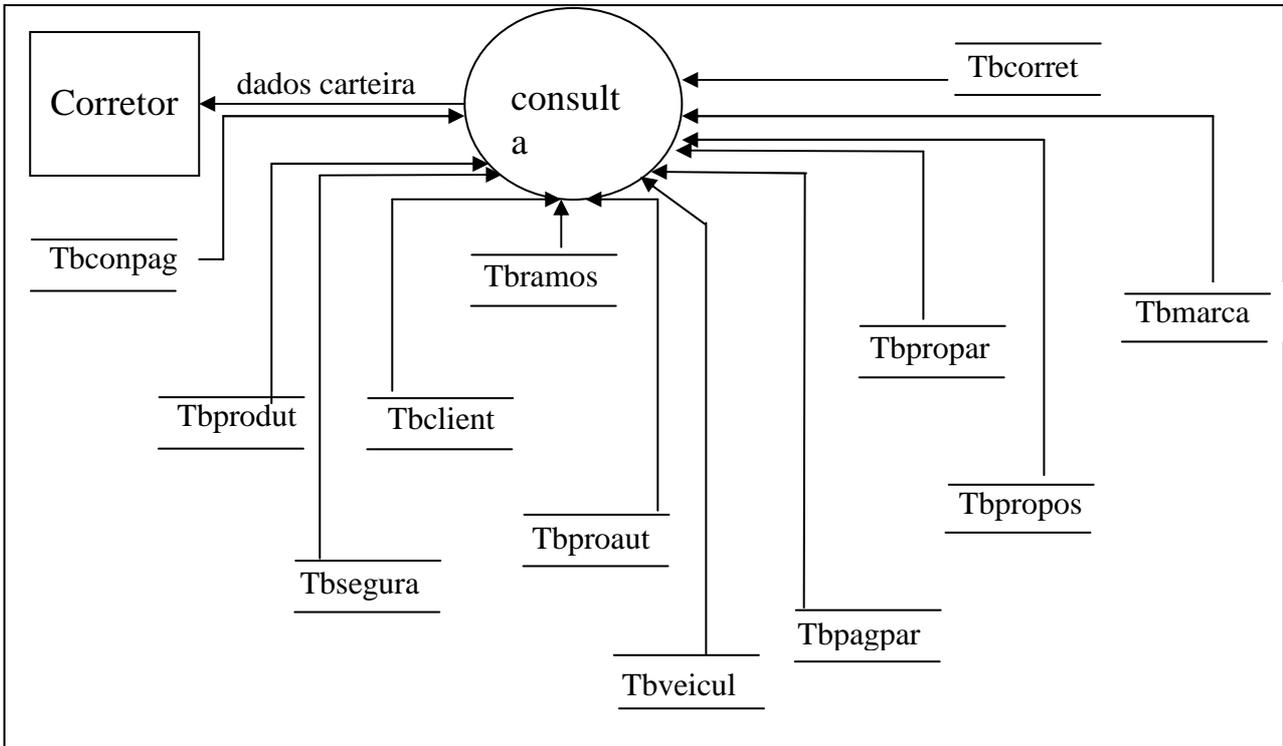
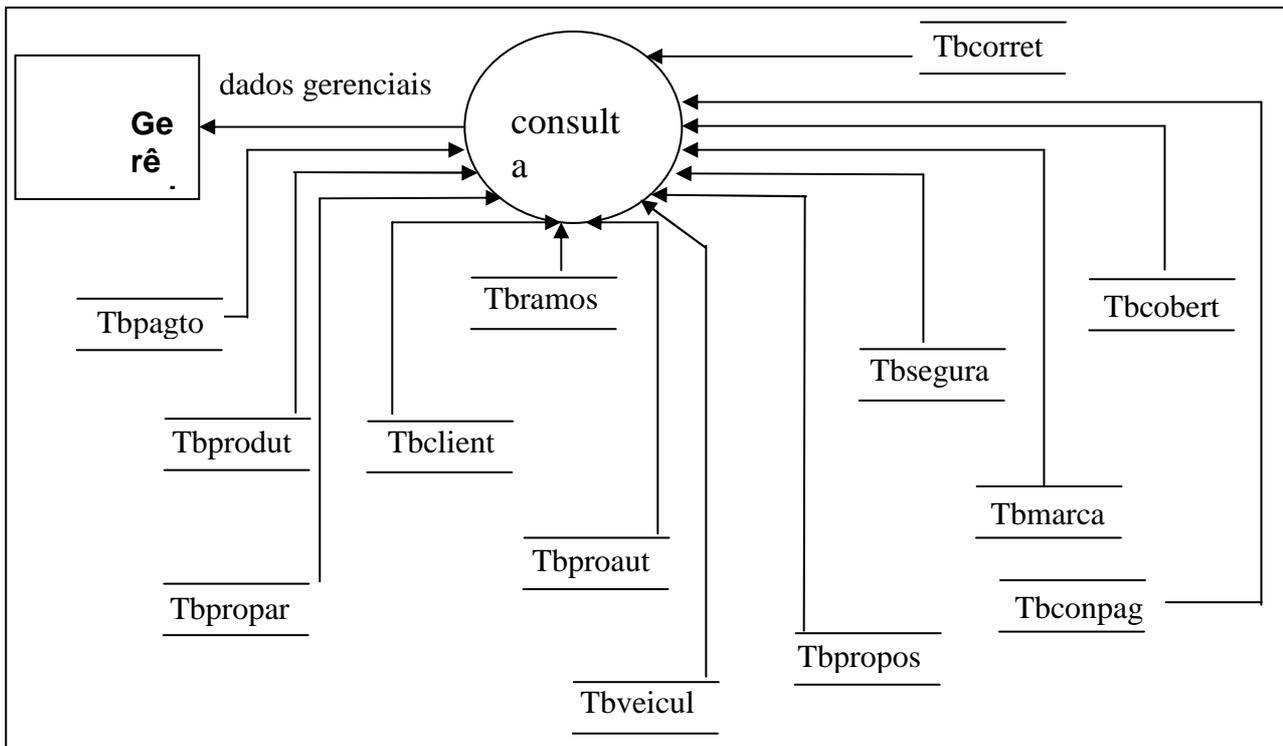


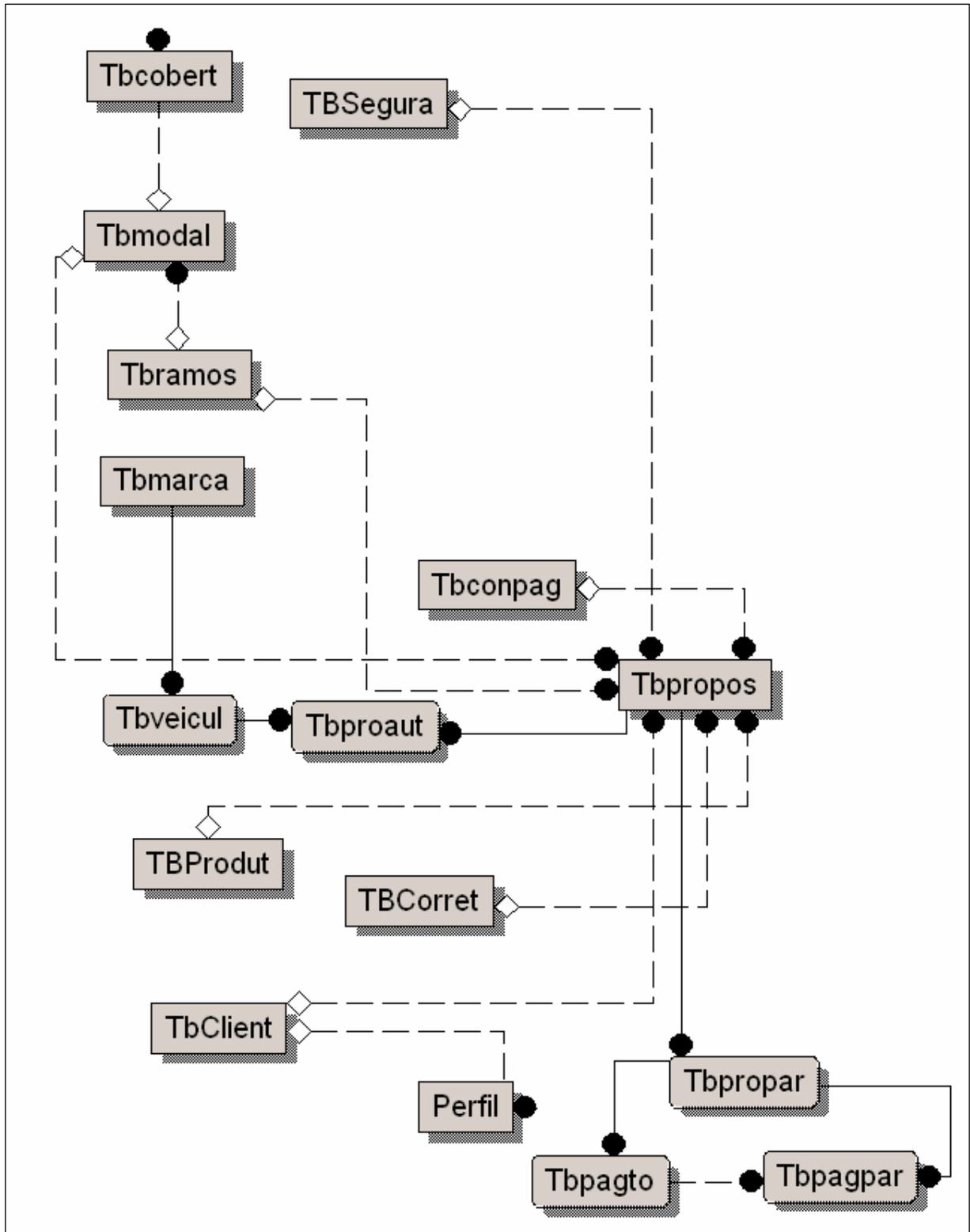
Figura 4 – Diagrama de Fluxo de Dados nível 0 (gerência)



6.4 MODELO ENTIDADE RELACIONAMENTO (MER)

Baseado nas figuras acima, obteve-se os seguintes relacionamentos, conforme mostrado na figura 5.

