

UNIVERSIDADE REGIONAL DE BLUMENAU
CENTRO DE CIÊNCIAS EXATAS E NATURAIS
CURSO DE SISTEMAS DE INFORMAÇÃO – BACHARELADO

SISTEMA PARA GESTÃO E DIVULGAÇÃO DE AMBIENTES
GASTRONÔMICOS

RION BRATTIG CORREIA

BLUMENAU
2008

2008/1-13

RION BRATTIG CORREIA

SISTEMA PARA GESTÃO E DIVULGAÇÃO DE AMBIENTES

GASTRONÔMICOS

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

Prof. Alexander Roberto Valdameri - Orientador

**BLUMENAU
2008**

2008/1-13

SISTEMA PARA GESTÃO E DIVULGAÇÃO DE AMBIENTES GASTRONÔMICOS

Por

RION BRATTIG CORREIA

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

Presidente: _____
Prof. Alexander Roberto Valdameri, Mestre – Orientador, FURB

Membro: _____
Prof. Paulo Fernando, Mestre – FURB

Membro: _____
Prof. Everaldo Artur Grahl, Mestre – FURB

Blumenau, 08 de Julho de 2008

Dedico este trabalho a deus, minha mãe e meus amigos, que com seu apoio, amor e carinho me ajudam a transpor qualquer obstáculo em minha vida.

AGRADECIMENTOS

À Deus, pelo seu imenso amor e graça e por nunca me abandonar em qualquer aspecto de minha vida, sempre me mostrando o caminho e a solução para os meus problemas.

À minha mãe, que sempre me incentivou em todos os meus momentos, sempre exaltou meu lado bom e ajudou a corrigir minhas falhas.

Aos meus amigos, pela paciência com que me suportam, incentivam e participam dos principais momentos de minha vida.

À família Zinabre.com que sempre me ajudou a crescer e a aprender intelectual e espiritualmente.

Ao meu orientador, Alexander Roberto Valdameri, por ter acreditado na conclusão deste trabalho.

Toda a minha vida eu tive um sonho, de conseguir todos os meus objetivos.

Homer Simpson

RESUMO

Este trabalho apresenta a implementação de um sistema web para a catalogação e divulgação de ambientes gastronômicos, mais conhecidos como “Botecos” sendo o ponto auge do sistema a divulgação da “Baixa Gastronomia”. Foi utilizada a WebML em seus modelos de dados e de hiperlinks para transpor os requisitos do sistema em sua especificação. Foi também implementado segundo técnicas de usabilidade na web para ter uma fácil interação com o usuário. A técnica de programação conhecida como Ajax foi usada para aumentar esta interação. A API do Google Maps foi utilizada para mostrar a localização dos estabelecimentos diretamente em um mapa. O software permite, por meio de uma área administrativa, o cadastramento de “Botecos”, fotos, vídeos e personalidades para estes e ainda a inclusão de notícias na capa do sistema na forma de Blog.

Palavras-chave: Boteco. Baixa gastronomia. WebML. Usabilidade na web. Google Maps.

ABSTRACT

This work shows an implementation of a web based system to catalog and disseminate the gastronomic environments best known as “Botecos” being The strongest point in the system is the dissemination of the “low gastronomy”. WebML was used both to build the data model and the hyperlinks model to interpret the system requirements specification. Was also implemented under techniques of web usability to have an easier interaction with the user. The technique known as Ajax was used to increase this interaction. Google Maps API was used to show the location of the environments directly on a map. The software allows, by means of an administrative area, register the “Botecos”, pictures, movies and personalities for those and also the inclusion of news to the front of the system in Blog style.

Palavras-chave: Boteco. Low gastronomy. WebML. Web usability. Google Maps.

LISTA DE ILUSTRAÇÕES

Figura 1 – Exemplo de um modelo de dados da WebML.....	21
Figura 2 – Exemplo de um modelo de hiperlinks da WebML.....	22
Figura 3 – Tela da ferramenta CASE WebRatio versão 5.0.0.....	24
Quadro 1 - Requisitos funcionais	26
Quadro 2 - Requisitos não funcionais	27
Quadro 3 - Regras de negócio	27
Figura 4 - Modelo de Dados (a)	28
Figura 5 - Modelo de Dados (b)	28
Figura 6 - Modelo de Dados (c)	29
Figura 7 – Modelo de Hiperlinks do Requisito RF01 (Cadastra boteco).....	30
Figura 8 - Modelo de Hiperlinks do requisito RF02 (Cadastra foto para boteco).....	31
Figura 9 - Modelo de Hiperlinks do requisito RF03 (Cadastra vídeos para o boteco).....	32
Figura 10 - Modelo de Hiperlinks do requisito RF04 (Cadastra personalidades para o boteco)	33
Figura 11 - Modelo de Hiperlinks do requisito RF05 (Cadastra notícias)	33
Figura 12 - Modelo de Hiperlinks do requisito RF05 (Cadastra usuário para a notícias).....	34
Figura 13 - Modelo de Hiperlinks do requisito RF06 (Cadastra receitas).....	34
Figura 14 - Modelo de Hiperlinks do requisito RF07 (Cadastro de mídias).....	35
Figura 15 - Modelo de Hiperlinks da RN01 (Cadastro de campanha para mídia).....	36
Figura 16 - Modelo de Hiperlinks da RN01 (Cadastro de empresa para a mídia)	36
Figura 17 – Tiny MCE	37
Figura 18 - Usabilidade no sistema	38
Quadro 4 - URL normal com passagem de parâmetro	39
Quadro 5 - Clean URL	39
Quadro 6 - Código do <i>ModRewrite</i>	39
Quadro 7 - Select Field de estado	40
Quadro 8 - Select Field de cidade.....	40
Quadro 9 - Criação do Objeto <i>XMLHttpRequest</i>	40
Quadro 10 - Função <i>ajaxPopulaCombo</i>	41
Figura 19 - Utilização do Boteco Maps.....	42
Quadro 11 - Dados XML gerados para o Google Maps API	42

Figura 20 - Google Analytics	44
Figura 21 - Relação de dados Boteco Categoria.....	45
Quadro 14 - Comando SQL de pesquisa de Boteco	45
Figura 22 - Tela de entrada do Admin.....	46
Figura 23 - Tela da relação de Botecos cadastrados.....	47
Figura 24 - Tela de inclusão/alteração de um "Boteco"	48
Figura 25 - Tela de fotos cadastradas para um determinado "boteco"	49
Figura 26 - Tela de inserção de foto para determinado "boteco"	49
Figura 27 – Tela do “boteco” para o usuário final	50
Figura 28 - Tela de notícias cadastradas.....	51
Figura 29 - Tela de cadastro de nova notícia.....	51
Figura 30 - Tela de notícias para o usuário final	52

LISTA DE SIGLAS

API – Interface de Programação de Aplicativos (Application Programming Interface)

ABNT – Associação Brasileira de Normas Técnicas

CSS – Cascading Style Sheet

HTML – HyperText Markup Language

J2EE – Java 2 Enterprise Edition

XML – *Extensible Markup Language*

WebApps – Web-based Systems and Applications

WebML – The Web Modeling Language

WWW – World Wide Web

SUMÁRIO

1 INTRODUÇÃO.....	13
1.1 OBJETIVOS DO TRABALHO	15
1.2 ESTRUTURA DO TRABALHO	15
2 FUNDAMENTAÇÃO TEÓRICA	16
2.1 CENÁRIO ATUAL.....	16
2.2 CONCEITOS.....	16
2.2.1 PHP	16
2.2.2 W3C	17
2.2.3 AJAX	17
2.2.4 GOOGLE MAPS API.....	18
2.2.5 USABILIDADE NA WEB	19
2.2.6 WEBML.....	20
2.2.6.1 MODELO DE DADOS	20
2.2.6.2 MODELO DE HIPERLINKS.....	21
2.2.6.3 MODELO DE APRESENTAÇÃO	22
2.2.7 WEBRATIO.....	23
2.3 TRABALHOS CORRELATOS.....	24
3 DESENVOLVIMENTO DO SISTEMA.....	26
3.1 REQUISITOS DO SISTEMA.....	26
3.2 ESPECIFICAÇÃO	27
3.2.1 MODELO DE DADOS	27
3.2.2 MODELO DE HIPERLINKS	29
3.3 IMPLEMENTAÇÃO	37
3.3.1 FERRAMENTAS UTILIZADAS.....	37
3.3.2 ASPECTOS IMPORTANTES DA IMPLEMENTAÇÃO	38
3.3.2.1 QUESTÕES DE USABILIDADE.....	38
3.3.2.2 AJAX	39
3.3.2.3 GOOGLE MAPS API.....	41
3.3.2.4 GOOGLE ANALYTICS	43
3.3.2.5 PESQUISA AVANÇADA DE BOTECOS.....	44
3.4 OPERACIONALIDADES DA IMPLEMENTAÇÃO.....	46

3.4.1 INCLUSÃO DE UM BOTEÇO.....	46
3.4.2 INCLUSÃO DE FOTOS PARA UM BOTEÇO	48
3.4.3 INCLUSÃO DE NOTÍCIAS.....	50
3.5 RESULTADOS E DISCUSSÕES.....	52
4 CONSIDERAÇÕES FINAIS	54
4.1 EXTENSÕES	54
REFERÊNCIAS BIBLIOGRÁFICAS	56
APÊNDICE A – Código usado para gerar o banco de dados do sistema.....	58

1 INTRODUÇÃO

O brasileiro e tradicional “boteco”, aquele de esquina, pequeno, de ladrilhos nas paredes e ventiladores de teto, está em alta novamente entre os jovens de classe média de grandes cidades. Jovens estes que saem pela noite à procura de lugares inóspitos que possam proporcionar um ambiente hospitaleiro, cerveja gelada, preços honestos e principalmente: a Baixa Gastronomia. Comida (geralmente petiscos) simples, porém com toques extremamente sofisticados que somente o proprietário do “Boteco” sabe proporcionar. Do rollmops¹, passando pelo pão-com-bolinho, até o conhecido PF², a baixa gastronomia detém diferentes conotações e nomes dependendo da região do Brasil. Mas uma coisa é certa, todos os pratos devem ser saborosos e caírem bem com uma cerveja gelada.

