

**RUBIO MARTIN SÁ FORTES**

**MODELO GERAL PARA ELABORAÇÃO DE APLICAÇÕES *WEB***

Trabalho de conclusão de curso apresentado ao Curso de Ciência da Computação.

UNIVERSIDADE PRESIDENTE ANTÔNIO CARLOS

Orientador: Prof. Elio Lovisi Filho  
Co-orientadora: Profa. Débora Paiva

**BARBACENA**  
**2003**

**RUBIO MARTIN SÁ FORTES**

**MODELO GERAL PARA ELABORAÇÃO DE APLICAÇÕES *WEB***

Este trabalho de conclusão de curso foi julgado adequado à obtenção do grau de Licenciado em Ciência da Computação e aprovado em sua forma final pelo Curso de Ciência da Computação da Universidade Presidente Antônio Carlos.

Barbacena – MG, dia de mês de ano.

---

Prof. Elio Lovisi Filho - Orientador do Trabalho

---

Prof. Frederico de Miranda Coelho - Membro da Banca Examinadora

---

Prof. Lorena Sophia C. de Oliveira - Membro da Banca Examinadora

## **AGRADECIMENTOS**

*Os mais sinceros agradecimentos aos professores orientadores, a Coordenação do Curso de Bacharelado de Turismo do Centro de Estudos Superiores Aprendiz e ao Bacharel em Turismo André Martin.*

## RESUMO

Um cliente para realizar uma viagem turística, isto é, procura sempre informações concisas e precisas sobre o destino a ser alcançado. Como nos dias atuais as informações estão dispostas de maneira dispersa, em vários sites de turismo, empresas aéreas e hotéis, o projeto visa reunir tais informações em um único modelo de website, facilitando assim a pesquisa do turista, utilizando-se das técnicas de Engenharia de Software voltada à aplicações *Web*.

**Palavras-chave:** *Web Engineering*, Turismo, Aplicações *Web*

## SUMÁRIO

<u>LISTAS.....</u>	<u>7</u>
<u>1 INTRODUÇÃO .....</u>	<u>9</u>
<u>2 ENGENHARIA DE SOFTWARE PARA WEB: VISÃO GERAL.....</u>	<u>12</u>
<u>3 DESENVOLVIMENTO DE UMA APLICAÇÃO PARA O DOMÍNIO DO TURISMO.....</u>	<u>32</u>
<u>4 MODELAGEM DA APLICAÇÃO DE TURISMO .....</u>	<u>40</u>
<u>5 CONCLUSÕES.....</u>	<u>51</u>
<u>REFERÊNCIAS BIBLIOGRÁFICAS.....</u>	<u>55</u>
<u>ANEXO I – INTERFACE PREVISTA.....</u>	<u>56</u>

## LISTAS

Figura 2.1 - Árvore de Requisitos de Qualidade.....	16
Figura 2.2 - Estrutura Linear.....	23
Figura 2.1 - Estrutura em Grade.....	23
Figura 2.2 - Estrutura Hierárquica.....	24
Figura 2.3 - Estrutura em Rede.....	24
Figura 3.1 – Website modificado por indivíduo não autorizado.....	34
Figura 3.2 – Website com informações inexistente.....	34
Figura 3.3 – Falha na pesquisa de destinos pelo website.....	35
Figura 4.1 – Caso de Uso Usuário/Agência.....	41

Figura 4.2 – Caso de Uso Informante/ <i>Web</i> Master.....	42
Figura 4.3 – Diagrama de Classes.....	47
Figura 4.4 – Diagrama de Atividade.....	48
Figura 4.5 – Diagrama de Estados.....	49

# 1 INTRODUÇÃO

## 1.1 JUSTIFICATIVA

Com o grande desenvolvimento da área de turismo no mundo, a necessidade de informações sobre vias de acesso, meios de transporte, vagas em hotéis, e outras informações referentes ao mesmo, se tornaram extremamente necessárias.

Num mundo globalizado onde a concorrência entre empresas turísticas é muito forte, a rapidez com que a informação é prestada ao cliente é fundamental. Com isso surgiu a idéia de se desenvolver uma rede integrada entre empresas turísticas e hotéis, de forma que as informações possam fluir com maior rapidez.

Quando um cliente chega à uma agência turística e está interessado em adquirir um pacote turístico, é ideal que sejam passadas todas as informações quanto à vagas, condições climáticas atuais e previstas do local imediatamente à requisição, para que o cliente opte por adquiri-lo o quanto antes.

Existem hoje em dia, *websites* especializados no assunto, mas no entanto, segundo a classe de profissionais de turismo, esses *websites* são confusos e inconsistentes.

Um dos grandes problemas desses sites é o fato de que aplicações desse tipo são desenvolvidas de maneira ad hoc, isto é, são implementados sem as metodologias de engenharia de software, e ainda, a engenharia de software tradicional não suporta muitas das vezes, as necessidades de um meta-modelo para aplicações voltadas para web.

## **1.2 OBJETIVO**

O Projeto visa a modelagem de um website onde essas informações estariam dispostas de maneira organizada e consistente. Essa modelagem servirá de meta-modelo para futuras implementações poupando assim tempo em desenvolvimento. Como será um modelo geral para websites de turismo, a análise de requisitos será definida através de uma pesquisa feita junto ao Centro de Estudos Superiores Aprendiz LTDA, onde é ministrado o curso de Bacharelado em Turismo, visando um apanhado geral sobre as informações necessárias para o website.

## **1.3 VISÃO GERAL**

Esta monografia apresenta em seu segundo capítulo uma visão geral sobre a Engenharia de Software para Web, analisando os atributos de qualidade de uma aplicação voltada para Web como Segurança e Padrões da Internet, formulação e análise de um Sistema Voltado para Web, Modelos de aplicações voltadas para Web, e ainda, os testes a serem feitos e a gerência do projeto.

No terceiro capítulo, a monografia apresenta as etapas do desenvolvimento de uma aplicação de turismo voltada para Web, detalhando o seu plano de desenvolvimento, a especificação de requisitos e requisitos funcionais, os requisitos de qualidade que englobam a confiabilidade da aplicação, eficiência e a portabilidade.

No quarto capítulo, são apresentados os modelos gerais da aplicação de turismo voltada para Web. São apresentados os diagramas de caso de uso, a descrição das classes e seus respectivos eventos, o diagrama de classes, o diagrama de atividades e finalmente o diagrama de estados.

No capítulo das conclusões, são apresentadas a revisão da monografia, as conclusões da monografia e as recomendações para futuros trabalhos.



## 2 ENGENHARIA DE SOFTWARE PARA *WEB*: VISÃO GERAL

Segundo Pressman [PRE2002], a Engenharia de Software voltada para *Web* é uma ciência ainda em evolução. O autor aborda aspectos gerais da Engenharia de Software tradicional a Engenharia de Software voltada para *Web* apresentando os tópicos relacionados de forma a facilitar o desenvolvimento de aplicações voltadas para *Web*.

Gennaro Costagliola, Filomena Ferruci e Rita Francese [GEN2000], apresentam em seu artigo “*Web engineering: Models and Methodologies for the Design os Hypermedia Applications*” os modelos e metodologias para aplicações hipermídia na engenharia de *Web*.

Nora Koch e Andréas Kraus [NOR2002], demonstram como a linguagem UML é aplicada no desenvolvimento de aplicações voltadas para *Web* em seu artigo “*The Expressive Power of UML – based Web Engineering*”.

Grady Booch, James Rumbaugh e Ivar Jacobson, [GRA2000], criadores da linguagem UML, detalham a linguagem no livro “*UML – Guia do Usuário*” apresentando todos os tipos existentes de diagramas e facilitadores para o desenvolvimento da aplicação voltada para *Web*.

Como as aplicações *Web* se tornaram integradas às estratégias de mercado de grandes e pequenas empresas, a necessidade de se construir um modelo confiável, viável e adaptável, a Engenharia de Software para *Web* vem se tornando cada vez mais importante. A Engenharia de Software para *Web*, não é uma cópia perfeita da Engenharia de Software, mas utiliza boa parte de seus conceitos fundamentais. Como os sistemas e as aplicações *Web* tendem a se tornar cada vez mais complexas, a utilização de técnicas usuais da Engenharia de Software, criariam um modelo não-suficiente para garantir um modelo aceitável, podendo

assim uma falha, comprometer toda a estrutura da aplicação. Por isso a necessidade de utilizar uma Engenharia de Software específica para *Web*, onde novas técnicas e métodos para modelagem da aplicação garantem um bom planejamento e gerenciamento do projeto de software [PRE2002].

As aplicações voltadas para Web necessitam de cuidados especiais em sua fase de modelagem. As técnicas e métodos propostos pela Engenharia de Software tradicional, apesar de serem inteiramente funcionais para as aplicações não voltadas para Web, não determinam todos os aspectos para que a aplicação voltada para Web seja bem projetada. Novos aspectos surgem quando tratamos de aplicações voltadas para Web. Esses aspectos são descritos no decorrer deste capítulo.

## **2.1 ATRIBUTOS DE SOFTWARE VOLTADO PARA WEB**

Há características em que um Software voltado para *Web* se diferencia da maioria dos softwares convencionais. Essas características são encontradas na grande maioria das aplicações *Web* [PRE2002].

**Intensidade de Tráfego na Rede:** Pela própria natureza, uma aplicação *Web* deve ser implementada em uma rede de computadores. Com isso, aspectos relacionados à velocidade em que os dados trafegam por ela devem ser analisados, de forma a não causar congestionamentos;

**Direcionador de Conteúdo:** Na maioria dos casos, a aplicação *Web* tem como finalidade principal, apresentar ao usuário final conteúdo hipermídia, como textos, sons, vídeos e figuras; e

**Evolução Contínua:** Em softwares convencionais, as atualizações são planejadas de forma sistemática e programada. Já em aplicações *Web* podem sofrer várias modificações ao longo de sua existência, chegando e ocorrer em determinados casos, atualizações em tempo real.

As aplicações voltadas para web se diferenciam das aplicações convencionais no aspecto de funcionarem em uma rede de computadores menos veloz que as redes locais onde as aplicações contam com velocidade suficiente para que não haja tráfego prejudicial ao seu desempenho. Na rede mundial de computadores, a intensidade de tráfego de informações e a limitação física imposta pelos meios de transmissão prejudicam as aplicações voltadas para Web.

Outro aspecto fundamental, é que as aplicações convencionais são muito diferentes das aplicações voltadas para Web no sentido de não necessitarem de um caminho específico para a apresentação das informações. Nas aplicações voltadas para Web, geralmente há a necessidade de se conduzir o usuário numa pesquisa das informações com um direcionador de conteúdo.

As aplicações voltadas para Web possuem atualizações freqüentes e necessárias. Nenhum software convencional tem essa característica. Na aplicação voltada para o turismo por exemplo, as tarifas de transporte e hospedagem podem variar continuamente de acordo com variações de câmbio e as condições climáticas são alteradas diariamente.