Figura 5 - Modelo Lógico

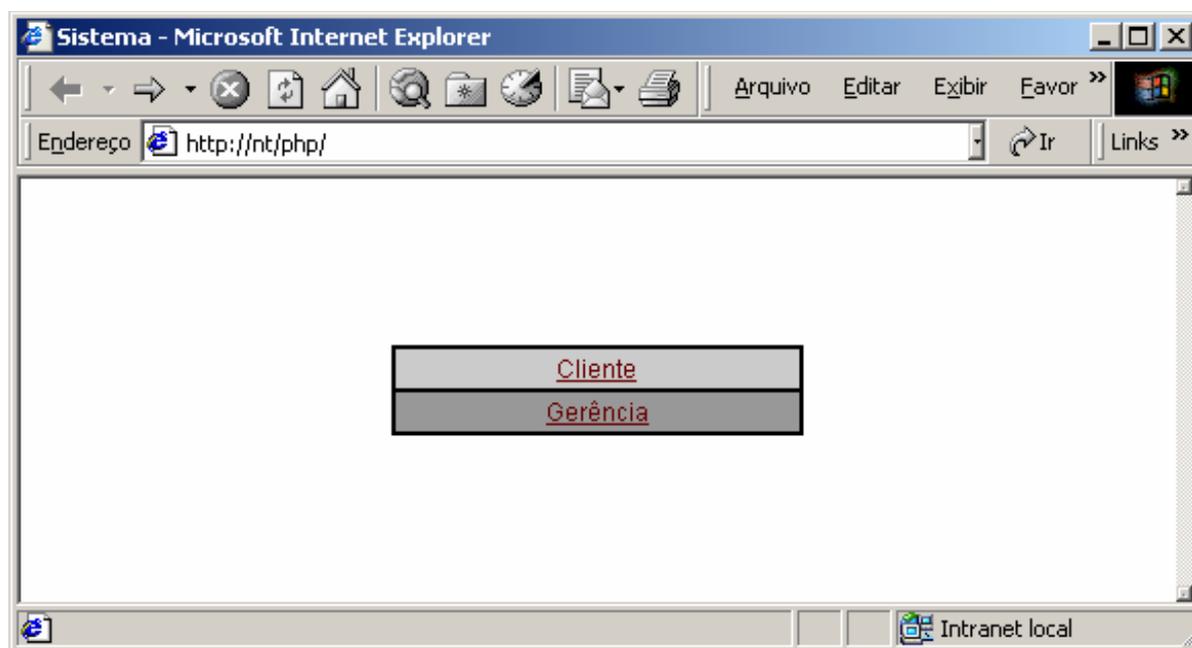


7. IMPLEMENTAÇÃO DO PROTÓTIPO

O sistema irá consultar os dados diretamente na base da empresa. Através de um *script* PHP rodando sob o servidor *WEB IIS*, é aberta uma conexão ao servidor de banco de dados *Sql Server 7.0*, que após validação das devidas permissões do usuário, liberará o acesso. Os dados serão consultados através de instruções SQL (*Structured Query Language*), e através de funções e operadores do PHP, devidamente tratados.

Nesta fase apresenta-se o protótipo da aplicação conceituada. Conforme a figura 6 a página inicial visa identificar o serviço ao seu respectivo usuário, de forma que o cliente, o corretor e a gerência/presidência da corretora possam ter informações de controle conforme sua necessidade e/ou permissão (Ver Fonte no Anexo 1).

Figura 6 - Identificação



A tela seguinte da opção "Clientes" (Figura 5) exige do usuário a digitação da chave de localização, no caso o CPF (Código Pessoal Física) ou o CNPJ (Código Nacional de Pessoa Jurídica) como mostra a Figura 6. Uma vez validado o CPF/CNPJ o sistema irá disponibilizar informações referentes a todos os

respectivos contratos de seguros, mostrados na Figura 6 (Ver Fonte no Anexo 2).

Figura 7 - Entrada da chave de localização

Digite CPF/CNPJ (somente os números) para entrar no sistema	
CPF/CNPJ:	<input type="text"/>
	<input type="button" value="Entrar"/>

Figura 8: Resultado da pesquisa por CPF/CNPJ

Informações do cliente:	
Código do cliente:	1981
Nome do segurado:	FABIO AUGUSTO SIQUEIRA
CPF/CNPJ:	01576361942
Endereço:	RUA RICARDO LANDMANN, 264 AP 101 BL B
Bairro:	SANTO ANTONIO
Cidade:	JOINVILLE
Estado:	SC
CEP:	89218200

Informações das apólices (clique nos links para ver detalhes):	
Número de apólices do segurado:	1
Número das apólices:	4503082

Conforme Figura 9, o usuário obterá sua identificação e informações a respeito de sua apólice selecionada na tela anterior (Figura 7) ou seja, a tela exibirá dados referentes a identificação do contrato de seguro, como número da proposta, número da apólice, seguradora responsável pelo risco contratado, parcelamento do prêmio e suas respectivas datas de vencimento, corretora e corretor responsável pela realização deste contrato e ainda, vigência (período de cobertura) (Fonte está no Anexo 3).

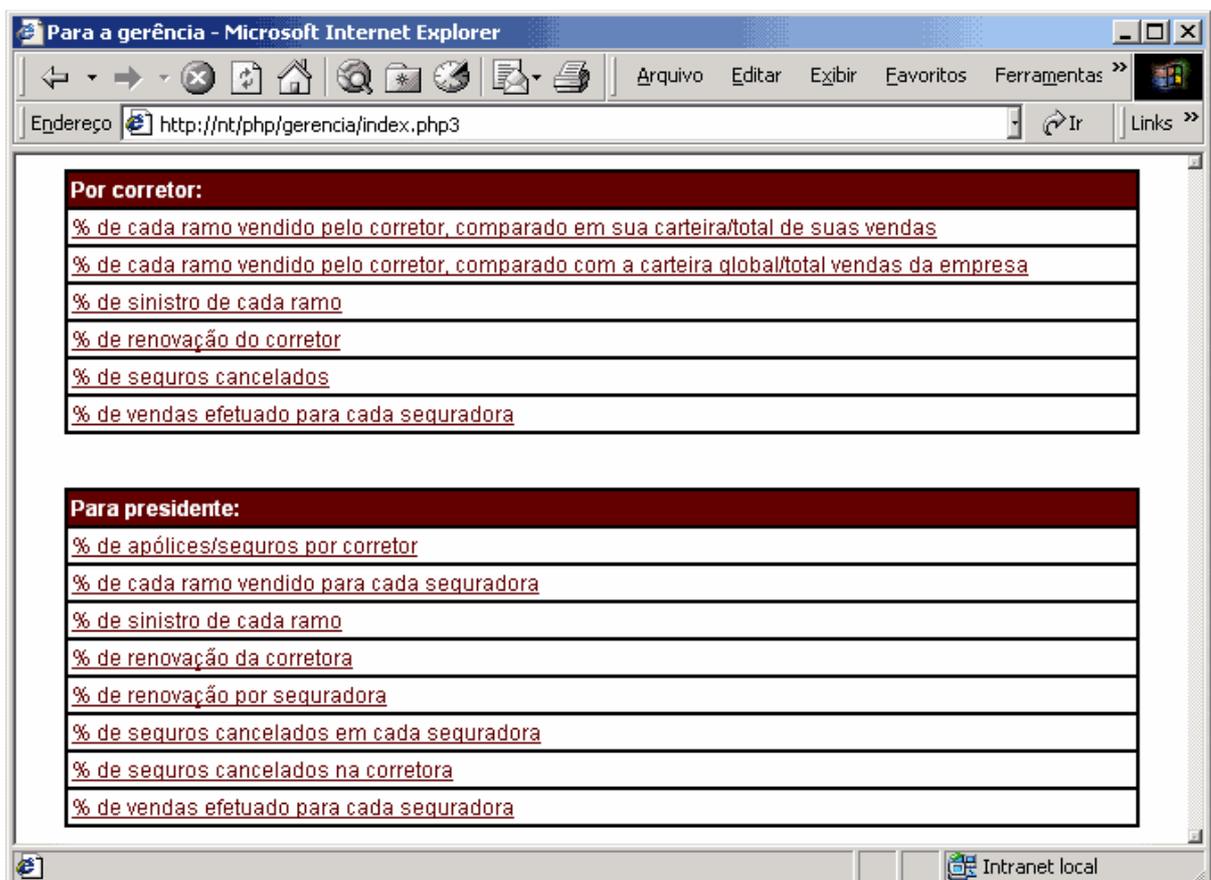
Figura 9 - Detalhamento da apólice

APÓLICE:	
Proposta nº:	643006583
Apólice nº:	4503082
Seguradora:	SASC-SUL AMERICA SANTA CRUZ
Corretora:	ADD/MAKLER ADM CORR DE SEGUROS LTDA
Corretor:	ALEXSSANDER MEDEIROS
Vigência:	De 24/10/2000 até 24/10/2001
Veículo:	FORD - ESCORT L (BASE) - Gasolina - Ano 1990
Chassi:	9BFZZZ54ZLB106887
Placa:	MBQ-2720
Prêmio:	608,63
Parcelamento:	Parcela 1: R\$ 155,65 - Vencimento: 03/11/2000 Parcela 2: R\$ 55,08 - Vencimento: 05/12/2000 Parcela 3: R\$ 55,06 - Vencimento: 04/01/2001 Parcela 4: R\$ 55,06 - Vencimento: 05/02/2001 Parcela 5: R\$ 55,06 - Vencimento: 05/03/2001 Parcela 6: R\$ 55,06 - Vencimento: 04/04/2001 Parcela 7: R\$ 55,06 - Vencimento: 04/05/2001 Parcela 8: R\$ 55,06 - Vencimento: 04/06/2001 Parcela 9: R\$ 55,06 - Vencimento: 03/07/2001 Parcela 10: R\$ 55,06 - Vencimento: 02/08/2001
Entre em contato:	Fone/Fax: 326 1002 Internet: www.addmakler.com.br

Este acesso estará disponível somente para usuário habilitado e identificado pela empresa, pois nesta seleção, o corretor detentor e responsável por uma parcela de clientes dentro da corretora de seguros, poderá avaliar seu desempenho. E a gerência/presidência terá controle dos dados diariamente atualizados, referente a seguradora, mercado, condições de mercado da corretora, desempenho do seu corretor individualmente e também seguradora.