A conotação de ser lugar para pessoas de baixo nível está mudando. Segundo Beneli (2007), “O tradicional bar pé sujo tirou o pé da lama, (...) e virou o mais novo *hype* do momento gastronômico, tudo por causa da pitoresca e saborosa culinária de boteco”.

A palavra "boteco" é diminutivo de "botequim", que, por sua vez, tem a sua origem na palavra "botica", armazém onde se vendia de tudo um pouco no começo do século passado. Os clientes iam para as boticas, faziam compras e aproveitavam para colocar a conversa em dia. Com o tempo, os proprietários das boticas começaram a servir aos fregueses aperitivos junto com uma bebida. Como muitos bares naquela época não eram tidos como locais para "homens de família", as boticas eram uma alternativa e logo se transformaram em um ponto de encontro, aonde os fregueses iam mesmo quando não precisavam abastecer suas despensas. Muitos botecos guardam essa tradição até hoje. (HAMA, 2005)

Aliado a este novo conceito, põem-se em pauta também a já fixada necessidade de utilização por turistas, e até mesmo por pessoas que querem saber o que mais a sua cidade têm a oferecer, de guias como, por exemplo, o Guia Quatro Rodas. Guia este que anualmente proporciona nas bancas versões atualizadas sobre as diversas categorias que abrange, como restaurantes, hotéis e atrações em todo o Brasil. Dá-se nota para cada estabelecimento que freqüentam, e atribuem comentários que vão para o livro.

Buscando a união entre este dois conceitos, um moderno, outro já muito bem fixado, apresenta-se uma solução para prescrever a utilização de um guia focalizado nesta Baixa Gastronomia que rodeia o Vale do Itajaí. Em outras palavras, unir em apenas um lugar, os diversos lugares que são atualmente procurados pelos jovens, e levar de maneira fácil e

¹ Filé de sardinha enrolado com um pedaço de cebola na conserva.

² Prato-Feito. Típico prato de arroz, feijão, macarrão e um pedaço de carne que têm por definição ser de baixo custo e ser consumido por trabalhadores da massa proletária.

catalogada os “botecos” que esta região dispõe por ter sido tão ricamente povoada em seus anos de descobrimento e colonização.

Atualmente, a Baixa Gastronomia, mesmo possuindo ênfase na mídia nos últimos tempos, não possui meios para ser divulgada a não ser a popular. É sempre divulgada por meio das pessoas que freqüentam cada um dos estabelecimentos, e pelo próprio proprietário.

A empresa Zinabre Ponto Com (Zinabre, 2008) é a responsável pela idéia do projeto. Idéia que surgiu como uma brincadeira entre os organizadores e colaboradores da empresa, por estes serem amantes da Baixa Gastronomia, e acabou virando um assunto sério com a percepção da atual falta de recursos nesta vasta área abrangida neste trabalho.

A empresa foi inicialmente fundada com o propósito de prestar assessoria fotográfica em festas e “baladas” da região de Balneário Camboriú. Ficando regionalmente conhecida e assim expandindo sua área de abrangência por todo o Vale do Itajaí e região. Criou-se então a necessidade de ampliar seu domínio em assessoria. Segmentou-se a empresa em um outra chamada ZNB.web. Empresa responsável pelo desenvolvimento de Web Sites para estas casas noturnas, assim como para outros segmentos do comercio e da indústria.

Atuando no ramo de desenvolvimento de Web Sites à quase quatro anos, a empresa detém um vasto conhecimento no desenvolvimento de projetos para a Web. Vislumbrou na idéia deste novo projeto outra segmentação do seu foco atual. Inicialmente algo que será mantido de dentro da empresa Zinabre Ponto Com e que futuramente possa ser expandido à formação de uma nova empresa.

Não há conhecimento de que se tenha um sistema proposto já em atuação nesta área, o que aumenta ainda mais a necessidade deste projeto. O que se tem atualmente é o trabalho do Guia Quatro Rodas, e de alguns outros guias como o hagam, mas nenhum no segmento a ser abordado.

Seguindo as tendências tecnológicas e a contínua necessidade de padrões de desenvolvimento de sistemas para a web, foi criada a WEBML (*The Web Modeling Language*), que, segundo Locatelli (2003), “auxilia os desenvolvedores a implementarem as principais funcionalidades de um WebSite”. Modelos estes que foram utilizados na modelagem deste trabalho.

1.1 OBJETIVOS DO TRABALHO

O objetivo deste trabalho é desenvolver um sistema na internet onde possam ser catalogados diversos “botecos”.

Os objetivos específicos do trabalho são:

- a) desenvolver um subsistema de notícias no formato *Blog*;
- b) utilizar alguns padrões de usabilidade propostas pelo grupo Nielsen Norman, sobre usabilidade na Web;
- c) integrar o sistema com ferramentas já existentes na internet de mapas (Ex: Google Earth, Google Maps, Yahoo Maps, Map Quest, etc) para visualmente identificar a localidade.

1.2 ESTRUTURA DO TRABALHO

O presente trabalho está organizado em quatro capítulos, dispostos conforme a seguinte estruturação:

O primeiro capítulo apresenta a introdução, onde há uma descrição geral do trabalho contextualizando a escolha do tema e seus objetivos.

O segundo capítulo discute a fundamentação teoria do trabalho, definindo alguns conceitos utilizados posteriormente no trabalho.

Já o terceiro capítulo aborda o desenvolvimento do sistema, os requisitos necessários para sua especificação e implementação.

O quarto e último capítulo trata das considerações finais acerca do sistema.

2 FUNDAMENTAÇÃO TEÓRICA

Na fundamentação teoria é apresentado um pouco da história que fez este trabalho tomar os rumos que tomou, os principais conceitos que abrangem em sua totalidade o sistema e os trabalhos correlatos que foram fonte de inspiração e pesquisa.

2.1 CENÁRIO ATUAL

Em pesquisa realizada pela equipe da empresa Zinabre Ponto Com (ZINABRE, 2008) em Balneário Camboriú nos 6 principais “botecos” da cidade, constatou-se grande falta de verba para a divulgação dos ambientes. Em entrevistas realizadas com os proprietários dos estabelecimentos, estes, revelaram não dispor de quase nenhuma verba para tal e que a quase pouca mídia era principalmente feita via conversa popular. Mostraram também grande interesse em ter o seu estabelecimento cadastrado e comentado em um site da internet.

2.2 CONCEITOS

A seguir alguns conceitos que foram utilizados na confecção deste trabalho e também sobre as técnicas que foram estudadas e aplicadas neste trabalho.

2.2.1 PHP

O PHP (*Hypertext Preprocessor*) é uma linguagem de programação livre. “(..) linguagem de script embutida no HTML. Muito da sua sintaxe é herdada de C, Java e Perl com algumas características específicas do PHP juntas. O objetivo da linguagem é permitir que desenvolvedores Web escrevam páginas geradas dinamicamente rápido” (PHP, 2007).

Um dos bancos de dados *open-source* mais famosos e populares do mundo. Com mais de 100 milhões de cópias baixadas ou distribuídas, o MySQL, é utilizado em grande parte de

aplicação voltadas a Web, além de *datawarehouses*, aplicações comerciais, entre outros (MYSQL, 2008). Encontra-se atualmente na versão 5.0, esta sendo a mais estável e completa até agora.

2.2.2 W3C

Segundo o próprio site, O *World Wide Web Consortium*, ou apenas W3C, desenvolve tecnologias interoperáveis (especificações, guias, softwares e ferramentas) para elevar a Web em seu total potencial (W3C, 2007).