Na criação de uma estrutura para um Software voltado para *Web*, a mesma deve conter aspectos que facilitem a administração das atualizações que possa vir a ocorrer [PRE2002].

Os cuidados contínuos com a aplicação e com a entrada de informações, permitem que a aplicação se torne estável, mas alguns aspectos devem ser considerados [PRE2002]:

**Imediatismo:** Aplicações voltadas para *Web*, possuem um imediatismo que, geralmente não é encontrado em nenhum outro tipo de software. As atualizações são constantemente necessárias, pois o conteúdo a ser apresentado pela aplicação varia de acordo com informações atualizadas. Assim, o desenvolvedor deve utilizar métodos de planejamento, análise, design, implementação e testes que foram adaptados para softwares voltados para *Web*.

**Segurança:** Como aplicações voltadas para *Web* estão em rede, devemos limitar o número de usuário que têm permissão para acessar as informações. Nesse caso, o desenvolvimento deve ser feito com base em uma estrutura que proveja segurança e integridade dos dados, permitindo que apenas usuários que tenham privilégio acessem informações pertinentes ao seu grupo de usuário.

**Estética:** Uma boa interface garante conforto ao usuário, provendo maior compreensão e até em alguns casos, como em *e-commerce*, induzir o usuário a se sentir confortável ao adquirir um determinado produto. Os dados são apresentados de forma organizada e objetiva evitando que o usuário se perca dentro da aplicação. Muitas cores em tonalidades fortes e que se contrastam com outros tons não são aconselháveis.

As aplicações voltadas para *Web* são acessadas por vários usuários ao mesmo tempo. Com isso, o tráfego de informações aumenta e isso gera sobrecarga na aplicação. O Desenvolvedor deve se preocupar em desenvolver uma aplicação que gere resultados rápidos, pois o usuário geralmente não se sente confortável em esperar.

Como foi dito, muitas pessoas acessam as informações de uma aplicação voltada para *Web*. As permissões de acesso a informações devem ser definidas para que usuários não-privilegiados acessem determinadas funções e dados da aplicação.

Um aspecto importante se tratando de aplicações voltadas para *Web* é a estética da aplicação. O usuário acessa a aplicação e espera que a mesma lhe apresente os dados de forma agradável e confortável. Uma aplicação sem estética tende a perder o acesso de seus usuários no caso de outras aplicações semelhantes apresentarem uma estética melhor.

Essas características gerais se aplicam a todas as aplicações *Web*, mas em níveis diferentes de influência. As categorias a seguir são mais comumente encontradas no trabalho de Engenharia de Software para *Web* [PRE2002].

*Informações:* Conteúdo somente para leitura, contendo navegação simples e *links* simples.

*Download:* O usuário efetua *download* de determinado arquivo de um dado servidor.

*Adaptável:* O usuário configura o conteúdo de acordo com suas próprias necessidades.

*Interação:* O usuário tem a capacidade de se comunicar com uma comunidade de usuários através de chats, mensagens instantâneas ou quadro de avisos.

*User input:* forma mais básica de comunicação entre o usuário e a aplicação.

*Orientado a Transação:* O usuário faz uma requisição e a aplicação retorna um determinado resultado;

*Orientado a Serviço:* A aplicação provê um determinado serviço ao usuário, por exemplo, um cálculo de juros sobre prestações a serem pagas;

*Portal:* A aplicação disponibiliza canais para outros conteúdos na *Web*, fora do domínio da aplicação;

*Acesso a Banco de Dados:* O usuário pode requerer consultas em um determinado Banco de Dados disponibilizado pela aplicação voltada para *Web*;

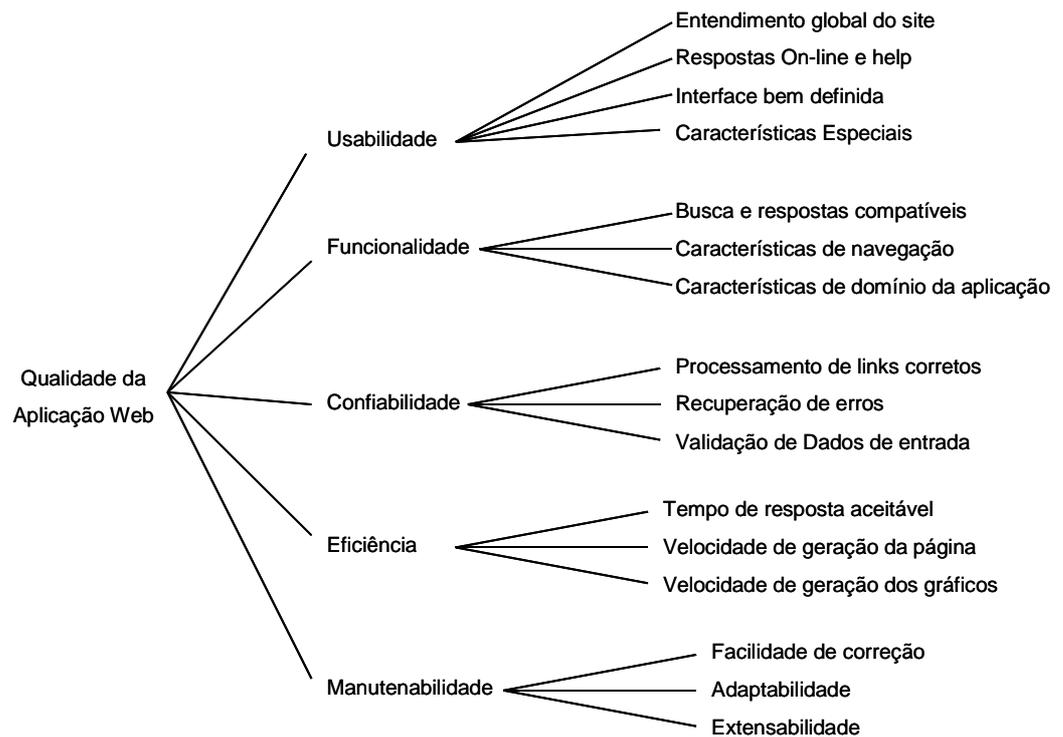
*Data Warehousing:* O usuário requisita um conjunto de dados em vários Banco de Dados.

Na aplicação desenvolvida para o domínio do turismo, algumas dessas características não são aplicáveis. Nela, são utilizados além da informações, a interação do usuário com a aplicação, a orientação a transação na requisição de origens e destino, a orientação a serviço no cálculo dos valores a serem apresentados e o acesso a banco de dados na busca das informações das cidades de origem e destino.

### 2.1.1 ATRIBUTOS DE QUALIDADE

O número de utilizadores de aplicações *Web* atualmente é incalculável. As previsões para o aumento desse número de usuários, são cada vez mais otimistas. Com esse enorme número de pessoas utilizando serviços de aplicações *Web*, fica complicado de satisfazer ao gosto de todos os usuários. Há aqueles que preferem aplicações que disponibilizam conteúdos completos ao invés de exibir informações específicas, alguns preferem aplicações com animações flash e outros somente texto, e as variações não terminam por aí [PRE2002].

A figura 2.1, denominada “árvore de requisitos de qualidade” por Olsina [OLS99] mostra de maneira sucinta, o que uma aplicação *Web* se tornar uma aplicação *Web* de alta-qualidade.



**Figura 2.4 - Árvore de Requisitos de Qualidade [OLS99]**

Esses requisitos de qualidade são indispensáveis no desenvolvimento de aplicações voltadas para Web. No entanto, a modelagem apresentada por essa monografia não

abrange esses aspectos pois os mesmos devem ser analisados em sua fase de desenvolvimento. A modelagem não possui por exemplo a capacidade de se medir a eficiência de uma aplicação voltada para Web, pois a mesma é medida através de testes cujos parâmetros são analisados em sua fase de funcionamento.

### 2.1.2 SEGURANÇA

Pelo fato da aplicação voltada para *Web* estar disponibilizada em uma rede, a princípio o acesso às informações estão abertas para acesso não-autorizado. Em alguns casos, o acesso não-autorizado a informações por pessoas com intenções maliciosas pode causar danos ao conteúdo disponibilizado.

Existem hoje, várias técnicas de proteção disponíveis como criptografia de dados, firewalls e outros métodos. A utilização desses métodos, diminui a possibilidade do acesso malicioso e do acesso a conteúdos não disponibilizados a usuários com menos privilégios [PRE2002].

Como foi visto anteriormente na seção 2.1 dessa monografia, a necessidade em se definir níveis de autorização para o acesso das informações é indispensável numa aplicação voltada para Web. Porém, é necessária uma abordagem ao aspecto da segurança em nível de acesso aos dados no servidor em que a aplicação está instalada. Vários métodos de segurança de dados se aplicam a diferentes níveis de proteção. A criptografia visa a camuflagem das informações e garante proteção em seu transporte. Os *firewalls* por sua vez, garantem a integridade dos servidores da aplicação contra as invasões de usuários não-autorizados pelo sistema operacional.

### 2.1.3 PADRÕES DA INTERNET

Durante muito tempo, a linguagem padrão da Internet foi o HTML (*Hypertext Markup Language*) que trazia consigo a possibilidade da inclusão de texto, figuras, animações e sons. Um novo padrão está chegando ao desenvolvimento de aplicações voltadas para *Web*. O XML (*Extensible Markup Language*), com a utilização do XML, o significado das tags

comuns são definidas na informação transmitida ao site cliente, ou seja, o conteúdo é atualizado dinamicamente [PRE2002].

A flexibilidade dessas linguagens proporciona o desenvolvimento de uma aplicação voltada para Web com vantagens em relações às demais linguagens. A grande maioria dos servidores de aplicações e seus respectivos clientes aceitam as aplicações desenvolvidas em HTML e XML de maneira nativa, sem que haja a necessidade de instalação de *plugins* adicionais, como é o caso da linguagem PHP.

## **2.2 UM ARCABOUÇO PARA ENGENHARIA DE SOFTWARE PARA WEB**

A necessidade de se utilizar um gerenciamento sólido sobre os princípios de Engenharia tornou-se muito importante. Torna-se então necessário o desenvolvimento de um *Web* framework que envolva modelo de processos efetivo, constituído por atividades e tarefas de engenharia [PRE2002].

A Engenharia de Software para *Web* começa com o processo de *Formulação* – atividade que identifica os pontos-chaves e os objetivos da aplicação e estabelece o escopo para o primeiro incremento. O *Planejamento* estima os custos do projeto, avalia riscos associados ao desenvolvimento e define um cronograma para o desenvolvimento da aplicação. A *Análise* estabelece os requisitos técnicos para a aplicação e identifica o conteúdo que será incorporado, bem como a estética da aplicação [PRE2002].