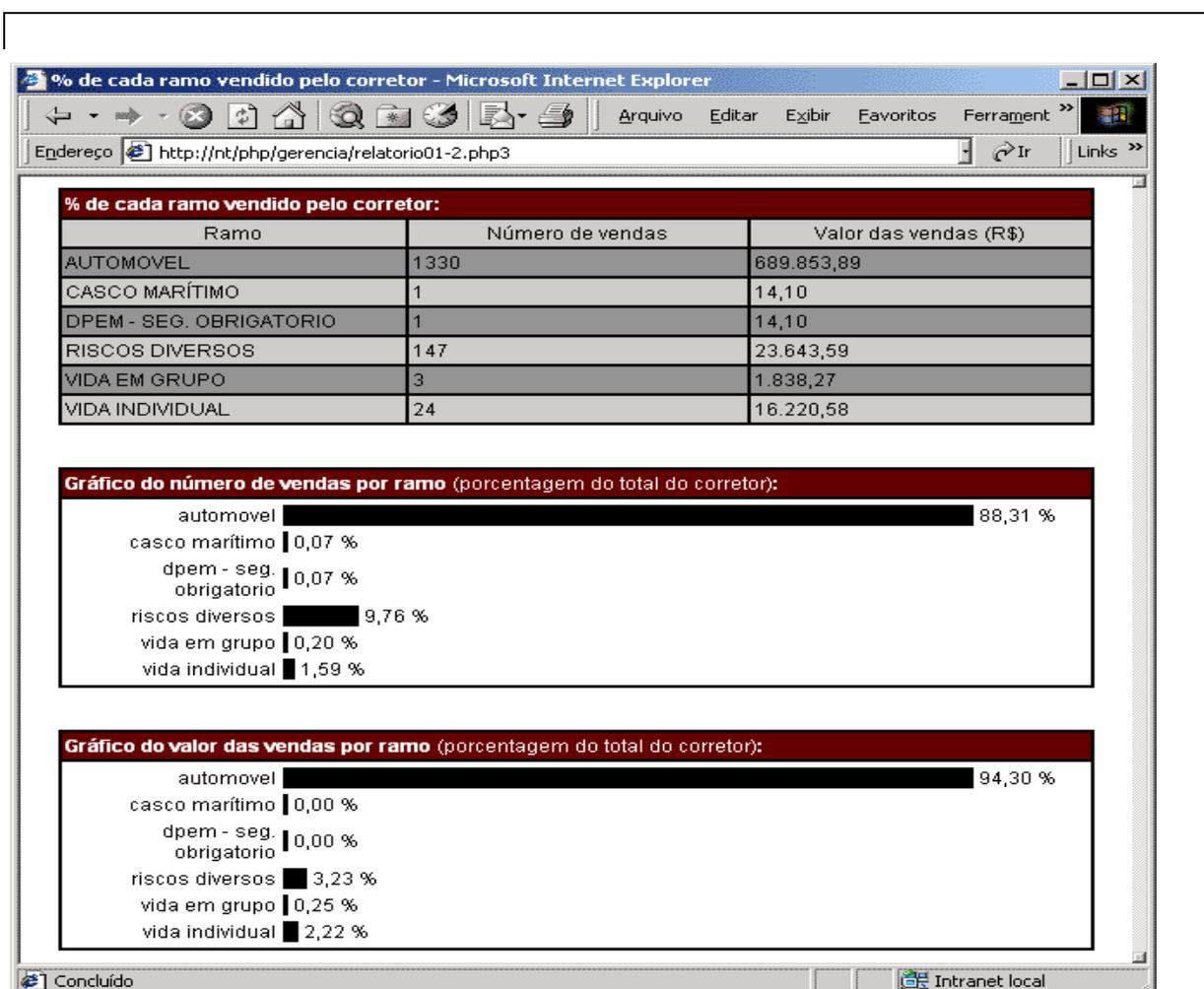
De acordo com a Figura 10, ele poderá gerenciar seu perfil de vendas e avaliar também resultados referente as mesmas, e localizar-se dentro da estrutura organizacional da empresa e terá acesso a informações até então restritas a um nível gerencial, e antes disponibilizada algumas vezes ao ano para sua avaliação de desempenho e cumprimento de metas.

Figura 10 - Consultas gerenciais



De acordo com a Figura 11, o corretor poderá acompanhar e gerenciar suas metas de vendas através da informação abaixo gerada, que disponibiliza ainda não só o percentual de vendas por produto por ele vendido, como também identifica seu mercado, valor, número e posicionamento. Informações estas que o permitirão administrar de forma estratégica seu crescimento ou única e exclusivamente seu resultado.

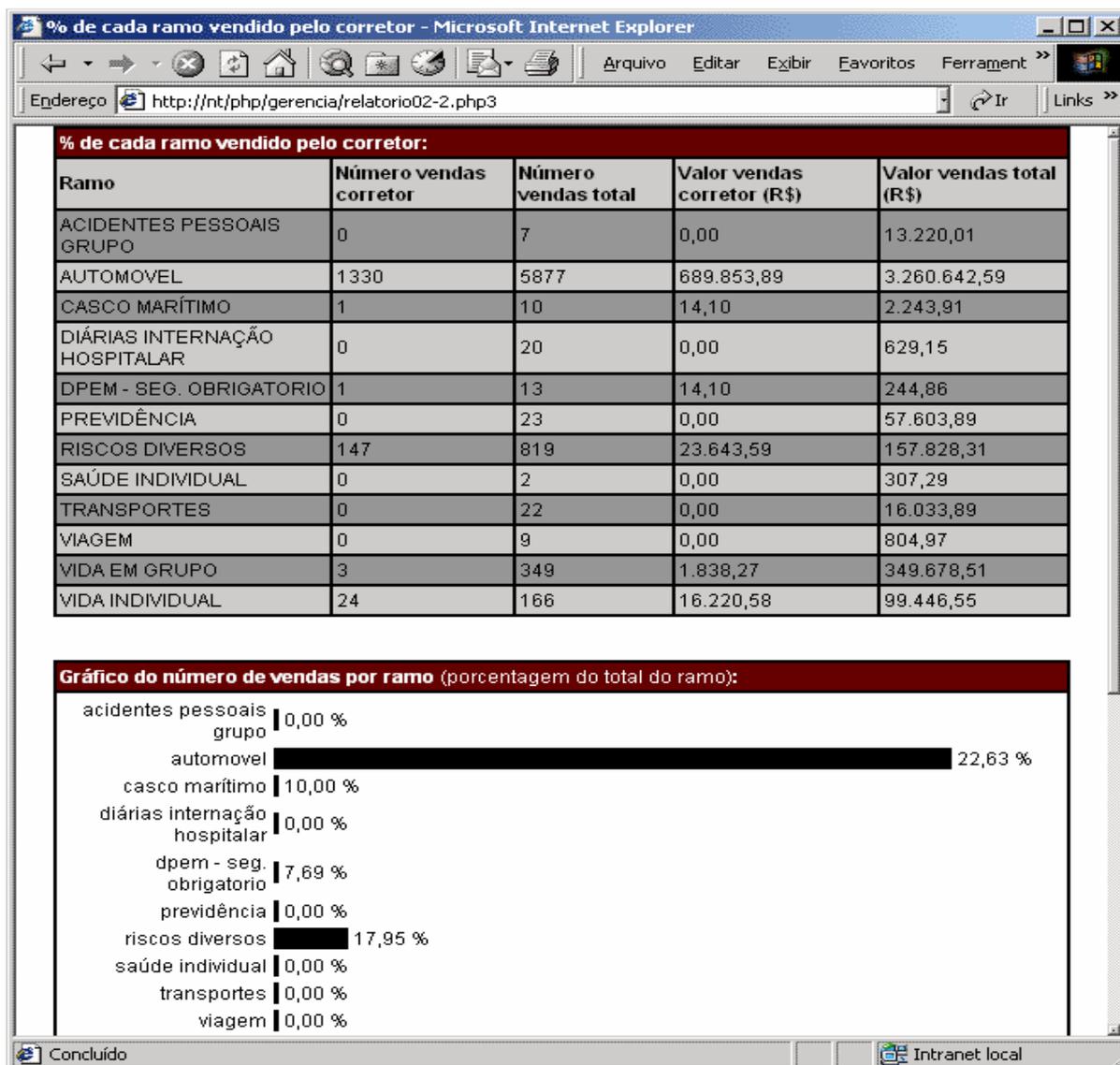
Figura 11 - Porcentagem de vendas do corretor - tela 1



Em uma forma mais abrangente, na figura 12, o corretor terá acesso a sua posição dentro da estrutura de vendas da corretora de seguros. Permite-se ainda que o corretor gerencie suas metas e acompanhe o ranking de vendas geral, através da informação abaixo acessada, que disponibiliza ainda não só o percentual de vendas por produto de cada corretor, como também

identifica seu mercado, o potencial de vendas de cada mercado no momento, valor e posicionamento no total geral da estrutura, comparando seu perfil com os demais corretores cadastrados.

Figura 12 - Porcentagem de vendas do corretor - tela 2



Nesta etapa (Figura 13) será possível avaliar os tipos (ramos) de seguros de maior risco, ou seja, determinado ramo, que proporcionalmente poderá comprometer o resultado estimado. Uma das avaliações de medição do grau de risco é sua incidência, outra é a sua proporcionalidade com o prêmio pago pelo cliente (segurado) para a companhia seguradora. Outra variável interessante de avaliação de desempenho do corretor é seu índice e documentos cancelados, independentemente do motivo corrente.