A maioria das páginas de internet são escritas em linguagens computacionais (como o HTML), que permite aos autores adicionar conteúdo multimídia e especificar como a página irá se apresentar, seu estilo, etc. Todas as linguagens tem a sua própria gramática, seu vocabulário e sua sintaxe e todos os documentos escritos com essas linguagens computacionais supostamente devem seguir estas regras (W3C, 2007).

Disponibiliza também em seu site uma ferramenta de validação de markup, em outras palavras, uma ferramenta de verificação da sintaxe dos documentos da Web.

Segundo FERRAZ (2003), “Construir um sistema com os Padrões Web significa tomar as providências necessárias para que cada documento no sistema conforme-se às especificações do formato em que é servido e a especificações adicionais de auxílio ao uso do mesmo”.

“Esses padrões hoje são estudados e felizmente os desenvolvedores estão aplicando em seus sistemas. Os desenvolvedores devem perceberem as incríveis vantagens que o desenvolvimento com os Padrões oferece, não só para a execução do trabalho, mas para a estruturação da Web em si. Para a estruturação da Web do futuro, onde ninguém terá que garimpar em buscadores para conseguir a informação que se precisa, mas a informação estará aonde você estiver, andará com você aonde quer que for, e você terá acesso a ela sem barreiras, na hora que quiser, onde quiser, e usando o dispositivo que for.” (TABLELESS, 2007).

2.2.3 AJAX

Técnica de programação definida como *Asynchronous JavaScript and XML*, ou simplesmente AJAX. Técnica esta que envolve a linguagem *JavaScript* e HTML / XHTML para que o sistema se comunique diretamente com o servidor usando o objeto *JavaScript XMLHttpRequest* para trocar informações sem recarregar a página (W3C, 2007).

Segundo Murray (2006), “Usando a tecnologia *JavaScript*, uma página HTML pode fazer chamadas assíncronas ao servidor que pode buscar conteúdo formatado em documentos XML. Ajax não é novo. Estas técnicas estão disponíveis aos desenvolvedores na plataforma Windows por vários anos. O que mudou recentemente foi a inclusão ao suporte para o objeto *XMLHttpRequest* na *runtime*³ do *JavaScript* na maioria dos navegadores”.

Ainda segundo Murray, “A página interage com a tecnologia *JavaScript* baseada em eventos como, o carregar de um documento, um clique do mouse, mudanças de foco em objetos ou até um *timer* (cronômetro). Interações em Ajax permitem uma clara separação entre a apresentação lógica e os dados”.

Alguns dos principais usos das interações do Ajax são:

- a) validação de formulário em tempo real. Formulários de dados, números seriais, códigos postais ou até códigos especiais de cupons que necessitam de uma validação no lado do servidor;
- b) auto-completação. Uma específica porção dos dados do formulário como, email, nome e nome da cidade podem ser auto completados enquanto o usuário está digitando;
- c) carregamento de conteúdo conforme necessidade (baseado em eventos do cliente);
- d) efeitos e controles de interface com o usuário sofisticadas. Controles do tipo árvore, tabelas de dados, editores de textos, calendários e barras de progresso permitem uma melhor interação com o usuário;
- e) atualização do conteúdo da página. Páginas HTML podem buscar dados do servidor para atualizar “dados em tempo real”, como resultados, ações da bolsa, tempo, etc;
- f) envio parcial de informação ao servidor. A página HTML pode enviar somente os dados necessário sem precisar recarregar toda a página.

2.2.4 GOOGLE MAPS API

Segundo Google (2008), “O Google Maps é um serviço do Google que oferece uma poderosa tecnologia de mapas amigável e informações sobre empresas locais, incluindo o endereço das empresas, informações de contato e direções de tráfego”.

³ Tempo de execução (runtime), é o período em que um programa de computador permanece em execução.

Ainda segundo o Google (2008), “A API do Google Maps permite usar *JavaScript* para incorporar o Google Maps em sua página da web. A API fornece diversos utilitários para manipular mapas (como na página <http://maps.google.com>) e adicionar conteúdo ao mapa através de diversos serviços”.

Os serviços do Google Maps estão disponíveis para qualquer site que seja gratuito para seus consumidores. Para usar de suas ferramentas é somente necessário se inscrever para uma chave que pode ser utilizada para um único “diretório” do seu servidor.

A documentação necessária para utilizar-se dos serviços está disponível no Google Code (GOOGLE, 2008).

2.2.5 USABILIDADE NA WEB

A usabilidade é um pontos fundamentais do sucesso de sistemas web atualmente, pois segundo NIELSEN (2007, p. 16), “A usabilidade é um atributo de qualidade relacionado à facilidade de uso de algo. (...) refere-se à rapidez com que os usuários podem aprender a usar alguma coisa, a eficiência deles ao usá-la, o quanto lembram daquilo, seu grau de propensão a erros e o quanto gostam de utilizá-la”. Isto é um dos fatores que levam este sistema a usar das lições apreendidas nos 716 Websites pesquisados pelo grupo com 2.163 pessoas de diversos países.

Em seu livro, ele também explica algumas regras básicas a serem utilizados ainda na confecção do *layout* de um website, para que este tenha sucesso, como por exemplo:

- a) as pessoas estão acostumada a encontrar a logomarca da empresa no canto superior esquerdo em um website, e ao clicá-la, esta deve levá-las à capa do website;
- b) o menu deve ficar na lateral esquerda do website, em baixo da logomarca, este é o lugar onde as pessoas primeiro procuram pelo menu;
- c) se houver um campo de busca, este deve funcionar, e deve ficar no canto superior direito da página.
- d) procurar usar cores e fontes legíveis, na dúvida optar por letras pretas sobre o fundo branco, para facilitar a leitura;
- e) em uma página com navegação por categorias e subcategorias, mostre para o usuário em que categoria ele se encontra e dê-lhe links para voltar para a categoria mãe;
- f) utilizar nomes de páginas que possam ser facilmente lembradas e identificadas do

conteúdo que dispõem. Esta usabilidade também é conhecida como *Clean Url*.

2.2.6 WEBML

Segundo visão dos idealizadores, (CERI, 2008) “A WebML provê de uma maneira gráfica, ainda que formal, especificações embutidas num processo de design completo, que pode ter a ajuda de ferramentas de design visual”. Ainda segundo o autor, ele afirma que os principais objetivos do processo de design da WebML são:

- a) expressar a estrutura de uma WebApp com um descritivo de alto nível, que pode ser usado para consultas (Banco de Dados), evolução e futuras modificações;
- b) prover várias visões de um mesmo conteúdo;
- c) separar o conteúdo da informação e sua composição em páginas, navegação e apresentação, que podem ser definidas e evoluídas independentemente;
- d) armazenar *meta-information* coletada durante o processo de design em um repositório, que pode ser usado durante o tempo de vida de uma aplicação para gerar páginas dinâmicas para a web;
- e) modelar usuários e comunidades explicitamente no repositório, para permitir a especificação das políticas de personalização e aplicações *one-to-one*;
- f) habilitar a especificação de operações de manipulação de dados para atualizar o conteúdo do site ou interagir com serviços externos arbitrários.

O processo de desenvolvimento de uma WebApp com WebML, (CERI, 2008) pode passar por três diferentes modelos, são eles:

- a) modelo de dados;
- b) modelo de hipertexto;
- c) modelo de apresentação.

2.2.6.1 MODELO DE DADOS

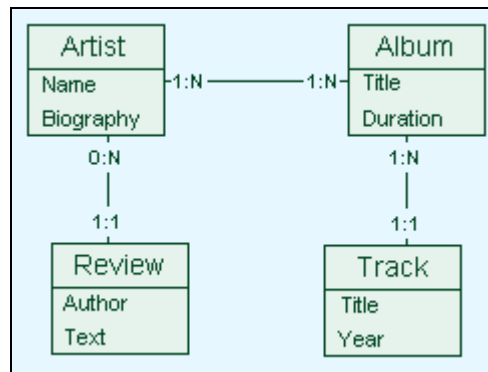
O modelo de dados da WebML foi adaptado dos modelos conceituais para design de dados, como já usado em outras linguagens de modelagem de dados para aplicativos, usados na engenharia de software. É compatível com os modelos Entidade-Relacionamento (E/R)

usado nos modelos de banco de dados e com o diagrama de classes também da UML usado para modelagem orientada a objetos (ODMG).

Fazem parte dois elementos principais no modelo: as **entidades** e os **relacionamentos**.

- a) entidade: agrupamento de dados com características em comum;
- b) relacionamento: representa o significado da ligação entre duas ou mais entidades.

Um exemplo de um modelo de dados pode ser vista na Figura 1.



Fonte: Overview of WebML – disponível em:
<http://webml.elet.polimi.it/webml/page1.do>.