A atividade da engenharia, incorpora duas tarefas paralelas. O *Projeto de Conteúdo* e *Produção*, que são tarefas a serem executadas pelo pessoal não-técnico da equipe de desenvolvimento. Nessa parte, são definidos os elementos de texto, gráficos, áudio e vídeo que serão integrados a aplicação *Web* [PRE2002].

A *Produção de Página* é uma atividade de construção que utiliza muito as ferramentas para desenvolvimento de aplicações *Web*. O Conteúdo adquirido na fase de engenharia é então mesclado com o projeto de arquitetura, navegação, e interface para

produção de páginas em HTML, XML, e outras linguagens orientadas a processo, como o Java. Na fase de *Teste*, são verificados os scripts, links e outros aspectos navegacionais da aplicação *Web* [PRE2002].

Finalmente o projeto é levado a *Avaliação do Cliente*, para verificar se os requisitos foram bem utilizados e a aplicação realmente solucionou o dado problema [PRE2002].

Na aplicação para o domínio do turismo o Projeto de conteúdo e Produção é totalmente desenvolvido por profissionais das áreas envolvidas com turismo. A prestação de informações sobre as cidades é feita pelas prefeituras, as condições climáticas pelos órgãos responsáveis, os meios de transporte pelas empresas de transporte e a hospedagem pelos hotéis, pousadas e áreas de camping.

## **2.3 FORMULAÇÃO E ANÁLISE DE UM SISTEMA VOLTADO PARA WEB**

Formulação e análise de um sistema voltado para *Web*, consiste em aplicar uma série de atividades de engenharia de software que começa com a identificação dos pontos-chaves gerais e termina com um modelo de análise ou uma especificação de requisitos do sistema [PRE2002].

A Formulação permite ao cliente e ao desenvolvedor estabelecerem os pontos cruciais do sistema e seus objetivos para a construção da aplicação voltada para *Web*. A Análise é a atividade que identifica os dados, a funcionalidade, e os requerimentos comportamentais para a aplicação voltada para *Web* [PRE2002].

### **2.3.1 FORMULAÇÃO**

Um conjunto de questionamentos devem ser elaborados para que se possa começar a formulação. São elas [PRE2002]:

- Qual a principal motivação para o desenvolvimento da aplicação voltada para *Web*?
- Porque a aplicação é necessária?
- Quem irá utilizar a aplicação?

A resposta a cada uma dessas simples questão deve ser dada de forma mais sucinta possível. Assim que todos os pontos-chaves das informações e da aplicação estiverem definidos, um perfil de usuário é criado. Esse perfil determina quais as melhores configurações da aplicação para melhor satisfazer ao usuário final [PRE2002].

A formulação se faz necessária pelo fato de existirem vários tipos de aplicações voltadas para *Web*. Hoje em dia, quase todos os serviços existentes em forma presencial podem ser adaptados para uma aplicação voltada para *Web*.

### 2.3.2 ANÁLISE

Quatro tipos de análises devem ser considerados para a criação de um escopo completo para uma aplicação voltada para *Web* [PRE2002].

**Análise de conteúdo:** Nessa análise todo o conteúdo da aplicação voltada para *Web* é analisado. Incluindo textos, imagens, vídeos e demais recursos a serem utilizados.

**Análise de interação:** É analisada a maneira que o usuário se interagirá com a aplicação voltada para *Web*.

**Análise Funcional:** O uso de cenários criados na fase de análise de interação definem as ações da aplicação voltada para *Web*. Todas as operações e funções são descritas em detalhes.

**Análise de Configuração:** O ambiente e a infra-estrutura em que a aplicação voltada para *Web* estará hospedada são analisados. Ela poderá estar sendo

utilizada numa Intranet ou Extranet. Nessa fase também é definido qual banco de dados será utilizado para a aplicação voltada para *Web*.

Embora uma especificação de requisitos detalhada de uma aplicação voltada para *Web* seja recomendada, esses documentos são raros pois a evolução contínua da aplicação torna-os obsoleto [PRE2002].

As análises descritas acima são de suma importância para o desenvolvimento de uma aplicação voltada para *Web*. Com essas análises o desenvolvimento da aplicação se torna mais consistente, evitando o desperdício de tempo e evitando também que informações desnecessárias sejam incluídas no projeto.

## **2.4 MODELOS PARA APLICAÇÕES VOLTADAS PARA WEB**

Para elaborar um modelo baseado em *Web* efetivamente, o desenvolvedor deve utilizar com quatro elementos técnicos.

**Princípios e métodos de modelagem:** Modularidade com alta coesão de informações, ocultação de informações, elaboração gradativa, e outras heurísticas da Engenharia de Software que tornam a aplicação voltada para *Web* fácil de ser adaptada, evoluída, testada e usada. O UML por exemplo pode ser adaptado para a Engenharia de Software voltada para *Web* [NAN98].

**Golden Rules:** Aplicações voltadas para *Web* vêm sendo desenvolvidas há anos. Com isso os desenvolvedores criaram um conjunto de regras que podem ser reutilizadas para elaboração da aplicação voltada para *Web*.

**Design Patterns:** o Design Pattern é um modelo geral que é desenvolvido para solucionar um problema genérico e que pode ser utilizado para implementação de outras aplicações voltadas para *Web*. Podem ser utilizados não somente nos elementos funcionais como também na documentação, gráficos e na própria modelagem da aplicação.

**Templates:** Um modelo pode ser utilizado para se construir o esqueleto da moldura do design pattern. O template implicitamente separa o conteúdo hipermídia da especificação estrutura do modelo.

A utilização desses elementos proporciona ao desenvolvedor uma série de facilidades em sua fase de implementação. O uso de design patterns e templates por exemplo, economiza muito tempo na implementação de uma aplicação voltada para Web por se tratarem de modelos já existentes que na maioria das vezes as alterações necessárias para adaptá-los as mais diversas necessidades são mínimas.

## 2.4.1 PROJETO DA ARQUITETURA

O Modelo de arquitetura para sistemas baseados em *Web*, define toda a estrutura hipermídia da aplicação, aplica o design pattern e os templates que irão compor a aplicação [PRE2002].

### 2.4.1.1 Estruturas de Aplicações voltadas para Web

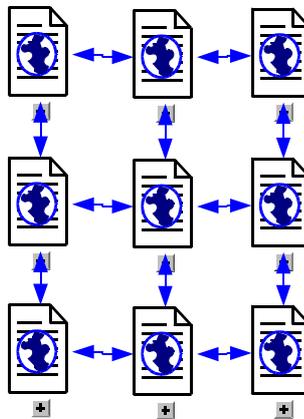
Em geral as arquiteturas de engenharia estão ligadas diretamente aos pontos-chaves da aplicação voltada para *Web*, ao conteúdo que será apresentado, aos usuários que irão utilizar a aplicação e à filosofia de navegação estabelecida. O desenvolvedor pode optar por quatro tipos de estruturas apresentadas a seguir [POW98]:

*Estrutura Linear:* É geralmente utilizada para representar estruturas de seqüências de interações e pode haver algumas variações. Um bom exemplo para esse tipo de estrutura, seria uma aplicação que serve como tutorial para um determinado fim. Um outro exemplo seria o caso em que o usuário tenha que entrar com os dados em forma seqüencial. Essa estrutura é exemplificada pela figura 2.2.



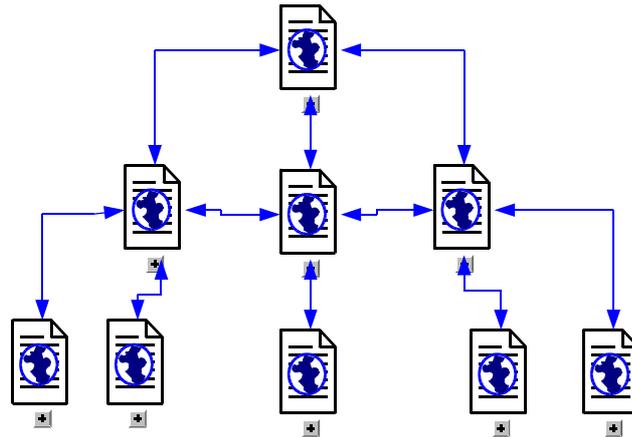
**Figura 2.5 - Estrutura Linear [POW98]**

*Estrutura em Grade:* Utilizada em casos em que a aplicação voltada para *Web* possa ser representada em duas ou mais dimensões. Imagine o caso de e-commerce, onde são apresentados diversos produtos semelhantes de várias empresas. Os tipos de produtos estariam dispostos em linha horizontal e os diversos fabricantes em linha vertical. Essa estrutura é representada pela figura 2.3.



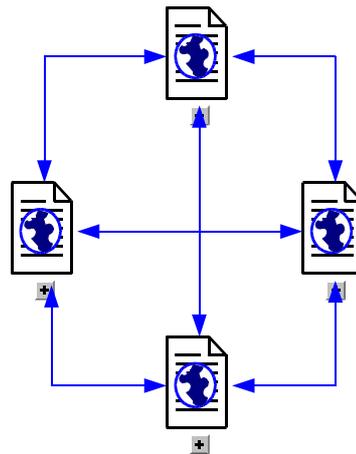
**Figura 2.6 - Estrutura em Grade [POW98]**

*Estrutura Hierárquica:* É sem dúvida a mais utilizada por aplicações voltadas para *Web*. À partir de uma página principal, o usuário escolhe um determinado trajeto a ser seguido e com isso, páginas abaixo da hierarquia vão sendo apresentadas [PRE2002]. Vale lembrar que, o usuário pode mudar de uma página para outra página da mesma hierarquia. A estrutura é apresentada pela figura 2.4.



**Figura 2.7 - Estrutura Hierárquica [POW98]**

*Arquitetura em Rede:* Nessa arquitetura, as páginas estão dispostas de maneira que o usuário tem livre acesso a todas as páginas a partir de qualquer uma das páginas. Isso garante flexibilidade ao usuário. A estrutura está representada pela figura 2.5.



**Figura 2.5 - Estrutura em Rede [POW98]**

As estruturas apresentadas acima podem ser combinadas para melhor representar um determinado tipo de aplicação voltada para *Web* para compor novas estruturas. A maioria da estrutura pode ser hierárquica e em algumas partes se apresentar de forma linear [PRE2002].

Cada aplicação voltada para *Web* possui suas características específicas. Em cada uma delas é necessária a utilização de uma estrutura em especial.

No modelo proposto pela monografia, a estrutura utilizada será a estrutura linear, por se tratar de uma aplicação que não necessita de navegação. O desenvolvedor pode projetar a aplicação de forma que todas as informações estejam dispostas em apenas uma página.

#### 2.4.1.2 Design Patterns (Modelos-padrão)

Um Design Pattern é uma abordagem genérica que pode ser utilizada para solucionar vários problemas específicos.