Figura 13 - Ramos de seguros de maior risco

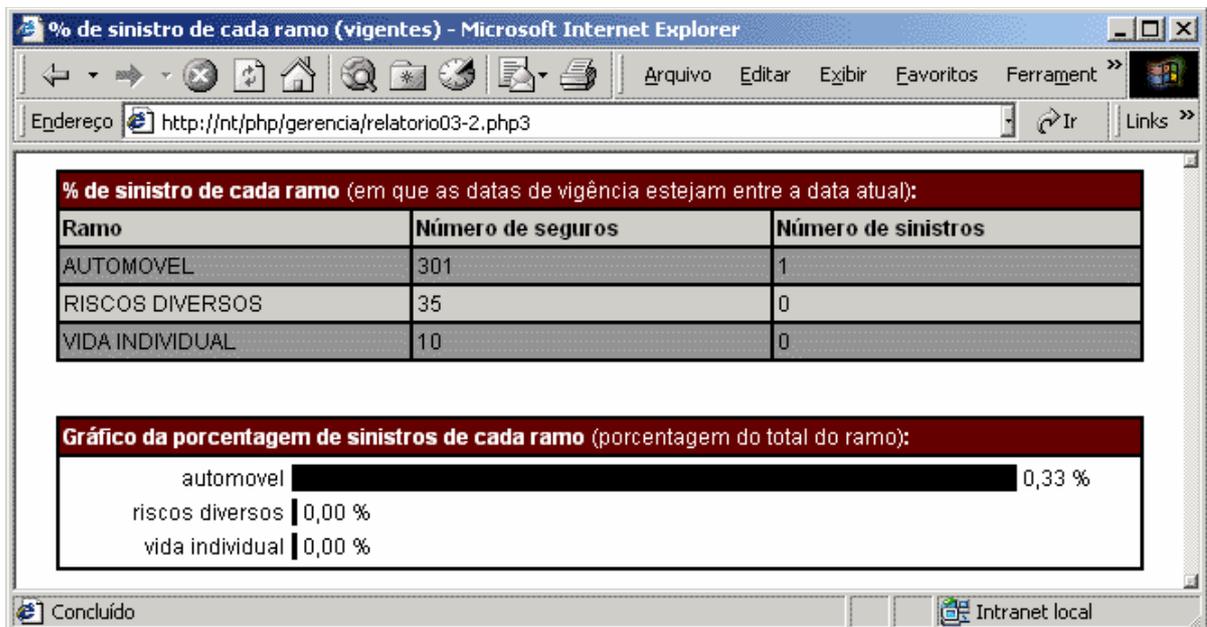
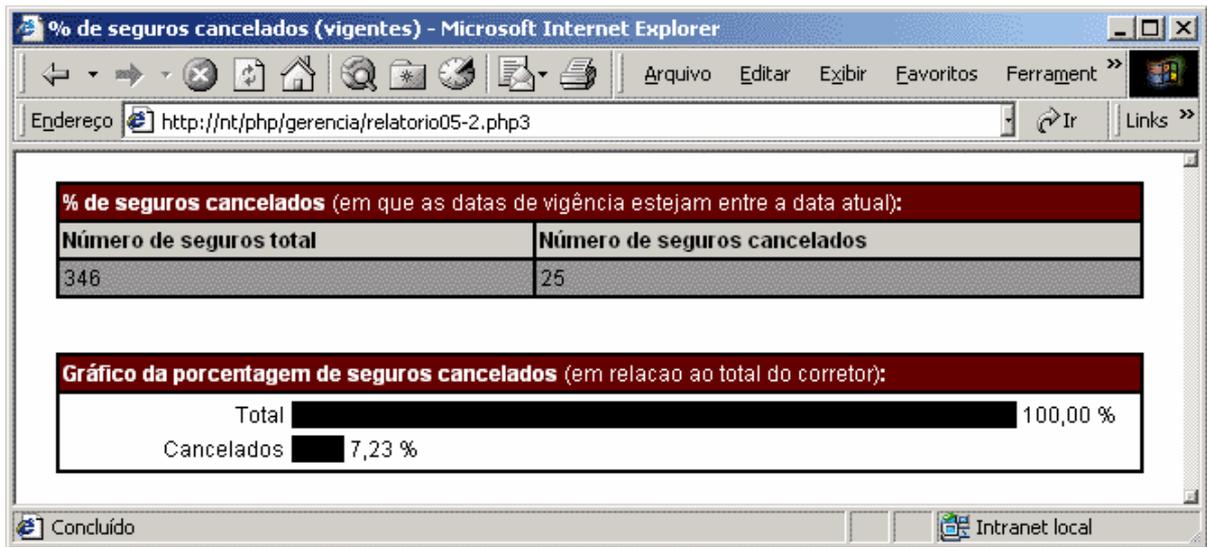
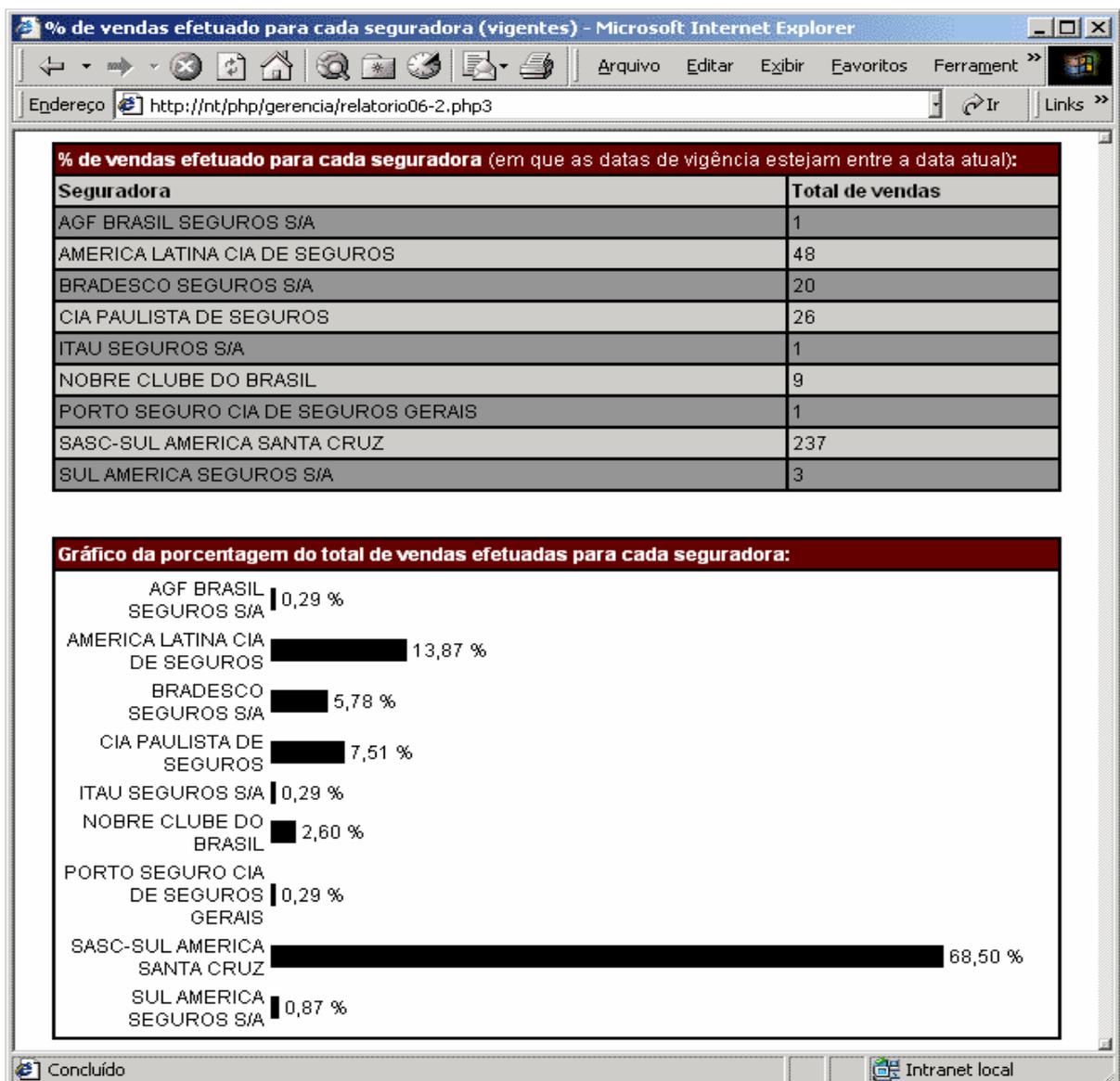


Figura 15 - Porcentagem de seguros cancelados



Nesta etapa da consulta (Figura 16), o corretor terá acesso a condição comercial que as seguradoras oferecem a corretora naquele instante. Como já conhece os produtos disponibilizados por cada companhia, esta informação o direcionará para um foco de vendas, comercial, bastante apurado, informando-o de que neste período a comercialização estratégica está sendo realizada com determinada companhia.

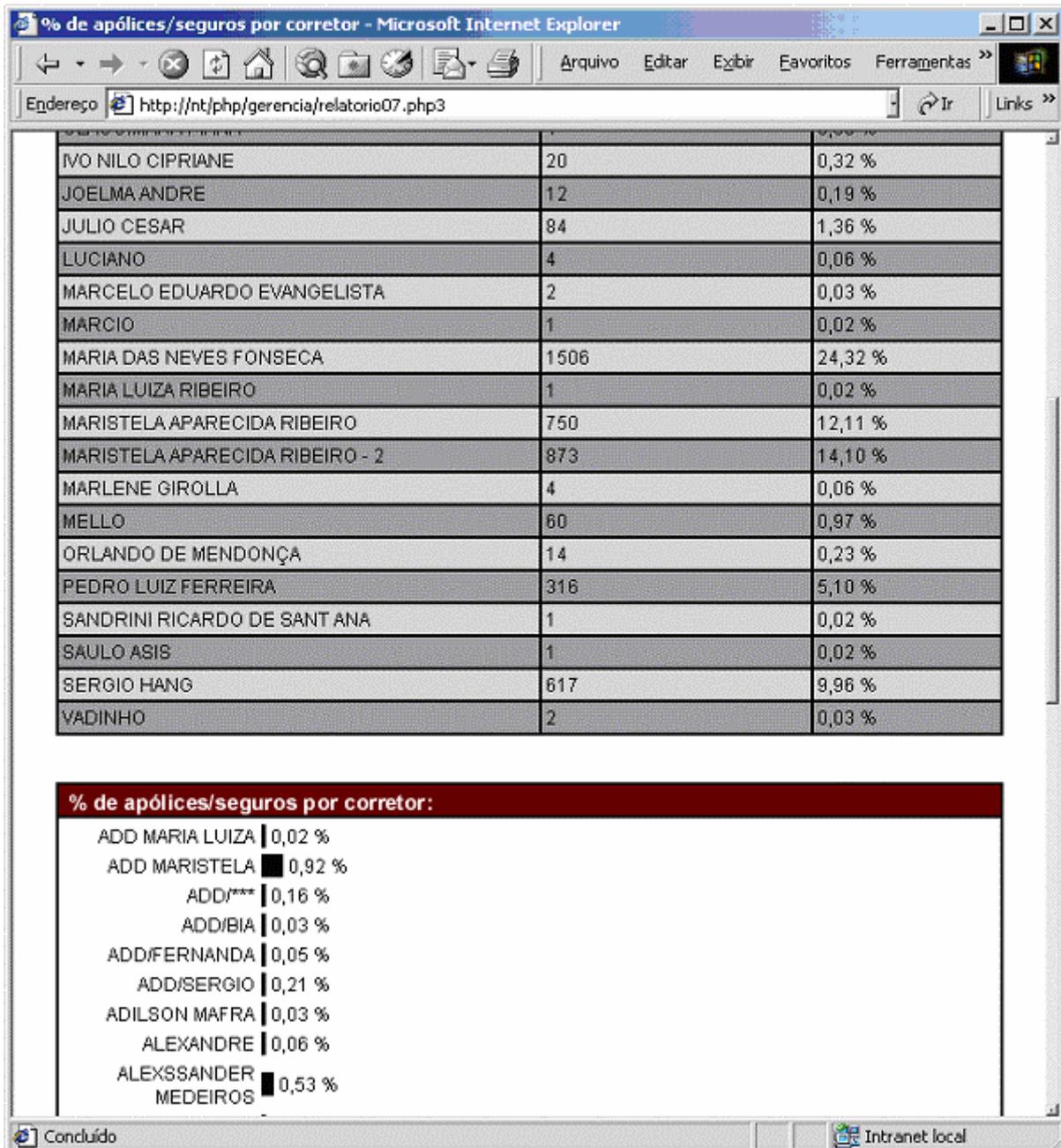
Figura 16 - Porcentagem de vendas de cada seguradora



De acordo com a opção: “GERÊNCIA”, conforme Figura 8, somente terá acesso gerência/presidência aos dados que serão acessados.