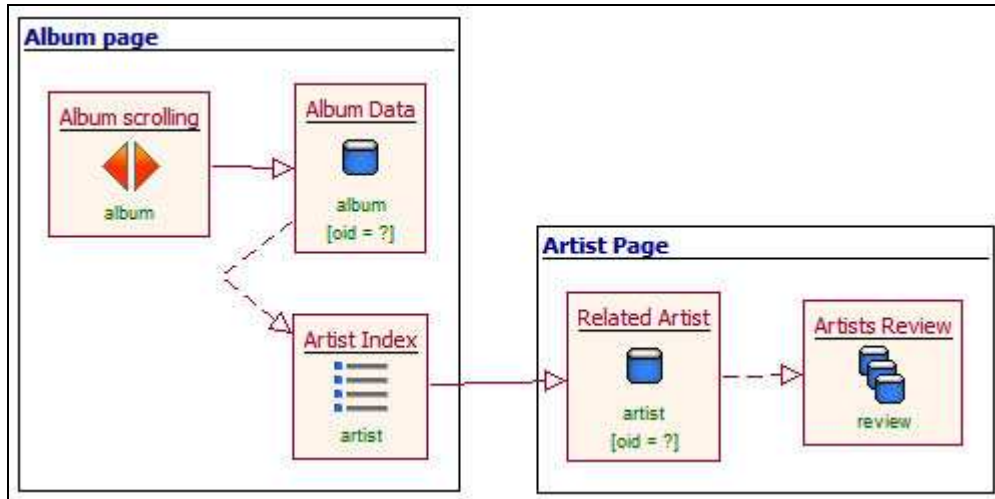
Figura 1 – Exemplo de um modelo de dados da WebML.

2.2.6.2 MODELO DE HIPERLINKS

O modelo de hipertexto especifica a **composição** e a **navegação** da WebApp.

- a) composição: descreve quais **Páginas** compõem o hipertexto, e quais os **Conteúdos** (chamados Unidades ou *Unit*) compõem cada página. *Units* são conteúdos utilizados para mostrar a informação descrita no modelo de dados. Sete tipo de *Units* são pré-definidos na WebML, são eles: *data*, *multi-data*, *index*, *entry* e *scroller*. Cada *Unit* é associada a uma entidade de onde seu conteúdo é retirado.
- b) navegação: especificada através de **Links**. Links podem ser definidos entre *Units* dentro de uma página, entre *Units* colocados em diferentes páginas e entre páginas. A informação carregada junto com um link é chamada de Contexto de Navegação (*Navigation Context*) ou simplesmente Contexto. Links que carregam informação são chamados de Links Contextuais onde os links que não carregam informação são chamados de Links Não-Contextuais.

Um exemplo de um modelo de hiperlinks pode ser vista na Figura 2. Esta figura foi refeita pela fonte não possuir qualidade suficiente.



Fonte: Overview of WebML – disponível em:
<http://webml.elet.polimi.it/webml/page1.do>.

Figura 2 – Exemplo de um modelo de hiperlinks da WebML.

2.2.6.3 MODELO DE APRESENTAÇÃO

WebML não inclui um modelo específico para expressar a apresentação, em nível conceitual. As especificações da WebML podem ser representadas usando XML, a apresentação é considerada como uma transformação de documento, mapeando as especificações da WebML da página em uma página escrita em uma implementação de linguagem concreta como JSP (*JavaServer Pages Technology*) ou ASP.NET (sucessor do ASP, *Active Server Pages*).

Desta maneira, a apresentação é feita através de modelos de estilo XSL (*Extensible Style Sheet Language*) incluídos nas Páginas, *Units* e Contextos. O XSL usa como entrada as especificações da WebML, codificadas em documentos XML conforme a “*WebML Document Type Definition*” e retorna como saída *templates* (modelos) de páginas com os códigos requeridos embutidos.

Segundo Locatelli (2003), “A WebML utiliza o XSL por utilizar a sintaxe do XML e por manipular objetos gráficos (“*flow objects*”) e não elementos. Mais especificamente transforma fragmentos de um documento XML em objetos gráficos”.

A seguir pode-se encontrar algum dos softwares utilizados para a confecção do

trabalho.

2.2.7 WEBRATIO

A WebML sugere como ferramenta CASE para a modelagem do sistema e até para o desenvolvimento de WebApps inteiras a partir dos modelos gerados, o software chamado WebRatio.

Software que suporta o desenvolvimento de WebApps desde a análise de seus requerimentos até a sua implantação. Ferramenta visual de design que oferece:

- a) codificação dos requerimentos da WebApp;
- b) modelagem dos objetos de *back-end*;
- c) modelagem das interfaces de *front-end*;
- d) possibilidade de usar objetos persistentes em uma fonte de dados já existente ou construir o banco de dados a partir do modelo de objetos;
- e) definir a aparência do *front-end* com a importação de layouts e estilos desenvolvidos por designers gráficos;
- f) gera o código-fonte completo em J2EE da WebApp e instala em qualquer plataforma pré-definida;
- g) gera a documentação automaticamente.

A WebApp gerada pelo WebRatio segue as regras de API's e *FrameWorks*⁴ como *Java Servlet*⁵, JSP e Jakarta Struts⁶. Também não utiliza nenhuma *runtime* proprietária e o código gerado é J2EE e ISO/ANSI SQL⁷. A versão atual do WebRatio é a 5.0.0 e continua sendo livre para fins não comerciais.

Uma tela do software pode ser visto a seguir na Figura 3.

⁴ Estrutura de suporte definida em que um outro projeto de software pode ser organizado e desenvolvido. Um *framework* pode incluir programas de suporte, bibliotecas de código, linguagens de script e outros softwares.

⁵ Servlet é uma tecnologia que insere novos recursos a um servidor, são extensões de servidores. Disponibiliza ao programador da linguagem Java uma interface para o servidor web, através de uma API. As aplicações baseadas no Servlet geram conteúdo dinâmico.

⁶ O Projeto Jakarta criou e mantém vários softwares livres para a plataforma Java, sendo Struts um *frameworks* para desenvolvimento de aplicações web.

⁷ Padronização da linguagem SQL de acesso à bancos de dados.

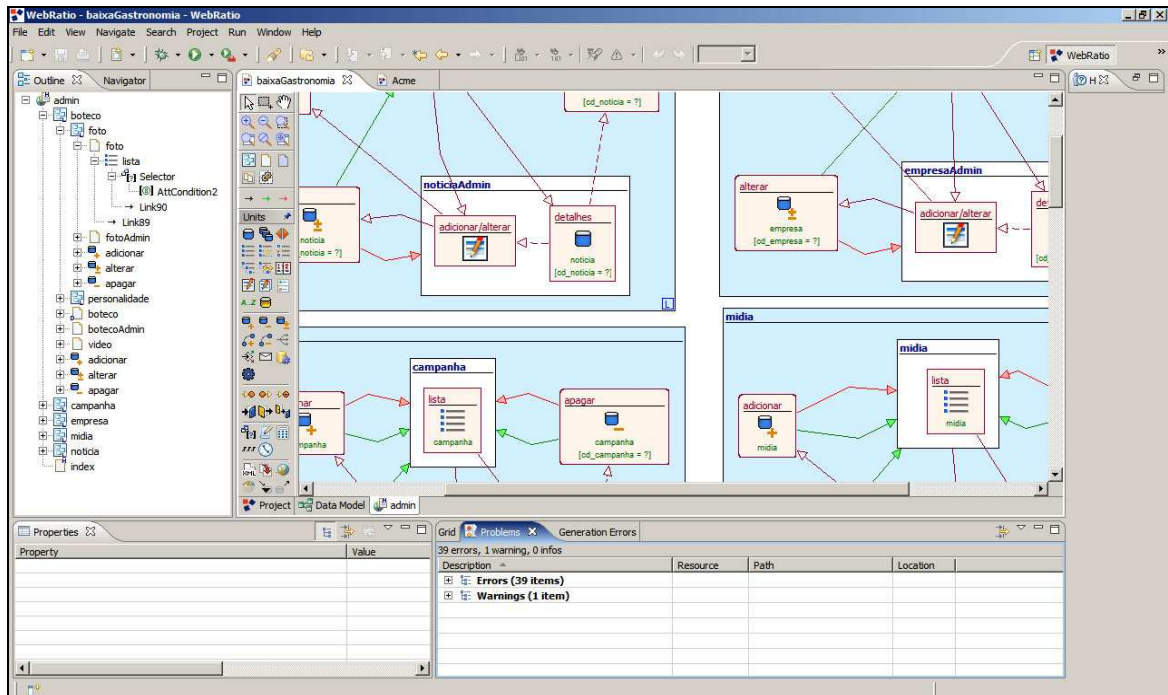


Figura 3 – Tela da ferramenta CASE WebRatio versão 5.0.0.

2.3 TRABALHOS CORRELATOS

O impresso do GUIA QUATRO RODAS (2007), o único que seleciona e classifica hotéis, restaurantes e atrações no Brasil. É uma das principais fonte de inspiração para este trabalho. Conforme citado anteriormente, é lançado anualmente com os principais estabelecimentos (em suma maioria patrocinadamente e de alto nível social), hotéis para se hospedar com diferentes taxas, melhores restaurantes e eventos que ocorrem pelo Brasil inteiro.