Quando o processamento de dados é requerido dentro de uma aplicação voltada para *Web*, os modelos-padrão propostos por [BUS96], [GAM95] e outros são aplicáveis. Os modelos-padrão de navegação propostos por [BER98] permitem ao usuário se orientar dentro da aplicação. Os modelos-padrão propostos são:

- *Cycle* – um padrão que permite ao usuário retornar à página visitada anteriormente.
- *Web Ring* – implementa uma grande rede em anel que liga o hipertexto em um determinado tour.
- *Contour* – ocorre quando um anel choca-se com outro permitindo a navegação entre os demais anéis.
- *Counterpoint* – padrão que adiciona comentários ao hipertexto, o qual interrompe a narrativa para dar maiores explicações sobre o assunto.
- *Mirrorword* – o conteúdo é apresentado usando diferentes meios, cada um com pontos de vista diferente.
- *Sieve* – padrão que induz o usuário a fazer determinadas opções com intuito de direcioná-lo a uma determinada informação.
- *Neighborhood* – padrão que usa uma navegação uniforme entre as páginas para prover uma navegação consistente ao usuário.

Esses modelos-padrão podem ser modificados de acordo com a necessidade de cada aplicação a ser desenvolvida [PRE2002].

## **2.4.2 PROJETO DE NAVEGAÇÃO**

Assim que a estrutura da aplicação voltada para *Web* e seus componentes estiverem definidos, o desenvolvedor deve definir caminhos para o usuário poder navegar de forma eficiente na aplicação para acessar seus conteúdos e serviços. Para completar, deve também identificar a semântica de navegação para os vários tipos de usuários e definir os mecanismos de execução da aplicação [PRE2002].

Uma aplicação voltada para *Web* de grande porte pode vir a ter vários tipos de usuários, como usuários comuns, usuários registrados, usuários com privilégios, entre outros. Com isso, cada conjunto de usuário terá permissão de acesso somente àquelas determinadas pelo desenvolvedor [PRE2002].

Com a definição dessas permissões, a modelagem de navegação deve encaminhar o usuário pelos dados que ele tem acesso, evitando assim que ele se perca em áreas restritas [PRE2002].

Prosseguindo na modelagem de navegação, o desenvolvedor deve definir os mecanismos de cada link de navegação e qual o melhor componente a ser usado para exibir esse link, tais como, textos, botões entre outros [PRE2002].

Na modelagem proposta por essa monografia, o projeto de navegação torna-se dispensável, pois a aplicação pode apresentar todas as informações em uma única página. É claro que existem diversas maneiras de se implementar uma aplicação a partir do modelo apresentado, mas o objetivo principal da aplicação para o domínio de turismo é simplificar ao máximo as consultas referentes a viagens a serem realizadas.

### 2.4.3 PROJETO DA INTERFACE

As aplicações voltadas para *Web* têm algumas características especiais que não são encontradas em aplicações comuns e devem ser consideradas [PRE2002].

A interface de uma aplicação votada para *Web* é sua primeira impressão. Ela revela o valor de seu conteúdo e a sofisticação de seus serviços e capacidade de processamento. Nielsen e Wagner [NIE96] sugerem algumas regras básicas para implementação de interface em aplicações voltadas para *Web*.

- Erros de servidores geralmente levam os usuários a saírem do site para procurarem informações em outros sites.
- A leitura na tela do computador, é geralmente vinte e cinco por cento mais lenta que a leitura normal. Não se deve colocar textos muito extensos para evitar o cansaço do usuário.
- Evitar as mensagens “página em construção” pois as mesmas causam expectativas e desapontamento ao usuário.
- Usuários preferem não rolar a tela. As informações principais devem estar dispostas de maneira a facilitar a leitura em resoluções mais comuns.
- Os menus de navegação e as barras de links no topo da página devem ser desenvolvidas consistentemente, e devem estar disponíveis em todas as páginas que o usuário navegar.
- A estética nunca pode prejudicar a funcionalidade.
- As opções de navegação devem ser óbvias. O usuário não deve ter que procurar na tela a maneira que ele vai acessar outros conteúdos ou serviços.

Uma interface bem projetada causa ao usuário uma boa impressão do conteúdo que está sendo visitado. Ela não precisa ser agressiva e sim bem estruturada e ergonomicamente confortável aos olhos do usuário [PRE2002].

O projeto de interface define a aparência da aplicação voltada para web. As cores utilizadas, a disposição dos dados na página e a indução ao correto uso da aplicação são características importantes definidas nesta fase. Um exemplo de interface onde a disposição das informações e funções da aplicação para o domínio do turismo é encontrado no Anexo I deste trabalho. Vale lembrar que o protótipo da interface apresentada, visa apenas a disposição dos dados a serem informados de forma consistente e objetiva e não a utilização de cores e demais aspectos gráficos..

## **2.5 TESTANDO A APLICAÇÃO VOLTADA PARA WEB**

A fase de teste consiste em fazer exaustivos exercícios sobre o software com intuito de achar possíveis falhas no decorrer de sua utilização. Como uma aplicação *Web* reside em uma rede de computadores, ela será utilizada por diversos sistemas operacionais, browsers, plataformas de hardware e protocolos de comunicação, e isso faz com que a busca por erros se torne uma tarefa exaustiva para os desenvolvedores [PRE2002].

A seguir são apresentados passos para realização de testes na aplicação voltada para *Web* [PRE2002].

- O modelo de conteúdo é revisado para identificação de erros.
- O modelo de Design é revisado para encontrar erros de navegação.
- Os componentes de processos selecionados são testados individualmente.
- A arquitetura é construída e os testes de integração são realizados.
- Os testes de funcionalidade e de entrega de conteúdo são efetuados.

- A aplicação é instalada em vários ambientes de utilização e testados em todos esses ambientes.
- Finalmente, a aplicação é monitorada por usuários anteriormente contactados, os beta-testers.

Como aplicações voltadas para *Web* evoluem continuamente, as atividades de teste deve ser contínua, conduzida pelo grupo de suporte da aplicação [PRE2002].

Na aplicação desenvolvida para o domínio do turismo, as informações deverão ser sempre atualizadas e conferidas pelos responsáveis pela prestação dessas informações e pelos desenvolvedores da aplicação. Devido ao grande número de modificações realizadas, como por exemplo a variação de câmbio e condições climáticas, o grupo de suporte da aplicação deve sempre manter contato com seus informantes para que dados incorretos sejam apresentados aos seus usuários.

## **2.6 GERÊNCIA DO PROJETO**

Engenharia para *Web* é um processo complicado e necessita de vários profissionais trabalhando em paralelo para produzir uma aplicação voltada para *Web* que possua qualidade.

Alguns profissionais são imprescindíveis para o bom desenvolvimento da aplicação *Web* [PRE2002].

- *Desenvolvedor de Conteúdo* – Reúne as informações relacionadas ao conteúdo da aplicação. Devem trabalhar com direitos de exibição da informação, coletar dados e tratar a parte relacionada ao marketing.
- *Web Publisher* – Deve organizar o conteúdo coletado para a inclusão na aplicação.

- *Engenheiro de Web* – Como temos dito, o engenheiro de *Web* deve fazer a análise, modelagem de arquitetura, modelagem de interface e modelagem navegacional.
- *Especialista em Suporte* – Como aplicações voltadas para *Web* evoluem continuamente, o especialista em suporte deve estar sempre dando manutenção à aplicação.
- *Administrador* – Também chamado *Web master*, é a pessoa responsável pela administração dia-a-dia da aplicação, como tráfego na aplicação.

O papel de cada membro da equipe de desenvolvimento deve sempre ser bem definido para que o processo de engenharia seja bem realizado. Cada profissional, desempenhando bem suas funções, garante um projeto final bem elaborado.

## **3 DESENVOLVIMENTO DE UMA APLICAÇÃO PARA O DOMÍNIO DO TURISMO**

### **3.1 PLANO DE DESENVOLVIMENTO**

#### **3.1.1 OBJETIVOS**

O objetivo principal do projeto é criar um modelo genérico que possa ser utilizado de forma a possibilitar ao usuário a consulta específica a informações relacionadas às cidades de origem e destino desejadas com todas as informações relacionadas à meio de transporte, hospedagem e atrações turísticas dessas cidades de maneira organizada.

#### **3.1.2 FUNÇÕES PRINCIPAIS**

A aplicação fornecerá dados relativos a viagens turísticas de forma que o cliente possa encontrar todos os dados necessários para efetuar uma viagem com segurança e conforto.

Serão informadas aos usuários, todas as informações sobre destino pretendido, meio de transporte e seus respectivos horários e valores e, quais as opções de hospedagem.

Essas informações, deverão ser concisas e consistentes, evitando desperdício de tempo e informações concorrentes.

### **3.1.3 QUESTÕES DE DESEMPENHO**

Por se tratar de uma aplicação *Web*, as informações a serem apresentadas, deverão ser apresentadas de forma rápida e objetiva, evitando uma quantidade excessiva de sons e vídeos na aplicação.

Caso seja considerada a hipótese da utilização de vídeos, o mesmo deverá ser exibido através de uma página à parte da aplicação principal, evitando assim de carregar a aplicação e estressar o usuário.

### **3.1.4 RESTRIÇÕES TÉCNICAS ADMINISTRATIVAS**

Como a aplicação *Web* tratará de informações reunidas de vários informantes como hotéis, empresas de transporte, e de órgãos responsáveis por meteorologia, essas informações serão apresentadas de forma totalmente atualizadas caso haja o correto fornecimento das mesmas. Principalmente no que se refere a vagas disponíveis em hotéis e pousadas e disponibilidade de horários e passagens nas empresas de transporte.