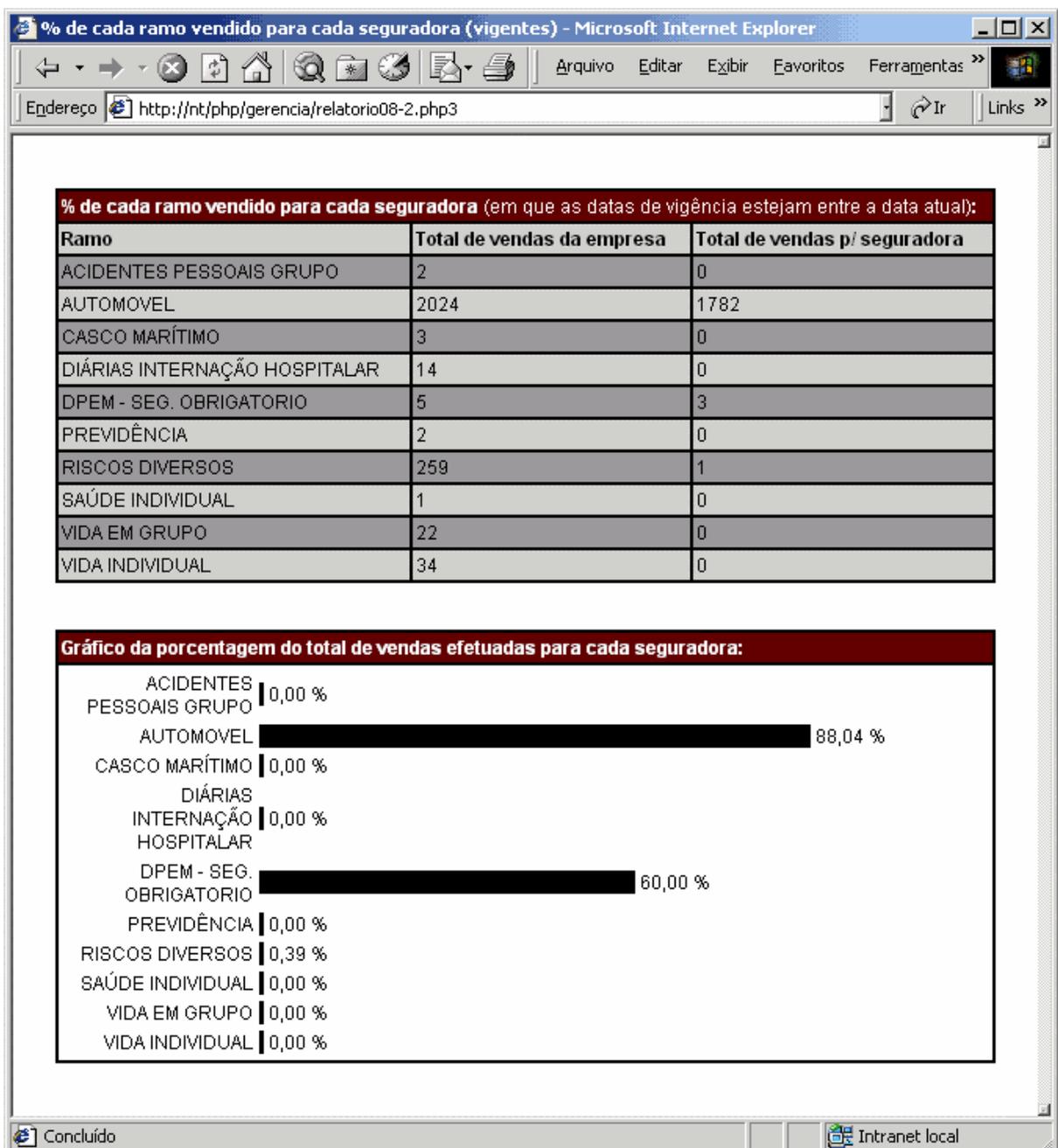
Nesta tela (figura17) será visualizado o percentual de seguros por corretor.

Figura 17 - Porcentagem de apólices



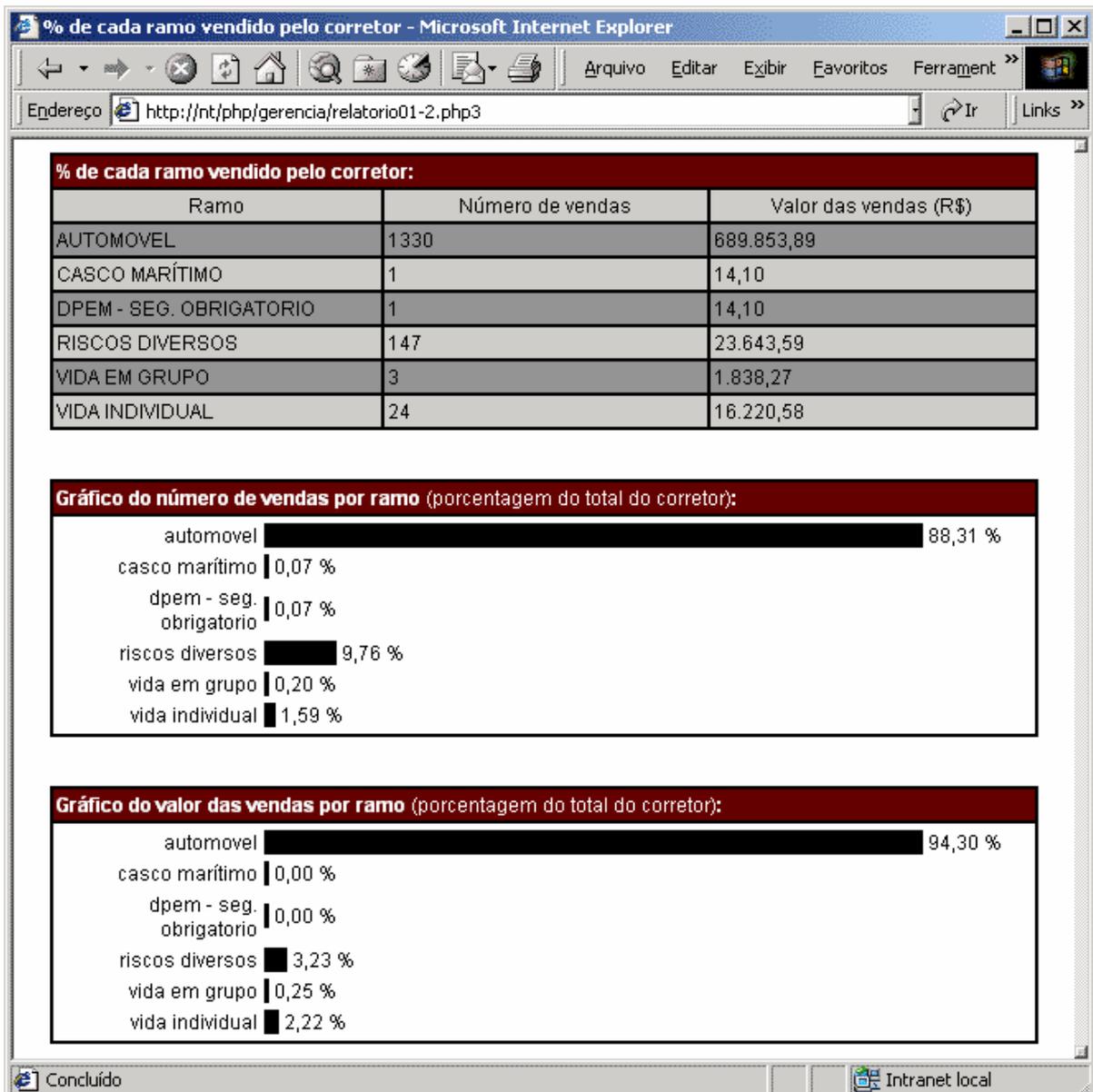
Nesta fase da consulta (Figura 18), a gerência terá acesso a condição comercial que as seguradoras oferecem à corretora naquele instante. Como já conhece os produtos disponibilizados por cada companhia, esta informação dará conhecimento do foco de vendas, comercial, que os corretores estão aplicando, e se está de acordo ou não com a meta estabelecida.