Segundo o próprio Guia, faz-se uma “seleção dentro dos critérios de avaliação do Guia” GUIA QUATRO RODAS (2007, p.18) de como são escolhidos os hotéis, restaurantes e atrações que serão publicados, ou seja, nem todos os aspectos da região são abordados, apenas os principais e de maior fluxo.

Também foram analisados dois trabalhos, o primeiro de Pós-Graduação intitulado “Engenharia de Software para o desenvolvimento de WebApps e as metodologias OOHDM e WebML” para se obter base teórica para a modelagem do sistema (LOCATELLI, 2003).

O segundo trabalho, de título “Sistema de informações executivas baseado em data warehouse aplicado a cooperativa central de crédito” (GRACIK, 2005), da qual foi utilizado o

WAE (Web Application Extension) da UML que define uma série de estereótipos aplicáveis na modelagem de sistemas web.

3 DESENVOLVIMENTO DO SISTEMA

Neste capítulo é abordado o desenvolvimento do sistema como um todo. Apresentando a análise de requisitos, a especificação do sistema e a sua implementação. Também mostra um pouco do código utilizado nas principais funções do sistema e finaliza mostrando a visão ambos do usuário final tanto do administrador para com o sistema.

3.1 REQUISITOS DO SISTEMA

O Quadro 1 **Erro! Fonte de referência não encontrada.** apresenta os requisitos funcionais previstos para o sistema e sua rastreabilidade, ou seja, vinculação com os casos de uso associados.

REQUISITOS FUNCIONAIS
RF01: O sistema deverá permitir o cadastro de “botecos”
RF02: O sistema deverá permitir o cadastro de fotos para os “botecos”
RF03: O sistema deverá permitir o cadastro de vídeos para os “botecos”
RF04: O sistema deverá permitir o cadastro de personalidades para os “botecos”
RF05: O sistema deverá permitir o cadastro de notícias em formato Blog
RF06: O sistema deverá permitir o cadastro de receitas
RF07: O sistema deverá permitir o cadastro de banners de mídia no sistema
RF08: O sistema deverá permitir que os usuários, ao entrarem em contato pelo formulário do site, sejam cadastros para receber notícias sobre atualizações (e-mail marketing)
RF09: O sistema deverá permitir que o usuário faça buscas detalhadas dos “botecos”
RF10: O sistema deverá emitir um relatório de visualização dos banners integrado ao Google Analytics
RF11: O sistema deverá emitir um relatório de visualização dos usuários cadastrados para receber notícias segmentado (cidade, estado, etc)

Quadro 1 - Requisitos funcionais

O Quadro 2 lista os requisitos não funcionais previstos para o sistema.

REQUISITOS NÃO FUNCIONAIS
RNF01: O sistema deverá ser desenvolvido na linguagem PHP, formatação CSS e estruturação Tableless ⁸
RNF02: O sistema deve utilizar o banco de dados MySql utilizando tabelas do tipo InnoDB para ser feito o uso da integridade referencial.
RNF03: O sistema deverá ser desenvolvido nos padrões Web definidos pelo W3C
RNF04: O sistema deverá ser desenvolvido, desde o layout até sua implementação, seguindo inicialmente as regras 'a', 'b', 'd' e 'f' descritas anteriormente no texto, definidas pelo grupo Nielsen Norman sobre usabilidade na Web.
RNF05: Para maior segurança o acesso de usuários à parte administrativa do sistema será controlado via servidor e não via aplicação.

Quadro 2 - Requisitos não funcionais

O Quadro 3 lista as regras de negócio envolvidas no sistema.

REGRAS DE NEGÓCIO
RN01: As mídias devem ser separadas por campanha e por empresa. Sendo que cada empresa poderá ter várias campanhas, e várias mídias poderão ser adicionadas à cada campanha.
RN02: Os botecos terão categorias pelas quais serão classificados segundo critérios estabelecidos pelos administradores. Estas serão qualidades alcançadas pelos botecos ou mesmo diferenciais do estabelecimento (Ex: Excelente Atendimento, Mesa de Sinuca, etc)

Quadro 3 - Regras de negócio

3.2 ESPECIFICAÇÃO

Após levantamento dos requisitos necessários para o sistema foi desenvolvida a especificação, através da geração dos diagramas da WebML, sendo o Modelo de Dados e também o modelo de HiperLinks com a ferramenta WebRatio. Os diagramas desenvolvidos serão apresentados a seguir.

3.2.1 MODELO DE DADOS

O modelo de dados, conforme especificação da WebML, está ilustrado a seguir pela Figura 4, Figura 5 e Figura 6. Os códigos utilizados para gerar as tabelas podem ser vistos no APÊNDICE A.

⁸ TABLELESS (2007), método para confecção de sites usando Padrões Web (*Web Standards*) como guia. Não é utilizado tabelas, usa-se exclusivamente o CSS.

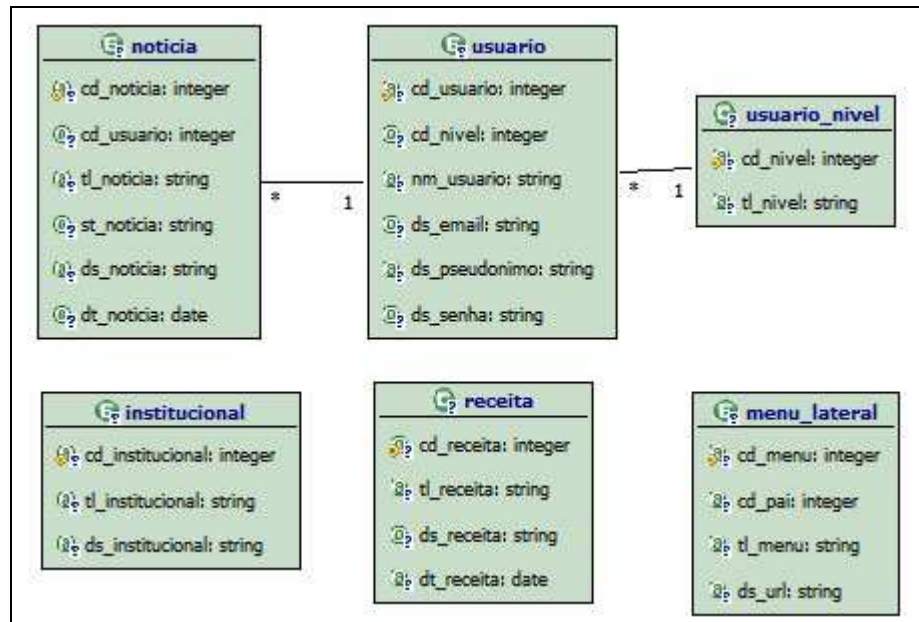


Figura 4 - Modelo de Dados (a)

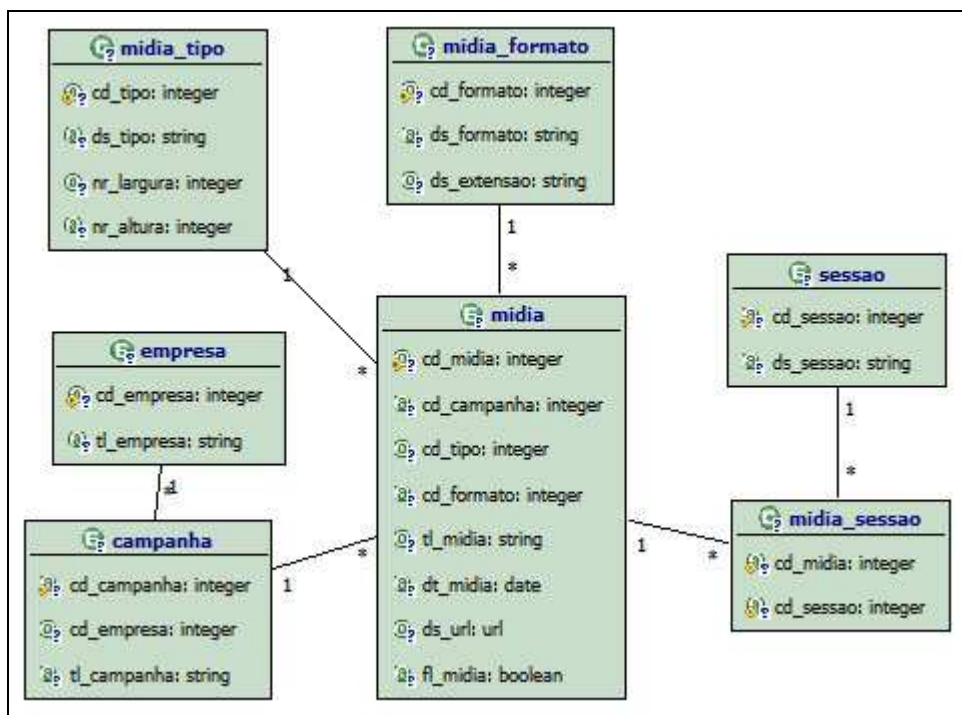


Figura 5 - Modelo de Dados (b)