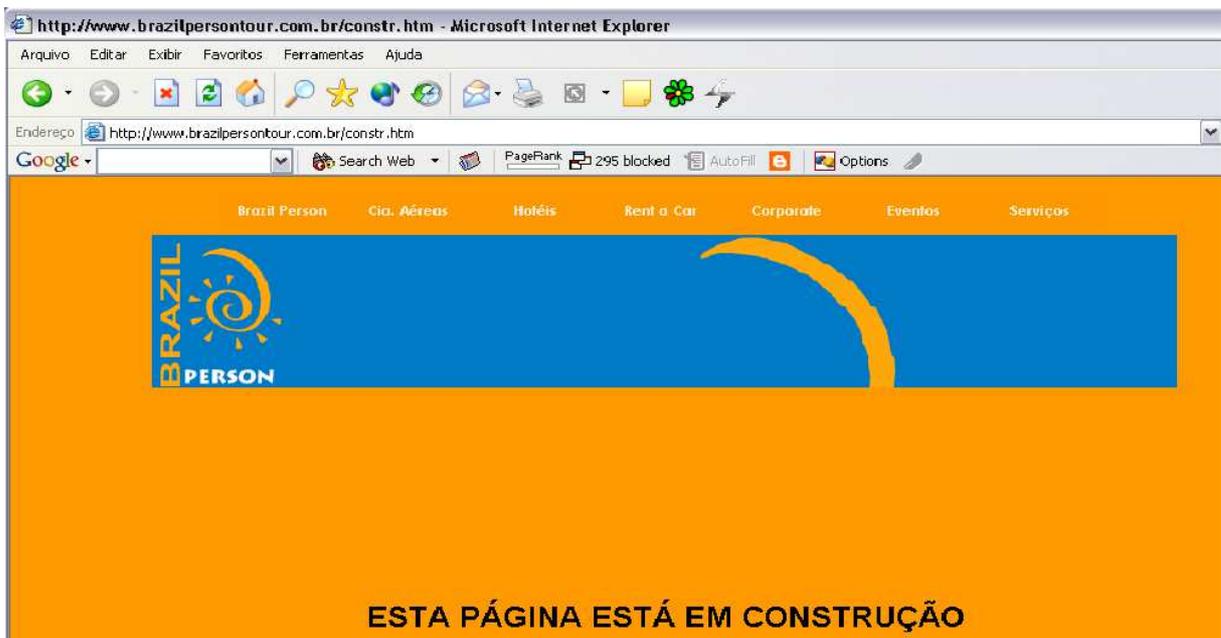
### **3.1.5 AVALIAÇÃO DOS MODELOS EXISTENTES**

Os websites existentes rede mundial de computadores não apresentam em sua maioria os atributos citados no capítulo anterior. Um exemplo claro dessa afirmação, foi uma análise feita no site <http://www.ipanorama.com>, onde as informações referentes à cidade de Barbacena-MG por exemplo, foram alteradas intencionalmente por um indivíduo o que mostra claramente a falta de segurança discutida no item 2.1.2, e mostrado na figura 3.1.



**Figura 3.1 – Website modificado por indivíduo não autorizado**

Outro website analisado, o <http://www.brazilpersontour.com.br/>, apresentou falhas em seus links, direcionando o usuário a conteúdos ainda não existentes como no caso de alugueis de espaços no link de eventos, onde o usuário não encontra as informações necessárias para o mesmo. Isto está demonstrado na figura 3.2 abaixo:



**Figura 3.2 – Website com informações inexistentes**

Outro website analisado falhou em relação ao requisito de qualidade “usabilidade”. No website [www.aviagem.com.br](http://www.aviagem.com.br), o usuário pode fazer pesquisa por destinos

variados, mas o website não apresenta de forma clara quais os destinos que a aplicação possui armazenado ao invés de apresentar uma lista de destinos, o que induz ao erro do usuário. O Website não encontrou a cidade de Barbacena como demonstrado na figura 3.3.

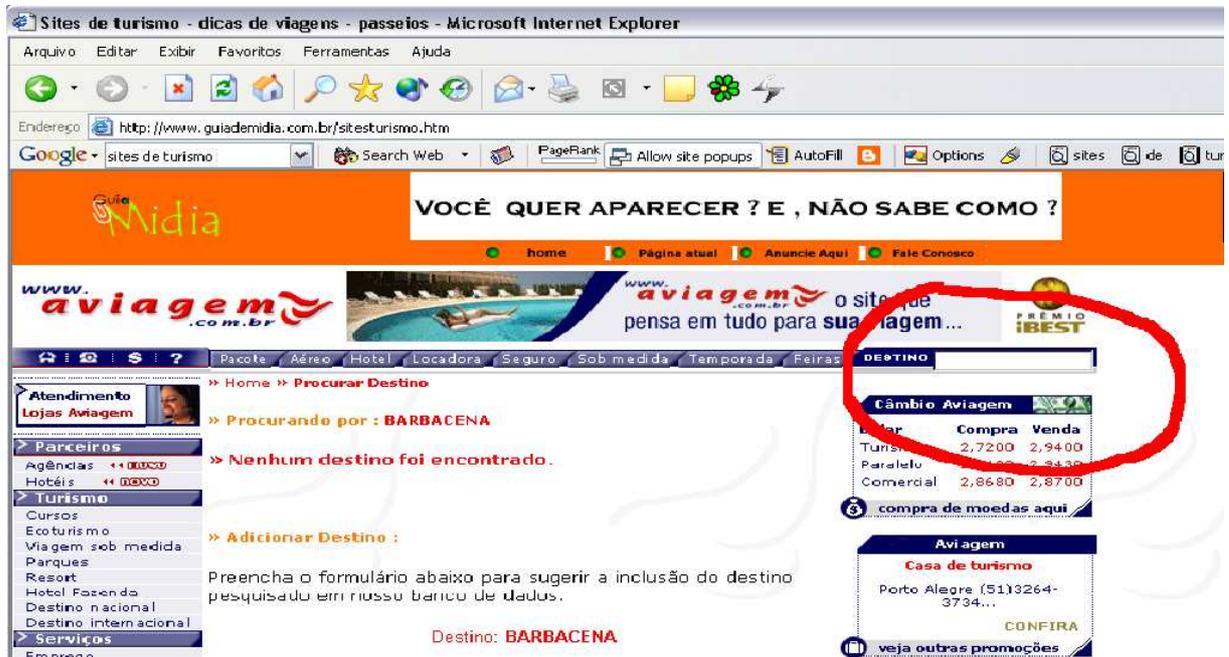


Figura 3.3 – Falha na pesquisa de destinos pelo website.

## 3.2 ESPECIFICAÇÃO DE REQUISITOS DA APLICAÇÃO

### 3.2.1 REQUISITOS FUNCIONAIS

#### 3.2.1.1 Inserção e Manutenção de Informações

- a) o usuário entrará com o dado referente ao destino, meio de transporte e hospedagem desejada.
- b) o Web Master será o responsável pela inserção e manutenção dos referentes dados:
  - Destino

Dado um determinado destino, é apresentado ao usuário as condições climáticas em determinadas estações do ano, informando as temperaturas médias nesses períodos e ainda informar as condições climáticas atualizadas e se possível, com previsões. São informados também, quais as atrações turísticas do destino, como festas regionais, culinária e opções alternativas de turismo na região do destino.

- Transporte

São informadas as formas de transporte existentes para se chegar ao destino desejado, por exemplo, transporte aéreo, transporte terrestre, transporte fluvial, indicando os locais de embarque e desembarque, os respectivos valores, o tempo estimado de viagem para cada tipo de transporte e as datas possíveis para partida e retorno. São informados ainda, quais são as opções de ligação aeroporto-hotel-aeroporto.

- Hospedagem

Nas informações referentes a hospedagem são informados os tipos de hospedagem existentes como hotéis, pousadas, áreas de camping, com as respectivas localizações, os valores para quartos individuais, para casais ou ainda coletivos, os detalhes dos serviços oferecidos pelo local de hospedagem, bem como opções extras de serviços e a disponibilidade de vagas ou reservas para determinadas datas.

- c) a aplicação deve permitir alterações realizadas pelo *Web* master nos itens listados no item b.
- d) a aplicação deve permitir exclusão de informações pelo *Web* master nos itens listados no item b.

- a) a aplicação deve informar periodicamente, quais os destinos mais consultados.

### **3.2.1.2 Consultas da aplicação**

- a) Na aplicação *Web*, o usuário contará com uma pesquisa ampla porém objetiva; ele fornecerá o destino pretendido. Com isso, serão apresentadas as informações referentes ao destino como clima e atrações turísticas em suas respectivas épocas do ano. Por exemplo, digitado o destino pretendido “Barbacena” a aplicação apresentará as médias de temperaturas em determinadas épocas, as festas regionais com suas respectivas datas previstas e quais os pontos turísticos da cidade.
- b) Dado o destino, serão apresentados os meios de transporte disponíveis e a partir da escolha do usuário, serão apresentados seus horários e outras informações de acordo com o item 3.2.1.1.b.
- c) Definido o meio de transporte, a aplicação requisita do usuário, o tipo de hospedagem desejado e apresenta as opções com suas respectivas disponibilidades de vagas, valores e outras informações de acordo com o item 3.2.1.1.b.
- d) A velocidade de consulta, se tratando de uma aplicação *Web*, dependerá da velocidade de conexão do usuário. No caso de linhas discadas, que seria a princípio, a pior das hipóteses, a consulta não deve levar mais que 20 segundos.

## **3.3 REQUISITOS DE QUALIDADE**

### **3.3.1 CONFIABILIDADE**

A aplicação deve ser concisa e objetiva, apresentando dados atualizados e corretos, e apresentados de forma organizada. Os dados a serem informados ao usuário são fornecidos por entidades responsáveis por turismo como agências turísticas, secretarias

municipais e estaduais de turismo, hotéis, pousadas e empresas de transporte. Cabe ao *Web* master apresentá-las com exatidão e responsabilmente, já que o objetivo da aplicação é que o usuário não necessite de confirmações posteriores.

### **3.3.2 EFICIÊNCIA**

A aplicação deve prever conexões de baixa velocidade, portanto, seu conteúdo hipermídia deve ser apresentado de forma mais coerente possível para que as respostas às consultas sejam apresentadas em tempo confortável ao usuário.

### **3.3.3 PORTABILIDADE**

A aplicação *Web* deve ser desenvolvida em linguagem comum a todos os tipos de browsers do mercado. Assim sendo, poderá ser adaptado para servidores Unix, Linux ou da plataforma Windows. A maioria das aplicações existente hoje em dia, são desenvolvidas em linguagem PHP ou ASP, utilizando o banco de dados MySQL que é um software freeware.

Os requisitos de qualidade apresentados acima, são requisitos analisados na fase de implementação e testes da aplicação voltada para web. A modelagem proposta por esse trabalho aborda outros tipos de requisitos de qualidade:

### **3.3.4 MANUTENABILIDADE**

As alterações nos conteúdos que são freqüentes e às vezes tem que serem imediatas, não interferem nas características da estrutura global da aplicação mantendo sempre a estrutura inicial.

### **3.3.5 ENTENDIBILIDADE**

Através de modelos graficamente representados, o desenvolvedor compreende melhor o que deve ser implementado. O uso de padrões para elaboração dos diagramas na

modelagem, como o UML que foi utilizado neste caso, facilita o entendimento global para todos os desenvolvedores.