Figura 18 - Porcentagem de cada ramo vendido para cada seguradora



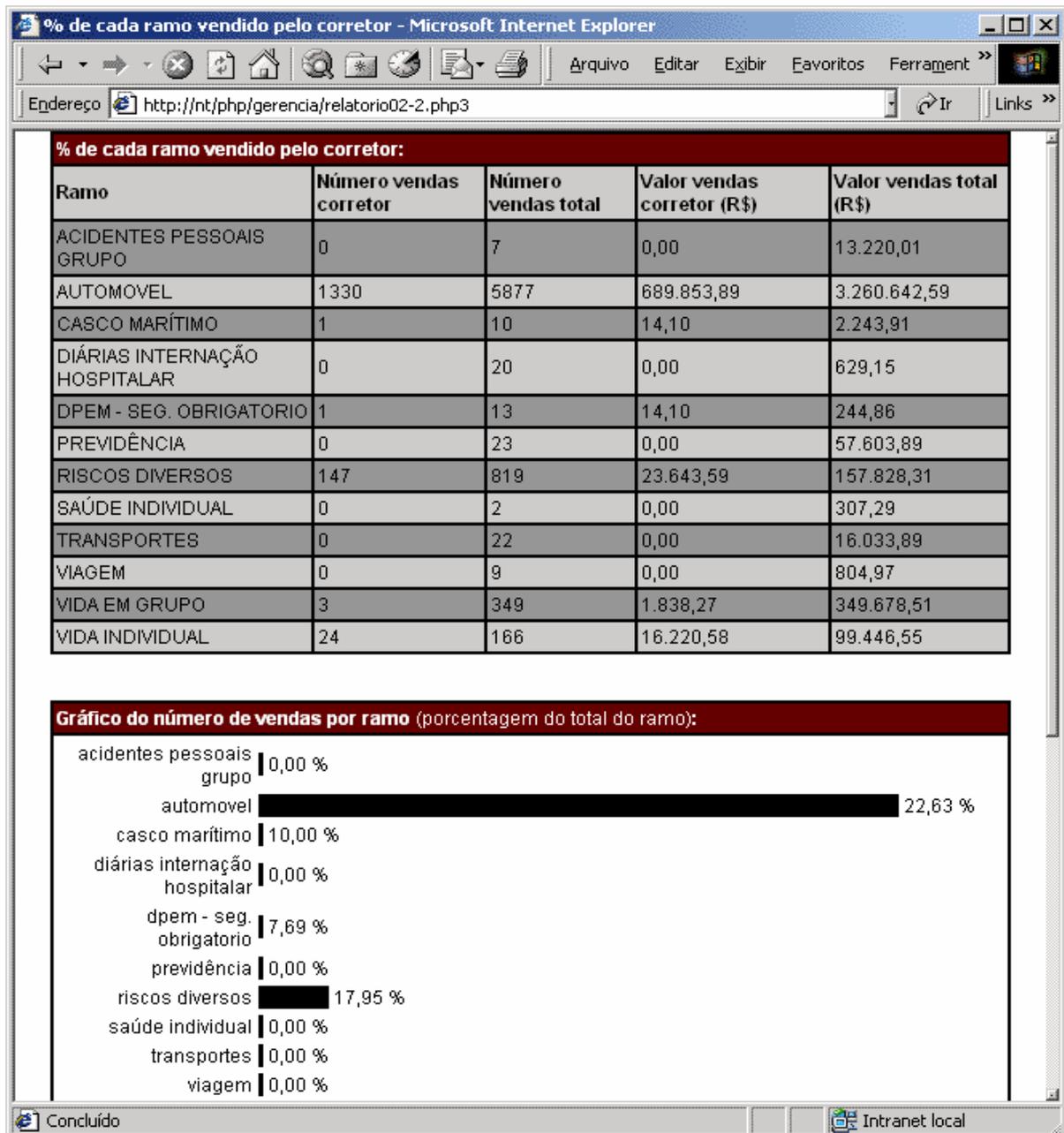
Nesta tela (Figura 19), será disponibilizada, a gerência, o número, valor e percentual de vendas em cada ramo (tipo de seguro). Onde o corretor e a gerência terão acesso aos dados correspondentes a vendas, em cada segmento e de cada corretor, permitindo a avaliação dos ramos e produtos com maior aceitação no mercado, e a expectativa de vendas correspondente às metas estabelecidas.

Figura 19 - Porcentagem de cada ramo vendido pelo corretor - Tela 1



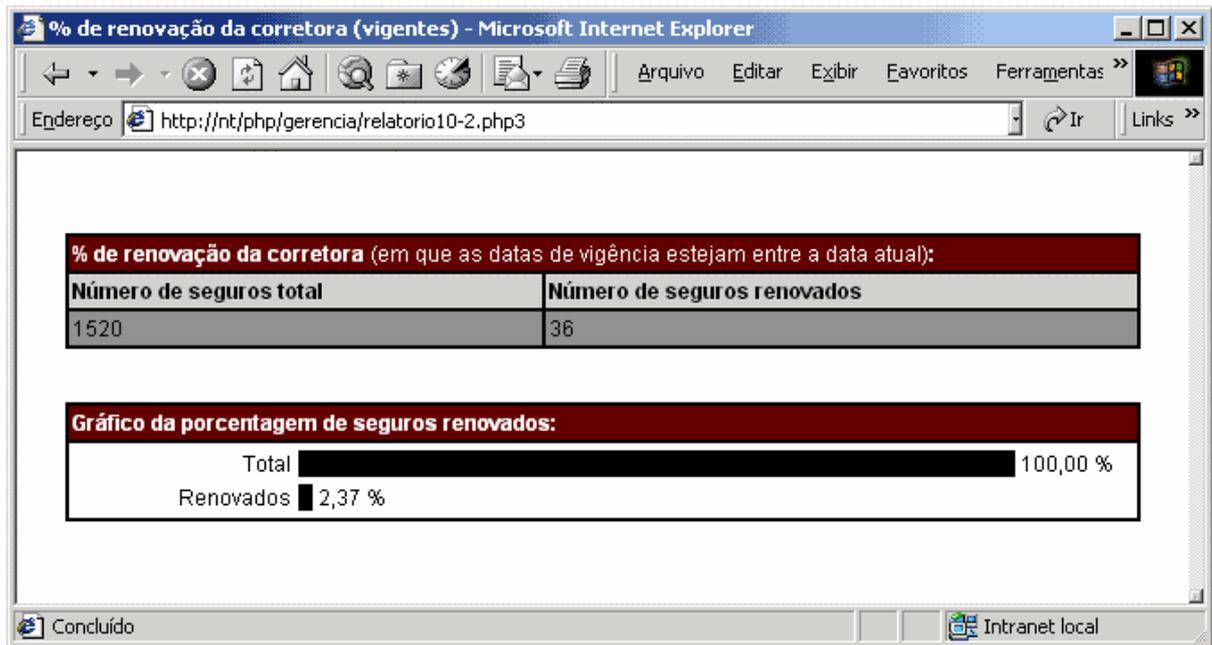
Esta tela difere da anterior (Figura 19) pelo fato de que (Figura 20) a gerência poderá além de todas as informações anteriores, vislumbrar a posição de cada corretor, por percentual de vendas em cada ramo, dentro do total geral da estrutura da corretora, permitindo a avaliação do cumprimento de metas e resultados, e também a identificação de onde se encontra o grande nicho do mercado.

Figura 20 - Porcentagem de cada ramo vendido pelo corretor - Tela 2



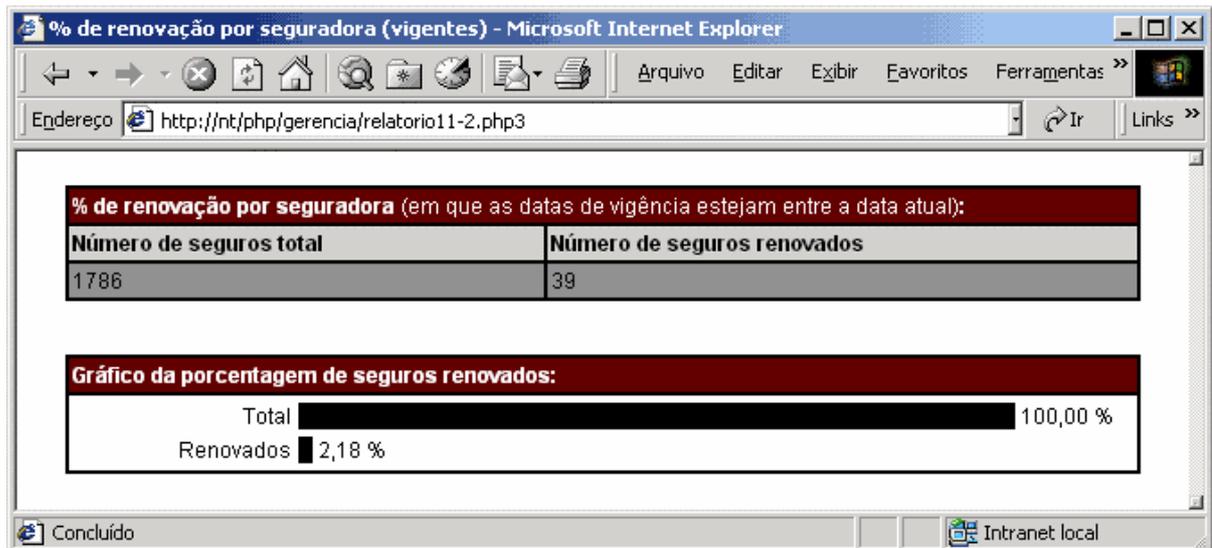
Nesta etapa da consulta, (Figura 21) a gerência verificará o nível de fidelização da carteira de clientes de cada corretor através do índice de renovação, e terá também o índice geral da corretora, Indicando quantos clientes, continua fazendo seguros com a corretora e corretor.