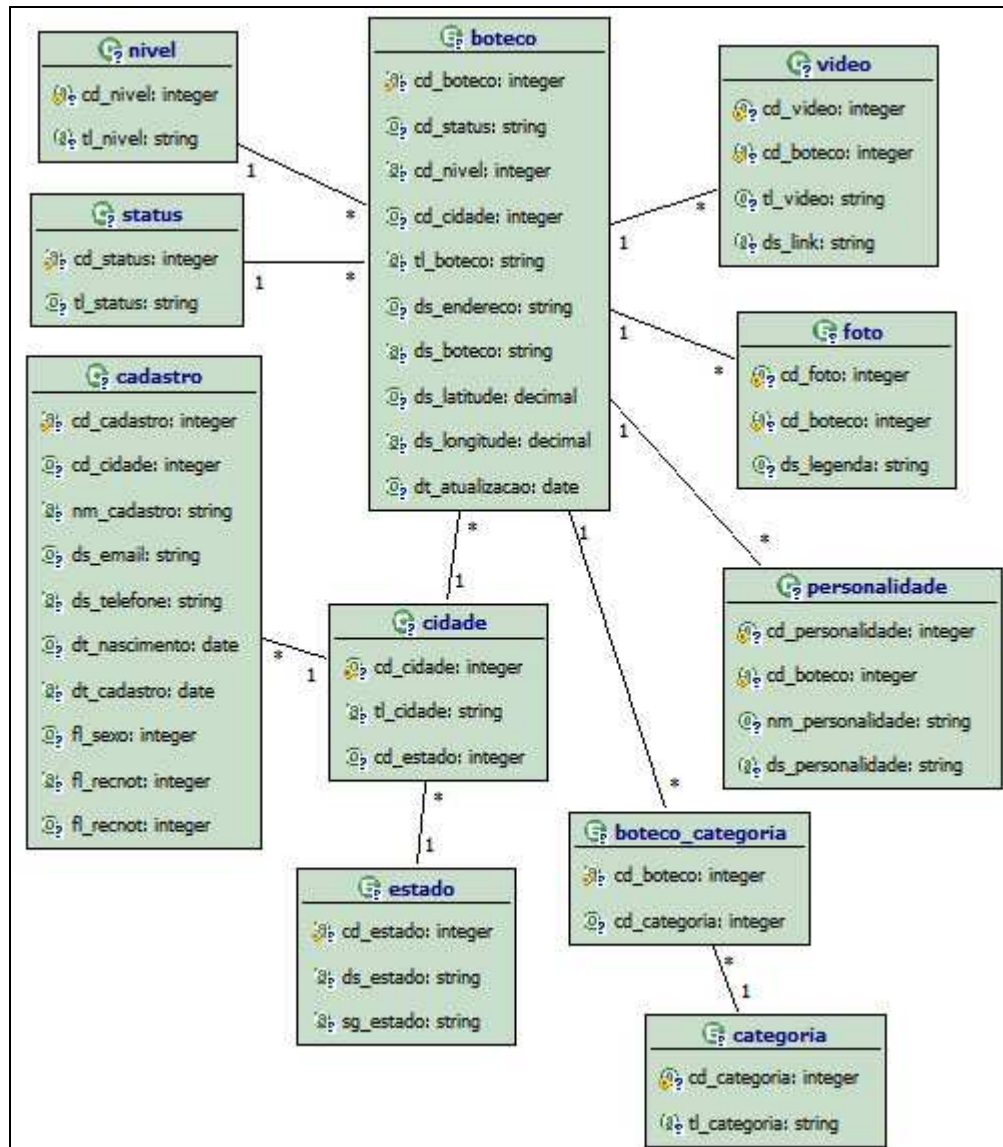


Figura 6 - Modelo de Dados (c)

3.2.2 MODELO DE HIPERLINKS

Nesta seção são listados todos os modelos de hiperlinks que demonstram visualmente o funcionamento dos requisitos no que tange à interação do usuário com o sistema.

A Figura 7 tem o modelo que demonstra o requisito RF01 – Cadastra Boteco. Esta figura foi separada e faz junção com a Figura 8, a Figura 9 e a Figura 10. Para este requisito são utilizadas duas páginas, “boteco” e “botecoAdmin”. A primeira com a listagem de todos os “botecos” cadastrados e os links para: edição do registro; listagem de fotos do boteco;

listagem de vídeos do boteco; listagem de personalidades para o “boteco”. Há também um link para fazer a inclusão de um novo “boteco”. Nota-se que tanto o link de inclusão quanto o link de alteração dos registros levam à página “botecoAdmin” onde os dados são incluídos, alterados ou excluídos. Nota-se que na página “botecoAdmin” há a seleção da cidade, estado, status, nível e as categorias.

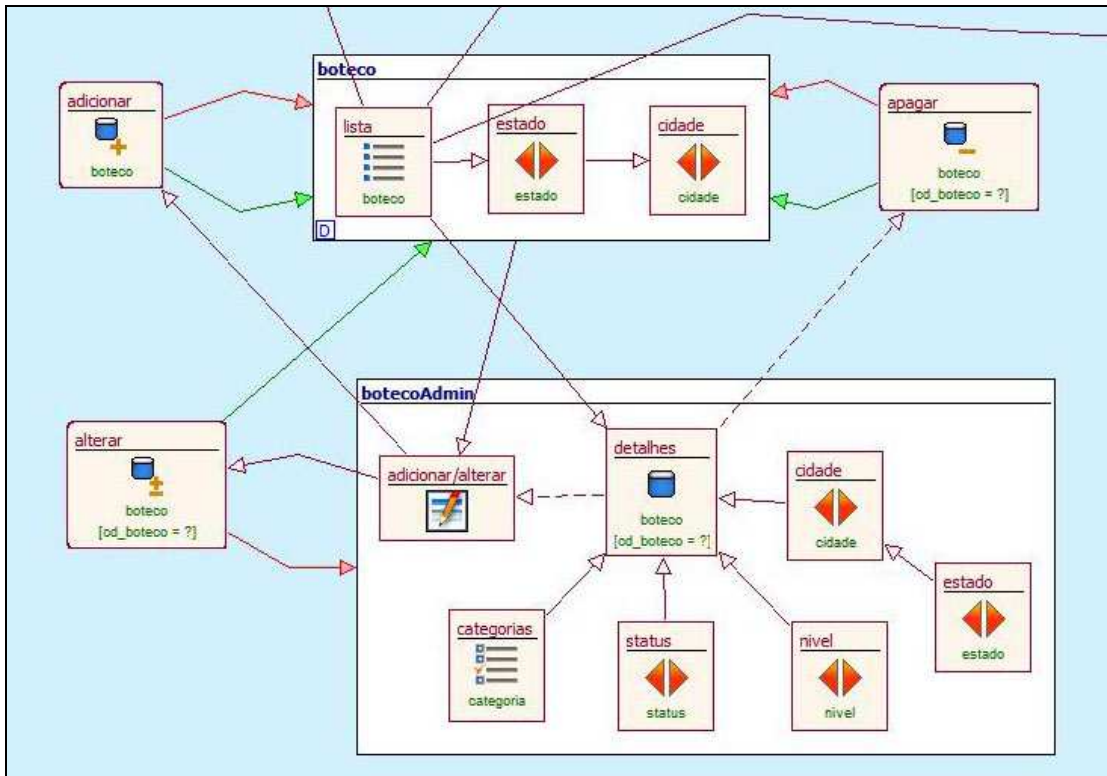


Figura 7 – Modelo de Hiperlinks do Requisito RF01 (Cadastra boteco)

A Figura 8 tem o modelo que demonstra o requisito RF02 – Cadastrar fotos para os botecos. Esta página é acessada através da página “boteco”, pois deve passar o código do boteco para o qual o administrador estará incluindo fotos. A página “foto” possui uma listagem de todas as fotos já cadastradas, a opção de alterar alguma foto previamente cadastrada e o link para incluir nova foto, ambas levando à página “fotoAdmin”.