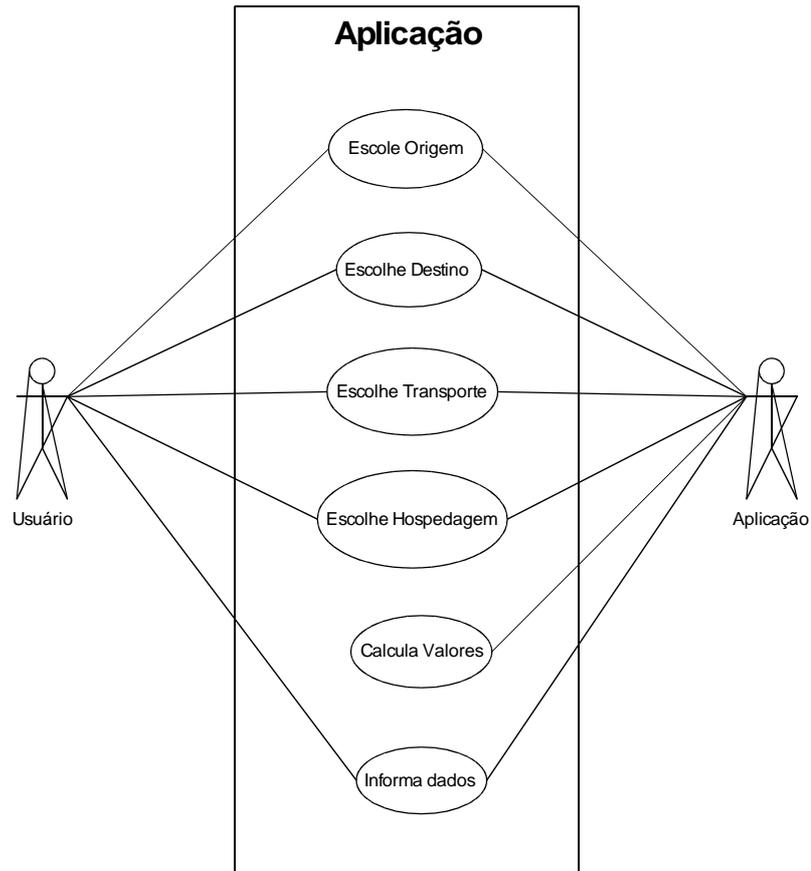
### **3.3.4 FLEXIBILIDADE**

As alterações nas estruturas apresentadas podem ser feitas desde que não se perca o foco inicial da aplicação. Isto não impede que o desenvolvedor mude a estrutura, mas deve sempre estar atento ao objetivo principal que é a objetividade com que a aplicação apresenta as informações.

## 4 MODELAGEM DA APLICAÇÃO DE TURISMO

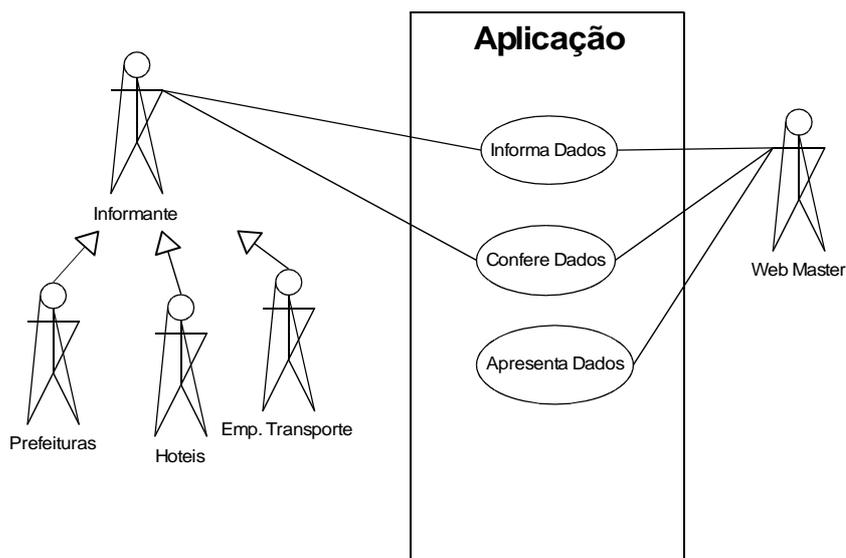
### 4.1 DIAGRAMAS DE CASOS DE USO

Os diagramas de caso de uso são importantes para apresentar os comportamentos dos elementos do sistema [PRE2002]. No caso da aplicação de Turismo, são apresentados dois casos de interação. No primeiro, representado pela Figura 4.1, é apresentada a parte da aplicação que representa a interação entre o usuário da aplicação e a entidade agência que trata de apresentar as informações solicitadas pelo usuário. No segundo, representado pela figura 4.2, é apresentada a parte da aplicação que representa a interação entre os informantes, no caso, prefeituras, hotéis e empresas de transporte. e o *Web* master que é o responsável pela apresentação desses dados na aplicação.



**Figura 4.1 – Caso de Uso Usuário/Aplicação**

No diagrama de caso de uso Usuário/Aplicação, o usuário escolhe a cidade de origem e a aplicação verifica e apresenta os dados referentes à cidade de origem. O usuário escolhe a cidade de destino e a aplicação verifica e apresenta os dados referentes à cidade de destino. A aplicação então, verifica e apresenta os dados referentes ao destino. A aplicação disponibiliza os tipos de transporte existentes para se levar ao destino. A aplicação apresenta as possibilidades de hospedagem. O usuário escolhe o tipo de hospedagem e a aplicação calcula os valores totais e apresenta o resultado final ao usuário.



**Figura 4.2 – Caso de Uso Informante/Web Master**

No caso de uso Informante/Web Master, os diversos tipos de informantes, prefeituras, hotéis/pousadas e empresas de transporte informam os dados referentes a suas respectivas áreas e o web master confere a integridade das informações. Uma vez conferidos, os dados devem retornar aos informantes para confirmação formal das informações. Dado o sucesso do recebimento das informações, o web master apresenta as informações na aplicação.

## 4.2 CLASSES E EVENTOS

### 4.2.1 CLASSE CIDADE

A classe CIDADE a classe definida para armazenar informações referentes as cidades de origem e destino escolhidas pelo usuário. Nela está contida o CODIGO da CIDADE, que é a identificação única da mesma, o NOME que é a descrição da cidade e UF que identifica o Estado a que pertence a cidade.

Seus eventos são: CADASTRAR - que armazena as informações no Banco de Dados retornando falso se algum erro for detectado ou verdadeiro caso a inserção ocorra com

sucesso, CONSULTAR - que retorna os dados referentes a classe e demais informações agregadas, EDITAR – que altera os dados cadastrados retornando falso se algum erro for detectado ou verdadeiro caso a edição ocorra com sucesso e APAGAR – que elimina o registro do Banco de Dados retornando falso se algum erro for detectado ou verdadeiro caso a exclusão ocorra com sucesso.

#### **4.2.2 CLASSE ATRA\_TUR**

A classe ATRA\_TUR armazena os dados referentes às atrações turísticas existentes em uma determinada cidade dentre eles a descrição da atração, o tipo da atração (festas, pontos turísticos, etc.) e os períodos em que ocorrem tais atrações. No caso de pontos turísticos, os valores de período serão referentes às melhores datas para se visitar o local.

Seus eventos são: CADASTRAR - que armazena as informações no Banco de Dados retornando falso se algum erro for detectado ou verdadeiro caso a inserção ocorra com sucesso, CONSULTAR - que retorna os dados referentes a classe e demais informações agregadas, EDITAR – que altera os dados cadastrados retornando falso se algum erro for detectado ou verdadeiro caso a edição ocorra com sucesso e APAGAR – que elimina o registro do Banco de Dados retornando falso se algum erro for detectado ou verdadeiro caso a exclusão ocorra com sucesso.

#### **4.2.3 CLASSE CULINÁRIA**

A classe CULINARIA armazena dados referentes a culinárias típicas que possuem as cidades referenciadas e com os hotéis que possuem culinária diferenciada.

Seus eventos são: CADASTRAR - que armazena as informações no Banco de Dados retornando falso se algum erro for detectado ou verdadeiro caso a inserção ocorra com sucesso, CONSULTAR - que retorna os dados referentes a classe e demais informações agregadas, EDITAR – que altera os dados cadastrados retornando falso se algum erro for detectado ou verdadeiro caso a edição ocorra com sucesso e APAGAR – que elimina o

registro do Banco de Dados retornando falso se algum erro for detectado ou verdadeiro caso a exclusão ocorra com sucesso.

#### **4.2.4 CLASSE CLIMA**

A classe CLIMA armazena dados referentes às condições climáticas em determinados períodos do ano, as temperaturas médias nos períodos e o tipo de clima geral da localidade.

Seus eventos são: CADASTRAR - que armazena as informações no Banco de Dados retornando falso se algum erro for detectado ou verdadeiro caso a inserção ocorra com sucesso, CONSULTAR - que retorna os dados referentes a classe e demais informações agregadas, EDITAR – que altera os dados cadastrados retornando falso se algum erro for detectado ou verdadeiro caso a edição ocorra com sucesso e APAGAR – que elimina o registro do Banco de Dados retornando falso se algum erro for detectado ou verdadeiro caso a exclusão ocorra com sucesso.

#### **4.2.5 CLASSE EMPRESA**

A classe EMPRESA armazena informações sobre as empresas de transporte ou hospedagem que disponibilizam serviços para efetuar o transporte dos clientes de sua origem e destino e a hospedagem disponível para a determinada cidade de destino. Esta classe está dividida em outras duas subclasses Emp\_Transporte e Emp\_Hospedagem pois, as características de cada uma são distintas mas compartilham atributos em comum.

Seus eventos são: CADASTRAR - que armazena as informações no Banco de Dados retornando falso se algum erro for detectado ou verdadeiro caso a inserção ocorra com sucesso, CONSULTAR - que retorna os dados referentes a classe e demais informações agregadas, EDITAR – que altera os dados cadastrados retornando falso se algum erro for detectado ou verdadeiro caso a edição ocorra com sucesso e APAGAR – que elimina o registro do Banco de Dados retornando falso se algum erro for detectado ou verdadeiro caso a exclusão ocorra com sucesso.

#### **4.2.7 CLASSE VAGAS**

A classe VAGAS tem a função de armazenar informações referentes a vagas disponíveis nos hotéis e pousadas. São informados o tipo de vaga (casal, solteiro ou coletivo), o número de vagas disponíveis e o valor das mesmas.

Seus eventos são: CADASTRAR - que armazena as informações no Banco de Dados retornando falso se algum erro for detectado ou verdadeiro caso a inserção ocorra com sucesso, CONSULTAR - que retorna os dados referentes a classe e demais informações agregadas, EDITAR - que altera os dados cadastrados retornando falso se algum erro for detectado ou verdadeiro caso a edição ocorra com sucesso e APAGAR - que elimina o registro do Banco de Dados retornando falso se algum erro for detectado ou verdadeiro caso a exclusão ocorra com sucesso.

#### **4.2.8 CLASSE SERVIÇOS**

A classe SERVIÇOS armazena os serviços básicos e extras das hospedagens disponíveis na cidade.

Seus eventos são: CADASTRAR - que armazena as informações no Banco de Dados retornando falso se algum erro for detectado ou verdadeiro caso a inserção ocorra com sucesso, CONSULTAR - que retorna os dados referentes a classe e demais informações agregadas, EDITAR - que altera os dados cadastrados retornando falso se algum erro for detectado ou verdadeiro caso a edição ocorra com sucesso e APAGAR - que elimina o registro do Banco de Dados retornando falso se algum erro for detectado ou verdadeiro caso a exclusão ocorra com sucesso.