Figura 21 - Porcentagem de renovação da corretora



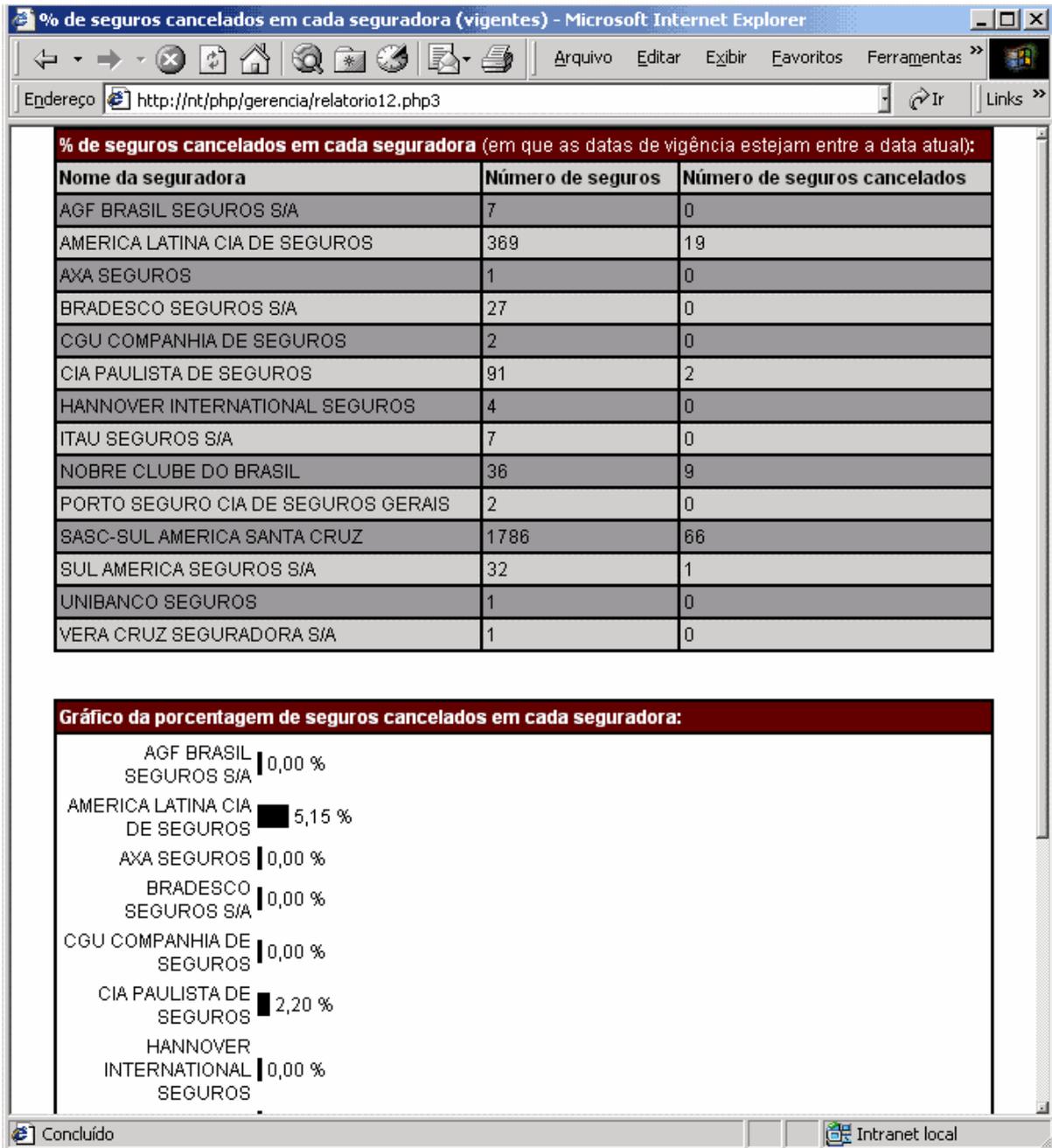
Na Figura 22, pode-se vislumbrar se a seguradora não é um dos agentes responsáveis pela não renovação e vice-versa. Poderá ser observado se há uma migração de cia seguradora, como anteriormente citado, pelo motivo de condições comerciais que pode mudar de uma em um ano, para outra no ano seguinte quando acontece a renovação.

Figura 22 - Porcentagem de renovação por seguradora



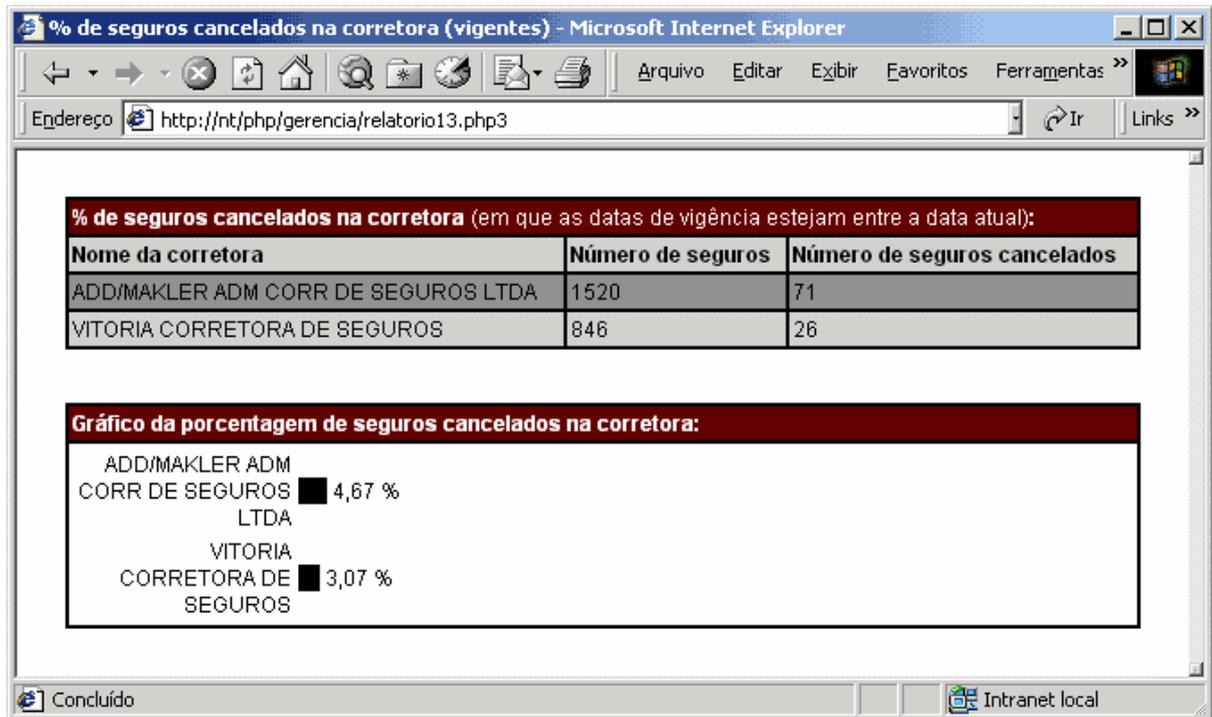
Através deste acesso (Figura 23), será obtido o índice de cancelamento de contratos de seguros, por seguradora.

Figura 23 - Porcentagem de cancelamentos por seguradora



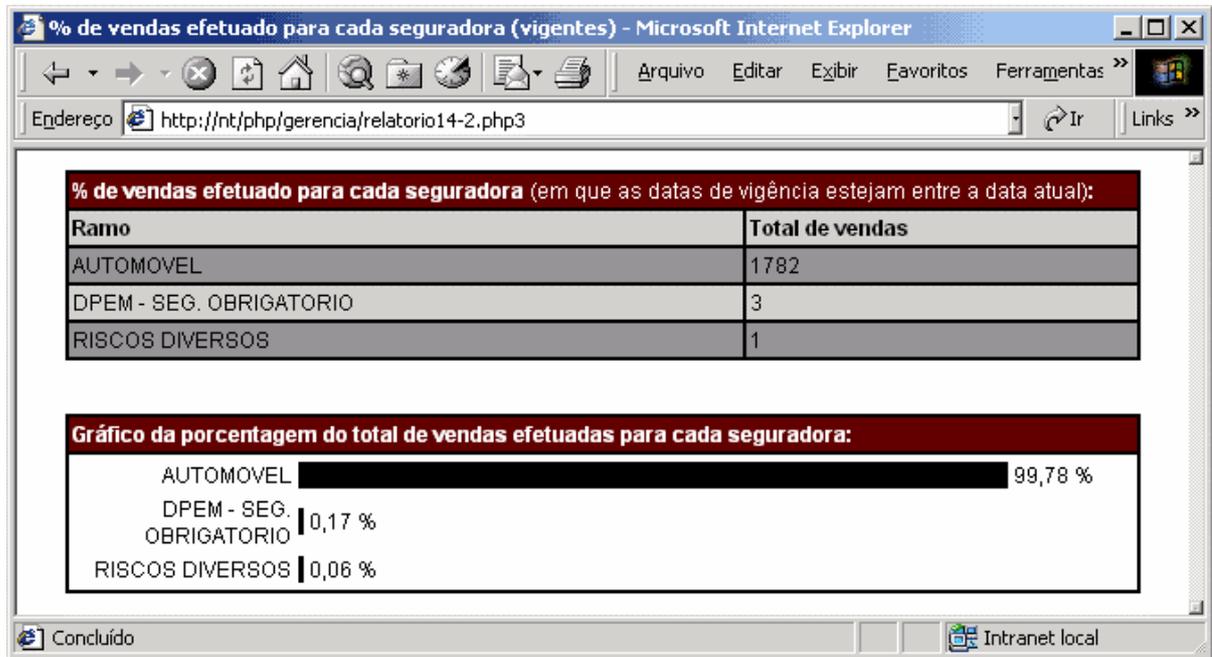
Nesta tela, figura 24 será exibido o índice total de cancelamento de corretora.

Figura 24 - Porcentagem total de cancelamentos por seguradora



Nesta página, estará disponível o total geral vendido por ramo de seguros para cada seguradora. (Figura 25)

Figura 25 - Porcentagem de vendas por seguradora



8. CONCLUSÃO

A realização deste estágio possibilitou o estudo da linguagem PHP, que se mostrou uma ferramenta poderosa e simples de utilizar na construção de *sites* para a *World Wide Web* dinâmicos, possibilitando uma maior interação com o usuário e a armazenagem das informações em Bancos de Dados.

Após a conclusão da aplicação, tornou-se claro que a combinação de *scripts server-side*, como é o PHP, com *scripts client-side*, possibilita um maior aproveitamento dos recursos disponíveis para criar páginas dinâmicas, e no processo de criação deve-se ponderar bastante para concluir qual dos dois tipos de *scripts* deve ser utilizado para determinado fim.

Entre as linguagens de *script server-side*, o PHP surgiu como uma ótima opção, por diversos motivos: o custo de aquisição, que não existe; a portabilidade, permitindo que uma aplicação seja desenvolvida em uma plataforma para ser executada em outra; a simplicidade, já que os *scripts* ficam no próprio código HTML, e possuem uma sintaxe bastante simples; a possibilidade de trabalhar com diversos bancos de dados e servidores *http*, além do grande número de funções pré-definidas, entre outras coisas.

<p>O uso do IIS, foi importante devido a utilização de uma infra-estrutura já instalada na empresa ADDmakler.Herco, desta forma dispensou qualquer alteração na solução de <i>Web Server</i> existente da empresa. A utilização do <i>SQL Server 7.0</i> como Sistema de Gerenciamento de Banco de Dados foi crucial para a validação do protótipo, pelo fato de que a aplicação que coleta todas as informações utilizadas no desenvolvimento do protótipo estar constituída nesta solução.</p>
--

O desenvolvimento do estágio propiciou academicamente o estudo, aprofundando em uma solução recente, mas, também já largamente utilizada na área da computação. Profissionalmente possibilitou o conhecimento de uma nova filosofia de trabalho, na qual exige-se comportamentos e fluxos de trabalho otimizados e precisos, que torna-se a cada dia mais imprescindível na gestão de um negócio.