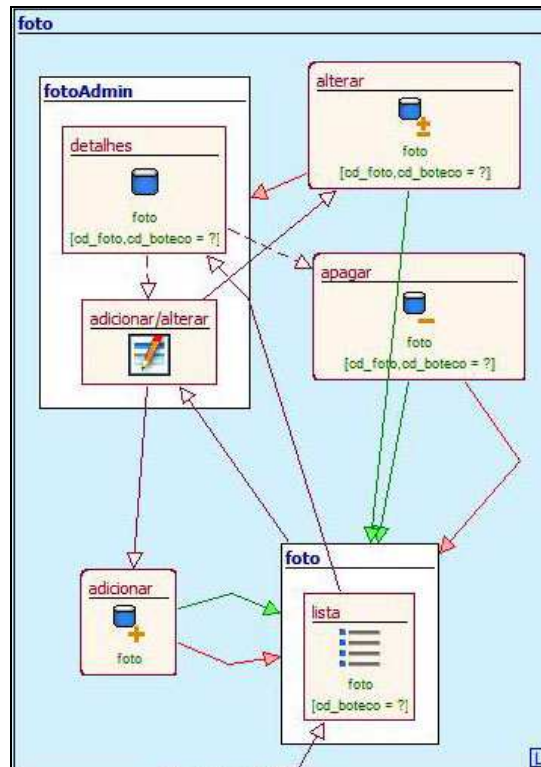


Figura 8 - Modelo de Hiperlinks do requisito RF02 (Cadastra foto para boteco)

A Figura 9 tem o modelo que demonstra o requisito RF03 – Cadastrar vídeos para os botecos. Da mesma forma que a inclusão de fotos, a inclusão de vídeos é acessada através da página “boteco” onde é passado o código do boteco para a inclusão do vídeo. Duas páginas fazem parte, “vídeo” e “videoAdmin”. Na primeira há a listagem dos vídeos já cadastrados e na segunda as funções de inclusão, alteração ou exclusão do registro.

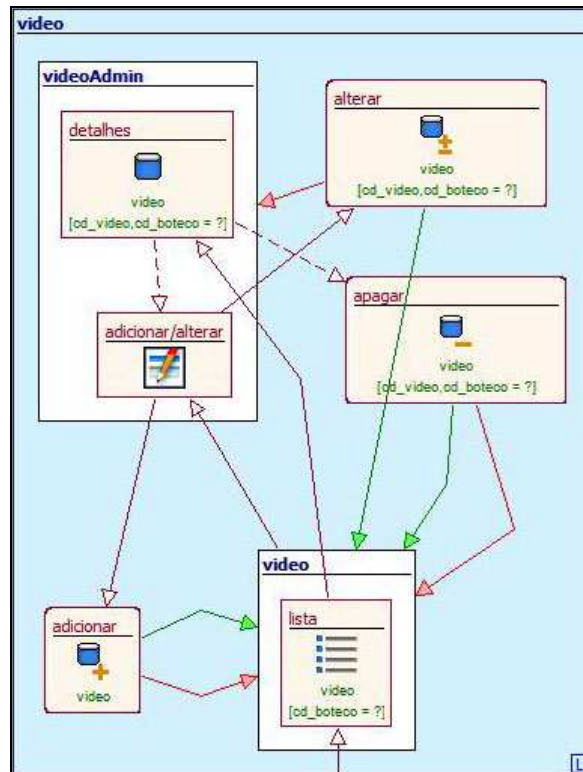


Figura 9 - Modelo de Hiperlinks do requisito RF03 (Cadastra vídeos para o boteco)

A Figura 10 tem o modelo que demonstra o requisito RF04 – Cadastrar personalidades para os botecos. Acessada também através da página “boteco” com a passagem do código do boteco. Possui duas páginas “personalidade” e “personalidadeAdmin”. A primeira com a listagem das personalidades e a segunda com as funções de inclusão, alteração e exclusão do registro.

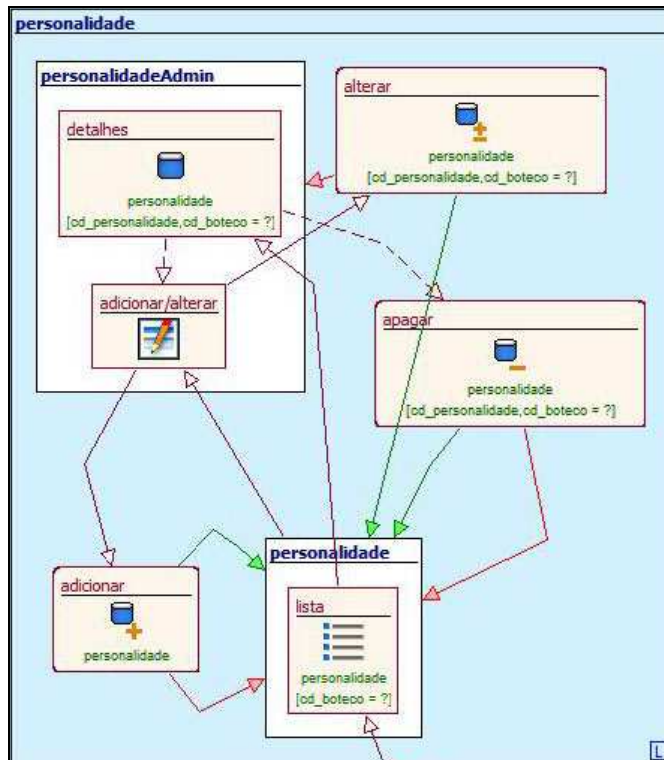


Figura 10 - Modelo de Hiperlinks do requisito RF04 (Cadastra personalidades para o boteco)

A Figura 11 e a Figura 12 fazem referência ao requisito RF05 – Cadastrar notícias no formato Blog. Na página “noticia” são mostradas todas as notícias previamente cadastradas. Os links tanto de inclusão, como de alteração da notícia também estão nesta página. Nota-se que na página “noticiaAdmin” há a seleção do usuário que está postando a notícia.

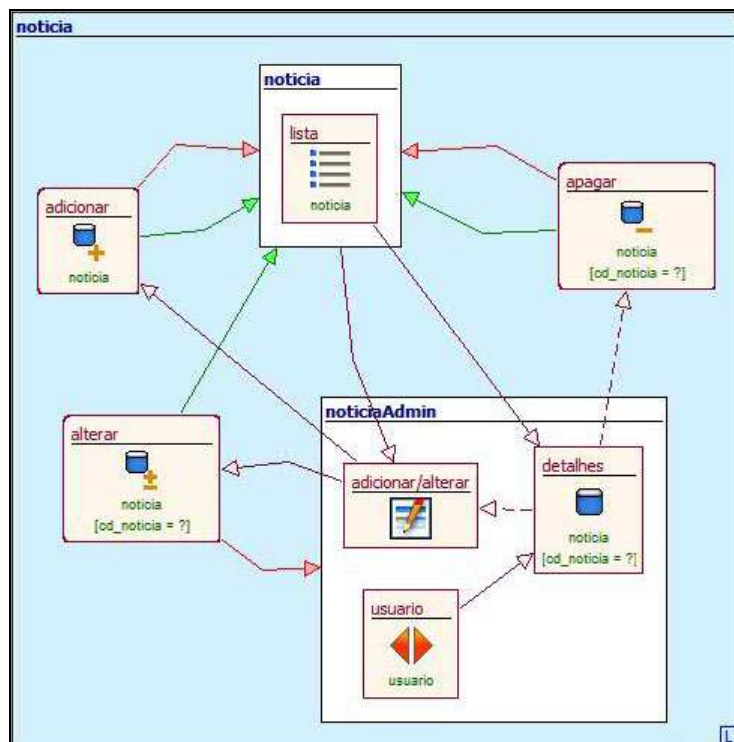


Figura 11 - Modelo de Hiperlinks do requisito RF05 (Cadastra notícias)

A inclusão, alteração ou exclusão dos usuários das notícias é feita pela página “usuárioAdmin”, acessada a partir da listagem dos usuários previamente cadastrados na página “usuário”.

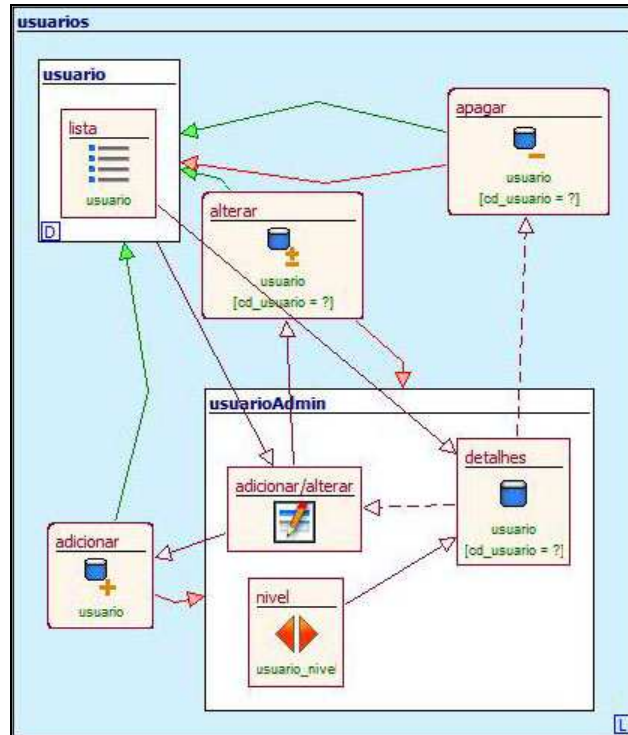


Figura 12 - Modelo de Hiperlinks do requisito RF05 (Cadastra usuário para a notícias)

A Figura 13 demonstra o requisito RF06 – Cadastro de receitas. A página “receita” obtém a listagem das receitas cadastradas e página “receitaAdmin” que por sua vez faz a inclusão, alteração e exclusão das receitas.