#### **4.2.9 CLASSE AGÊNCIA**

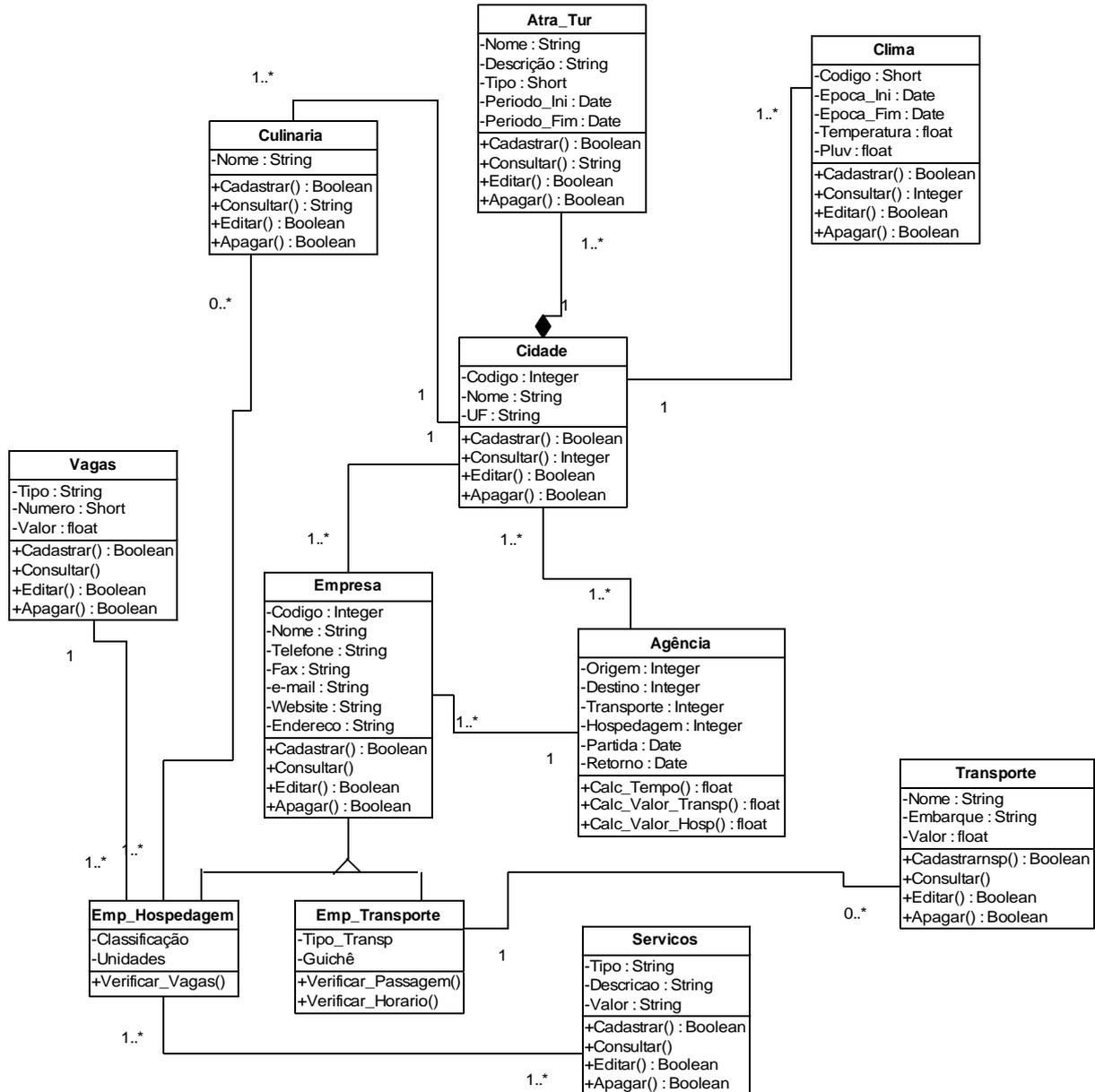
A classe AGÊNCIA foi determinada para receber informações das classes CIDADE e EMPRESA para que nela sejam tratadas as informações e apresentar os resultados para o usuário.

Seus eventos são: CALC\_Tempo – Que calcula o tempo estimado de viagem entre a cidade de origem e a cidade de destino, CALC\_VALOR\_TRANSP – que calcula o valor individual e total do transporte do usuário e CALC\_VALOR\_HOSP – que calcula os valores referentes à hospedagem escolhida pelo usuário.

### **4.3 DIAGRAMA DE CLASSES**

Uma classe é a descrição de um conjunto de objetos que compartilham os mesmos atributos, operações, relacionamentos e semântica. Atributos são qualidades específicas de classes, isto é, um atributo é uma determinada característica da classe [GRA2000].

Abaixo é apresentado pela figura 4.3 o diagrama de classes para a aplicação de turismo.

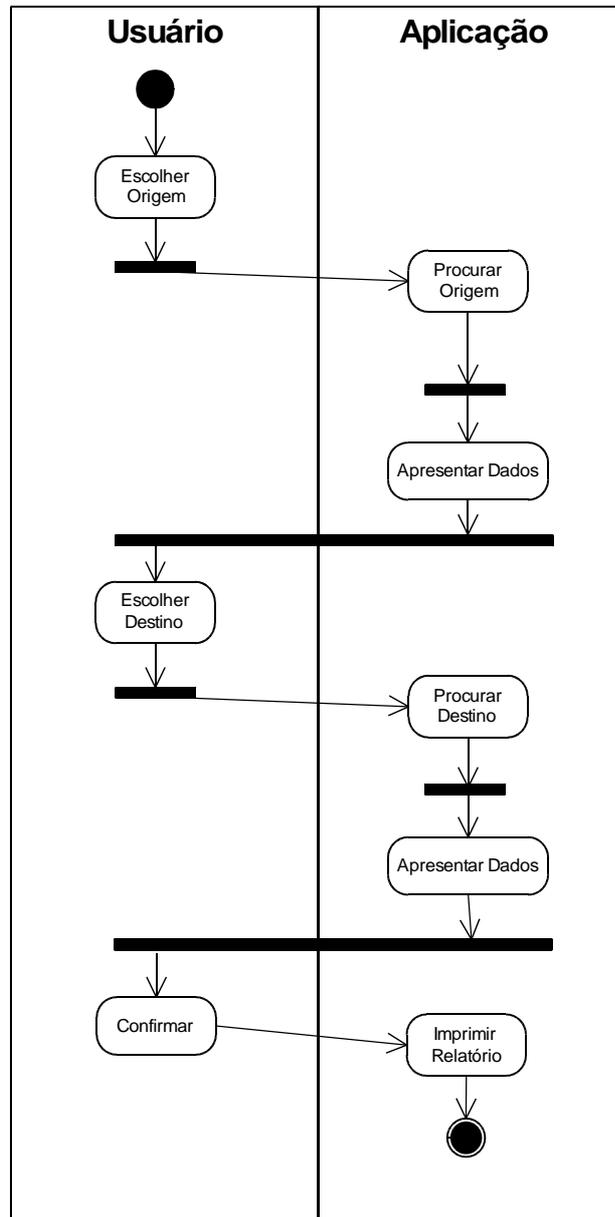


**Figura 4.3 – Diagrama de Classes**

A utilização da herança das classes *Emp\_Hospedagem* e *Emp\_Transporte* foi necessária porque tratando-se de empresas de transporte e hospedagem, as classes possuem atributos em comum, porém possuem atributos específicos de cada subclasse. Isso garante ao modelo uma consistência maior em manutenibilidade pela facilidade em fazer alterações nas subclasses, a compreensibilidade do modelo melhora pelo fato das subclasses serem apenas a parte distinta da classe e a flexibilidade, pois novas subclasses podem ser incluídas sem alterar o modelo drasticamente.

## 4.4 DIAGRAMAS DE ATIVIDADES

O diagrama de atividades visa primeiramente demonstrar graficamente o processo ocorrido em determinada parte da aplicação. Na figura 4.4 é mostrado o diagrama de atividade na consulta de um usuário a uma determinada origem escolhendo-se o destino.



**Figura 4.4 – Diagrama de Atividade**

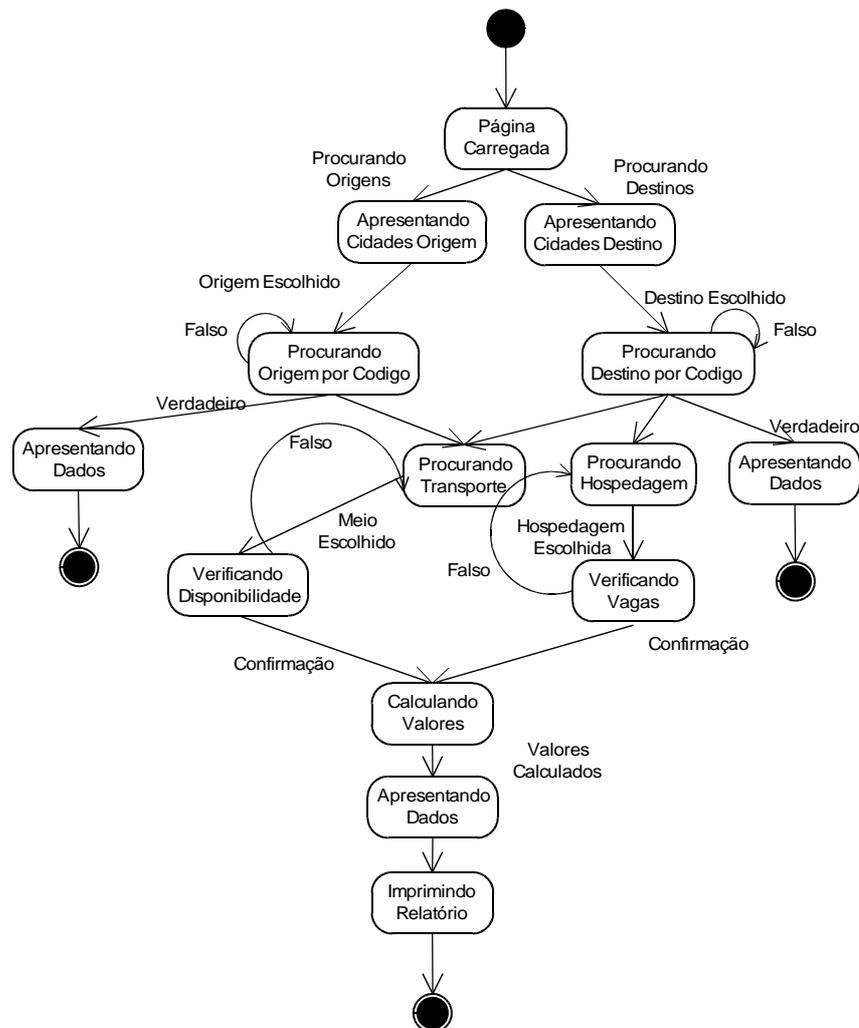
Primeiramente o usuário escolhe a origem de partida, a aplicação procura os dados referentes à cidade de origem e os apresenta ao usuário. Com isso o usuário está apto a

escolher a cidade de destino. A aplicação procura os dados referentes ao destino e apresenta ao usuário. Com a confirmação do usuário a aplicação informa as opções de transporte e hospedagem para o usuário. Uma vez escolhidos o transporte e a hospedagem, o programa imprime um relatório com todos os dados.