8.1 LIMITAÇÕES E PROBLEMAS

Uma das dificuldades para a realização do trabalho desenvolvido no estágio, foi conciliar a concentração de tempo entre estágio e atividades profissionais, outra foi entender os conceitos vinculados a Internet, realidade até então superficial e teórica. Quanto a linguagem PHP, além de ter que adquirir o conhecimento, a dificuldade foi fazer a comunicação com o gerenciador de Banco de Dados SQL Server (que exigiu conhecimento em SQL Server).

8.2 SUGESTÕES

Como sugestão de continuidade, pode-se preparar o sistema para a entrada de dados, fazendo assim, não apenas consultas na corretora, como também viabilizar a realização de orçamentos de seguros on line. Migrar a aplicação para a linguagem ASP, nativa no IIS. Implementar rotinas de segurança. Aperfeiçoar a interface, especialmente nos itens de navegabilidade.

GLOSSÁRIO

Add-ons: é um programa escrito para trabalhar em conjunto com um serviço on-line ou prover maior funcionalidade ao browser

Backup: rotina de segurança dos dados em locais e/ou mídias alternativas.

Browser: programas que permitem visualizar textos e imagens através da rede. Ex: Netscape.

Cliente: no contexto cliente/servidor, um cliente é um programa que pede um determinado serviço a um servidor, outro programa. O cliente e o servidor podem estar em duas máquinas diferentes, sendo esta a realidade para a maior parte das aplicações que usam este tipo de interação.

Cluster: unidade para armazenamento do disco, a menor unidade de armazenamento

Cookies: arquivos utilizados para armazenar informações na máquina cliente.

Databases: banco de dados.

Download: é um mecanismo que permite que se pegue um arquivo de qualquer lugar na rede, onde ele esteja disponível, e traga-o para o seu micro, ficando com uma cópia dele.

Default: padrão, o sugerido ou atribuído pelo sistema sem que nada seja configurado.

E-business: comércio e/ou negócio eletrônico.

Full time: todo o tempo, sem interrupção.

Home page: página de apresentação. Pode ser pessoal, comercial, etc. Ao se acessar um *site* qualquer ligado a rede, a *home page* desse *site* é a primeira coisa a ser vista. Ela deve fornecer respostas para as perguntas: o que, como, quem, para quê, por quê e quando.

HTML: HTML é a sigla para *HyperText Markup Language*. A HTML é uma linguagem de formatação de hipertextos. É o recurso básico para a construção de páginas para a WWW, pois permite que a multimídia flua numa página através de comandos especiais (*tags*). Pode-se colocar, numa página: sons, imagens, desenhos e textos.

Interface: é um meio de comunicação entre dois itens diferentes. Por exemplo, entre um usuário e um programa. A interface com o usuário se compõe dos meios pelos quais um programa se comunica com o usuário, incluindo uma linha de comandos, menu, caixas de diálogo, sistemas de ajuda *on-line*, etc. As interfaces com o usuário podem ser classificadas como baseadas em caracteres (texto), baseadas em menus ou gráficas.

Internet: é o nome dado a coleção de computadores interconectados espalhados pelo mundo.

Link: conexão ou ligação entre dois documentos na *Web*.

Login: credenciais utilizadas para acessar o sistema.

Mbytes: *Mega Bytes*, unidade de medida do volume de dados em mídia física.

Navegar: na *Internet* significa procurar informação na WWW.

Protocolo: é o idioma de comunicação entre computadores. Existem três tipos de protocolo: Protocolo de rede (IP) — conduz a mensagem; protocolo de transporte (TCP) — garante a integridade da mensagem; protocolo de aplicação (vários) — são a interface com o usuário.

Query SQL: Pedido de acesso/restauração de informação dentro de um banco de dados.

Script (roteiro): pequeno programa formado por seqüência de comandos para automatizar tarefas rotineiras.

Site - (lugar): refere-se ao lugar onde se situa uma página.

Tags: qualquer comando HTML é chamado de *tag*, “marcação”.

URL: (*Universal Resource Locator*): Uma URL é o endereço único e próprio que cada página tem na *Web*. Este endereço é visto, geralmente, em algum lugar no topo do *browser*. Cada conexão tem uma URL por trás dele, que é quem vai definir, para o *browser*, a que fonte ele deve ir, no caso de ativada aquela conexão. Mesmo que a conexão não seja ativada, basta que posicione-se o cursor em cima da conexão para que o *browser* mostre na barra de status, a URL à qual aquela conexão está ligada. A URL ainda pode ser em forma de *e-mail*.

WWW (*World Wib Web*): sigla para teia mundial de informações. O mesmo que *Web*.

ANEXO 1 - ENTRADA DA CHAVE DE LOCALIZAÇÃO

```

<html>
<head>
  <title>Sistema - Cliente</title>
</head>

<style type="text/css">
td {font-size:12px; font-family:arial}
</style>

<body topmargin=0 leftmargin=0 marginheight=0 marginwidth=0 bgcolor=white>

<table border=0 cellpadding=0 cellspacing=0 width="100%" height="100%">
<tr>
<td align=center>
<table border=1 cellpadding=2 cellspacing=0 bordercolor=black width=400>
<tr>
<td bgcolor=#CCCCCC colspan=2 align=center><b>Digite CPF/CNPJ (somente os números)
para entrar no sistema</b></td>
</tr>
<tr>
<td height=10 bgcolor=#999999 colspan=2>&nbsp;</td>
</tr>
<form action="cliente.php3" method="post">
<tr>
<td align=right bgcolor=#CCCCCC>CPF/CNPJ:</td>
<td bgcolor=#CCCCCC><input type="text" name="cpf"></td>
</tr>
<tr>
<td bgcolor=#999999>&nbsp;</td>
<td bgcolor=#999999><input type="submit" value="Entrar"></td>
</tr>
</form>
</table>
</td>
</tr>
</table>

</body>

</html>

```

ANEXO 2 - RESULTADO DA PESQUISA POR CPF/CNPJ

```
<?php
include("../mssql.php3");
include("../funcoes.php3");
?>
<html>
<head>
    <title>Sistema - Informações do cliente</title>
</head>

<style type="text/css">
td { font-size:12px; font-family:arial }
a { color:#660000 }
.branco { color:white }
</style>

<body topmargin=0 leftmargin=0 marginheight=0 marginwidth=0 bgcolor=white>

<table border=0 cellpadding=0 cellspacing=0 width="100%" height="100%">
<tr>
<td align=center>
<table border=1 cellpadding=2 cellspacing=0 bordercolor=black width=600>
<?php
ClienteInfo($host, $user, $pass, $db, $cpf);
?>
</table>
</td>
</tr>
</table>

</body>

</html>
```

ANEXO 3 - DETALHAMENTO DA APÓLICE

```
<?php
include("../mssql.php3");
include("../funcoes.php3");
?>
<html>
<head>
    <title>Sistema - Informações da apólice <?php $apolice ?></title>
</head>

<style type="text/css">
td { font-size:12px; font-family:arial }
a { color:#660000 }
.branco { color:white }
</style>

<body topmargin=0 leftmargin=0 marginheight=0 marginwidth=0 bgcolor=white>

<table border=0 cellpadding=0 cellspacing=0 width="100%" height="100%">
<tr>
<td align=center>
<table border=1 cellpadding=2 cellspacing=0 bordercolor=black width=600>
<?php
ApoliceInfo($host, $user, $pass, $db, $apolice);
?>
</table>
</td>
</tr>
</table>

</body>

</html>
```

REFERÊNCIAS BIBLIOGRÁFICAS

- [BAR2000] BARRETO, Maurício Vivas de Souza. **Curso de Linguagem PHP**. Sergipe: Universidade Federal de Sergipe, 2000.
- [CON2000] CONECTIVA. **PHP, Web e PostgreSQL: intranet**. Endereço Eletrônico: <http://www.conectiva.com.br/> Data da consulta: 01/08/2000.
- [DAT1991] DATE, C. J. **Introdução a sistemas de banco de dados**. Rio de Janeiro: Campus, 1991.
- [DIA2000] DIAS, Valdir. **PHP**. Endereço eletrônico: <http://gopher.org> Data de consulta: 09/08/2000.
- [ELE1999] ELEMENTOS BÁSICOS DE SEGUROS. Mauá: Escola Técnica Tupy, 1999.
- [IIS2000] IIS. Endereço Eletrônico: <http://www.Microsoft.com.brasil/iis/> Data da consulta: 09/08/2000.
- [FIS2000] FISCHER, Herber. **PHP – Guia de Consulta rápida**. Rio de Janeiro: Novatec, 2000.
- [GUP1990] GUPTA, Umang, William GIETZ. **Guia do Programador em SQL**. Rio de Janeiro: Campus, 1997.
- [MIC2000] MICROSOFT PORTUGAL. **Soluções/Procedimentos adequados**. Endereço Eletrônico: <http://www.microsoft.com/portugal/Backstage/solutions> Data da consulta: 01/03/2000.
- [NAR1998] NARDINO, Luciane. **Protótipo de sistema de informação para uma corretora de seguros**. Blumenau, 1998. Monografia (Bacharelado em Ciências da Computação) Centro de Ciências Exatas e Naturais, Universidade Regional de Blumenau.
- [PHP2000] PHP. Endereço Eletrônico: http://www.php.net/*p Data da consulta: 09/08/2000.

- [SEG2000] SEGUROS. Endereço Eletrônico: <http://www.seguros.com.br> Data da consulta: 09/08/2000.
- [SOA2000] SOARES, Wallace Erica. **Programando em PHP**. Rio de Janeiro : Erica, 2000.
- [SQL2000] SQL SERVER. **O que é SQL Server**. Endereço Eletrônico: <http://www.sql.com.br> Data da consulta: 10/08/2000.
- [VIV2000] VIVAS, Maurício. **Aplicações Web utilizando PHP**. Endereço Eletrônico: <http://www.mauricio.vivas.com.br> Data da consulta: 03/09/2000.
- [WAP2000] WAPCLUB. **ASP, PHP E WAP**. Endereço Eletrônico: http://www.wapclub.com.br/wap_asp_php.asp Data da consulta: 10/08/2000.