## 4.5 DIAGRAMAS DE ESTADOS

O diagrama de estados na UML mostra a seqüência de estados que um objeto se apresenta em determinados pontos da aplicação com suas respectivas ações, eventos e condições de espera, associados com suas transições. [NOR2002]

A figura 4.5 abaixo apresenta o diagrama de estados da aplicação de turismo.



### **Figura 4.5 – Diagrama de Estados**

No momento inicial, a página principal da aplicação é apresentada. A aplicação exibe uma cidade origem padrão e uma cidade destino padrão para facilitar o entendimento do usuário. Escolhida a cidade de origem, a aplicação procura por código os dados referentes a essa cidade e os apresenta. Quando o usuário escolhe a cidade de destino, a aplicação procura por código da cidade os dados referentes e disponibiliza a pesquisa de meio de transporte e meio de hospedagem existentes para essa cidade, verificando vagas e horários. Caso seja confirmado o destino, transporte e hospedagem, a aplicação calcula os valores totais e os apresenta em relatório e na própria aplicação.

## 5 CONCLUSÕES

### 2.7 REVISÃO

Esta monografia apresentou em seu segundo capítulo uma visão geral sobre a Engenharia de Software para Web, apresentando os atributos de qualidade de uma aplicação voltada para Web como Atributos de Qualidade, Segurança e Padrões da Internet, Formulação e análise de um Sistema Voltado para Web, Modelos de aplicações voltadas para Web, e ainda, os Testes a serem feitos e a gerência do projeto.

No terceiro capítulo, a monografia apresentou as etapas do desenvolvimento de uma aplicação de turismo voltada para Web, detalhando o seu plano de desenvolvimento, a especificação de requisitos e requisitos funcionais, os requisitos de qualidade que englobam a confiabilidade da aplicação, eficiência e a portabilidade.

No quarto capítulo, foram apresentados os modelos gerais da aplicação de turismo voltada para Web. Foram apresentados os diagramas de caso de uso, a descrição das classes e seus respectivos eventos, o diagrama de classes, o diagrama de atividades e finalmente o diagrama de estados.

### 2.8 CONCLUSÕES

A utilização das técnicas de modelagem da engenharia de software no desenvolvimento de aplicações voltadas para *Web* é ainda hoje uma ferramenta pouco utilizada pelos desenvolvedores. O desconhecimento de tais técnicas e metodologias ou ainda a ideologia de que aplicações voltadas para *Web* não necessitam de tratamento especial em

seu projeto, vem fazendo com que diversas aplicações *Web* hoje em utilização na rede mundial de computadores tenham seus aspectos fundamentais de qualidade questionados pelos usuários.

Um exemplo prático desse fato, são as aplicações voltadas para prestar informações turísticas a clientes de agências turísticas. Na maioria delas, as informações estão dispostas de forma confusa e ineficientemente apresentadas aos usuários.

A utilização de técnicas de modelagem da Engenharia de Software para o desenvolvimento de aplicações voltadas para *Web*, tem importância fundamental para se chegar a uma aplicação de qualidade.

Com isso, as modelagens apresentadas, visam ajudar aos desenvolvedores de *Web* sites turísticos servindo de um ponto de partida para o desenvolvimento da aplicação de turismo voltada para *Web*. A partir dos modelos básicos apresentados, os desenvolvedores podem adaptar os mesmos às suas necessidades específicas sem alterar a característica principal das mesmas que é implantar a Engenharia de Software para a aplicação voltada para *Web* utilizando esses modelos para o desenvolvimento de uma aplicação com qualidade.

No modelo proposto por esse trabalho, o usuário tem mais possibilidades de chegar ao resultado desejado pois as informações são mais completas e mais específicas do que as apresentadas pelas aplicações analisadas. Os itens que diferenciam a aplicação voltada para *web* proposta por essa monografia, que caracterizam a superioridade em relação aos demais sites analisados, <http://www.brazilpersontour.com.br/>, <http://ipanorama.globo.com/> e [www.aviagem.com.br](http://www.aviagem.com.br), estão descritos a seguir:

- Nenhum dos websites analisados disponibilizam a opção de se escolher uma determinada cidade de origem o que impossibilita ao usuário de planejar um roteiro turístico. No website <http://ipanorama.globo.com/>, são apresentados apenas os links para as páginas da cidade, onde a maioria delas não apresenta as informações apresentadas pelo modelo proposto.

- O modelo proposto apresenta informações sobre meios de transportes e o que ocorre no site <http://www.brazilpersontour.com.br/> por exemplo, é apresentado ao usuário apenas os links para companhias aéreas, o que dificulta a pesquisa do usuário.
- O modelo apresentado reúne todas as informações necessárias para o usuário dispostas de maneira objetiva e clara. Nos demais websites, essas informações estão em páginas diferentes o que dificulta a pesquisa do usuário.

Um fato que deve ser considerado é que o modelo proposto resulta de uma idéia inovadora para websites de turismo. Não existe ainda uma aplicação que faça tudo o que o modelo propõe para que sejam feitas comparações diretas entre os modelos. No entanto, levando em consideração que um usuário acessa um website de informações turísticas para realizar uma viagem, as aplicações existentes hoje na rede mundial de computadores falham no fornecimento das informações necessárias para realizar uma viagem seja por falta de informação ou desorganização das informações apresentadas.

## **2.9 RECOMENDAÇÕES**

As ferramentas de modelagem são muito versáteis e funcionais. Começando por esse princípio, o desenvolvedor deve sempre começar seu projeto de software obedecendo as metodologias e técnicas da Engenharia de Software como a elaboração dos diagramas apresentados nesse trabalho.

Cabe aos que pretendem implementar a aplicação de turismo voltada para web, tomar esse documento como passo inicial e de acordo com suas necessidades, aprimorá-lo e desenvolver novas modelagens à partir do que se encontra disponível nesse documento.

As ferramentas de implementação de aplicações Web são muito variadas. Devido a flexibilidade de algumas linguagens, estas são mais utilizadas. A linguagem PHP por exemplo, oferece ao desenvolvedor vários recursos para que a aplicação atenda todos os seus requisitos. Então, poderia facilmente atender a aplicação de turismo voltada para web

trabalhando com o banco de dados MySQL, que além de ser um software gratuito, atende a todas as especificações de um gerenciador de banco de dados.

Quanto a interface, as ferramentas são mais diversificadas ainda. Aplicações desenvolvidas em Macromedia Dreamweaver e Macromedia Flash, possibilitam ao web designer preparar uma interface confortável e com recursos avançados de design.

A escolha das ferramentas de implementação e de interface gráfica varia de desenvolvedor para desenvolvedor. Nesse caso, o que importa é que essa escolha não altere a estrutura apresentada nesta monografia e que, não torne desconfortável a navegação do usuário na aplicação.

## REFERÊNCIAS BIBLIOGRÁFICAS

- [PRE2002] PRESSMAN, Roger S. **Software Engineering**: McGraw-Hill, 2002.
- [GEN2000 et. al.] Gennaro Costagliola, Filomena Ferrucci, Rita Francese. **Web Engineering: Models and Methodologies for the design of Hypermedia Applications**: Dipartimento di Matematica e Informatica, Università di Salerno Via S. Allende, Italy 2000.
- [NOR2000 et. AL.] Nora Koch, Andreas K., **The Expressive Power of UML – based Web Engineering**: Ludwig-Maximilians-Universität München, Germany 2000.
- [GRA2000] Grady B., James R., Ivar J., **UML Guia do Usuário**: Ed. Campus, Rio de Janeiro – Brasil, 2002.
- [BUS 96] Bushmann, M. et. al., **Pattern-Oriented Software Architecture**, Wiley, 1996.
- [GAM95] Gamma, E., et. Al., **Design Patterns**, Addison-Wesley, 1995.
- [BER98] Bernstein, M., **Patterns in Hypertext**, Proc. 9<sup>th</sup> ACM Conf. Hypertext, ACM Press, 1998, pp. 21-29.
- [NIE96] Nielsen, J. and A. Wagner, **User Interface Design for the WWW**, Proc. CHI 96 conference on Human Factors in Computing Systems, ACM Press, 1996, pp. 330-331.
- [OLS99] Olsina, L., et. al., **Specifying Quality Characteristics and Attributes for Web Sites**, Proc. First ICSE Workshop on Web Engineering, ACM, Los Angeles, 1999.

## ANEXO I – INTERFACE PREVISTA

### Aplicação de turismo Voltada para Web

<b>Escolha a origem pretendida:</b> Rio de Janeiro <input type="button" value="OK"/>	<b>Escolha o destino pretendido:</b> Barbacena <input type="button" value="OK"/>	<b>Meio de Transporte</b> Rodoviário <input type="button" value="OK"/>	<b>Hospedagem</b> Hotel <input type="button" value="OK"/>
<b>Informações turísticas</b> Estado: RJ Clima: Janeiro a Março: Quente Média de Temperatura: 38º Probabilidade de Chuvas: 50% Maio a Junho: Moderado	<b>Informações turísticas</b> Estado: MG Clima: Janeiro a Março: Moderado Média de Temperatura: 30º Probabilidade de Chuvas: 50% Maio a Junho: Frio	<b>Empresa de transporte:</b> Empresa: Util Telefone: (21)3213-3213 Fax:(21)3213-3241 website: www.util.com.br e-mail: informações@util.com.br Endereço: Av. Basil, 09	<b>Empresa:</b> Empresa: Hotel Lucape Telefone: (32)3333-5234 Fax:(32)3332-4561 website: www.lucape.com.br e-mail: informações@Lucape.com.br Endereço: Pça. dos Andradas, 80
<b>Atrações Turísticas</b> Nome: Carnaval Descrição: Festa popular nacional - Samba Tipo: Festa Início: 28/02 Fim: 04/03	<b>Atrações Turísticas</b> Nome: Festa das Rosas e Flores Descrição: exposições de Rosas e Shows Tipo: Festa Início: 10/10 Fim: 17/10	<b>Locais de embarque:</b> Rodoviária Principal Valor de Passagem: R\$25,00 Horários: 08:00 - 12:00 - 18:00 - 23:45	<b>Serviços adicionais</b> Café da Manhã Jantar Piscina Estacionamento  <b>Vagas:</b> Tipo: Solteiro Número: 5 Valor individual: R\$30,00