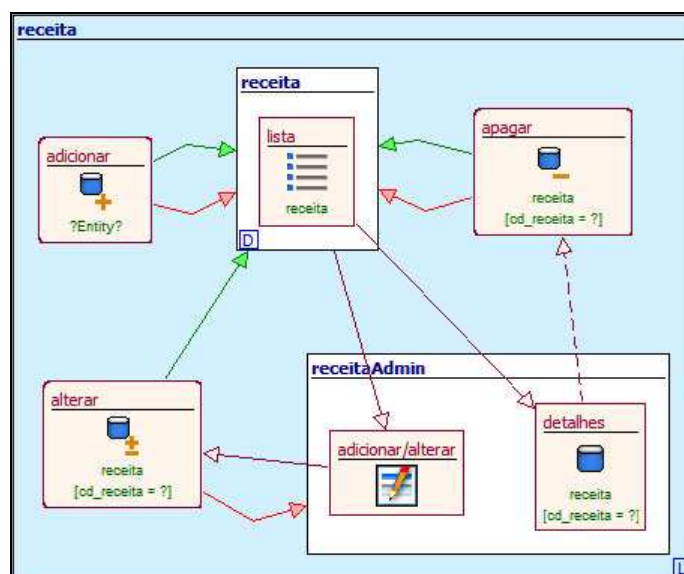


Figura 13 - Modelo de Hiperlinks do requisito RF06 (Cadastra receitas)

A Figura 14 ilustra o requisito RF07 – Cadastro de mídias de banners. Logo abaixo

estão a Figura 15 e a Figura 16 que acompanha a Figura 14 na RN01 – Cadastro de empresas e de campanhas para cada empresa.

A Figura 14 é composta por duas páginas, “mídia” e “mídiaAdmin”. A primeira com a relação de todas as mídias cadastradas e a segunda com as funções de alteração e exclusão dos registros. Na página “mídiaAdmin” há a seleção do formato tipo, da empresa, campanha da mídia.

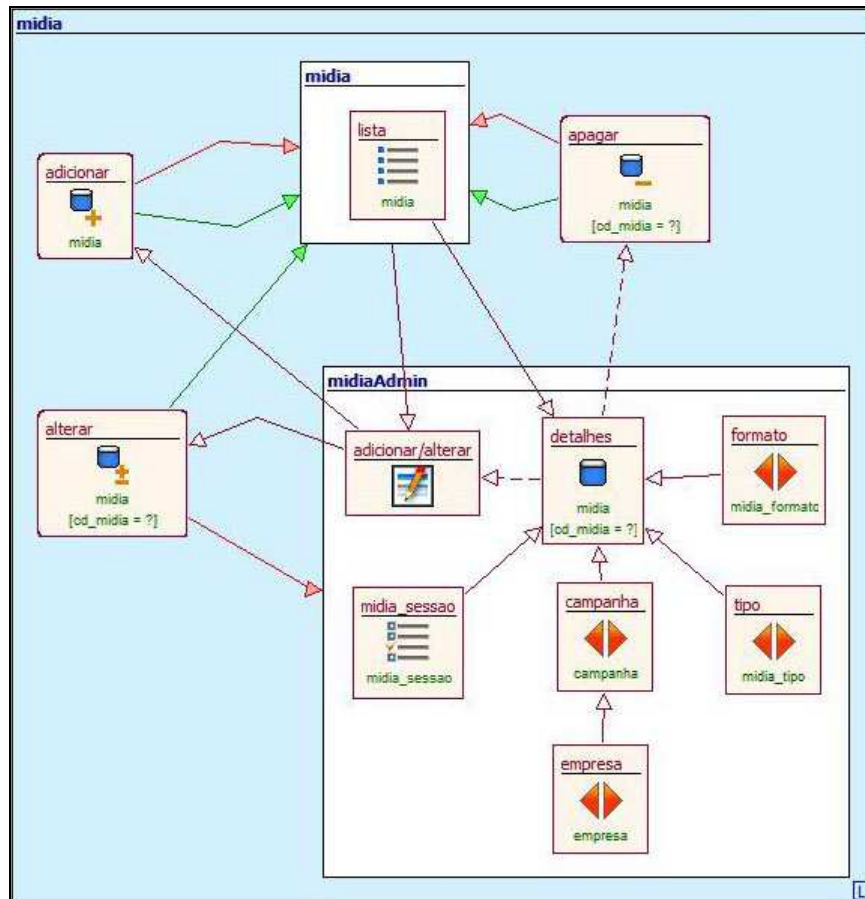


Figura 14 - Modelo de Hiperlinks do requisito RF07 (Cadastro de mídias)

A Figura 15 é acessada através da página “empresa” pois necessita o código da empresa para poder listar as campanhas. Também é composta por duas páginas “campanha” e “campanhaAdmin”, onde a primeira é responsável pela listagem das campanhas já cadastradas para a empresa em questão. A segunda página contém as funções para alteração e exclusão do registro.

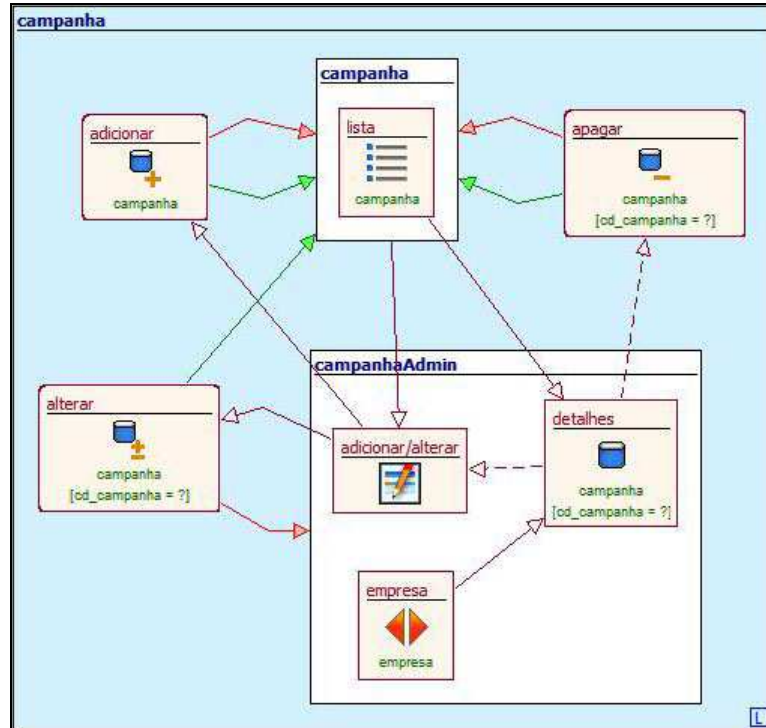


Figura 15 - Modelo de Hiperlinks da RN01 (Cadastro de campanha para mídia)

A Figura 16 com as páginas “empresa” e “empresaAdmin” mantém os cadastros de empresas para as mídias. A primeira página contém a listagem das empresas e a segunda página contém as funções de alteração e exclusão das empresas.

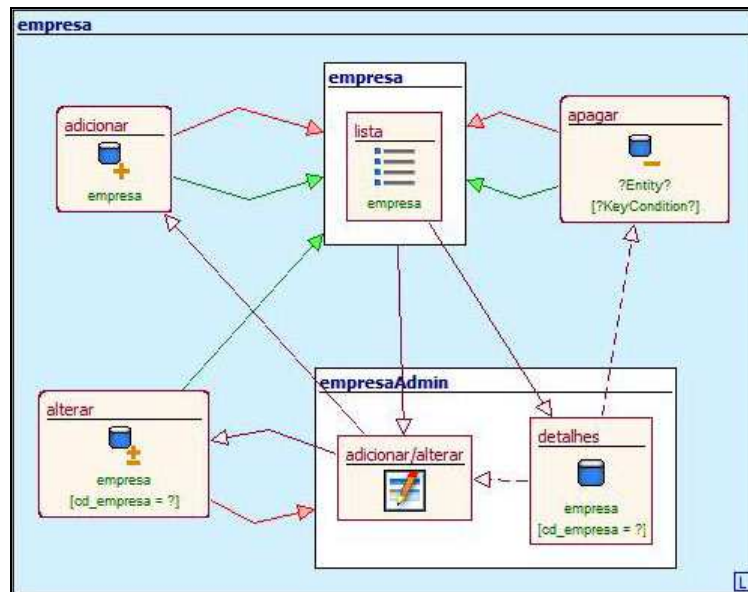


Figura 16 - Modelo de Hiperlinks da RN01 (Cadastro de empresa para a mídia)

3.3 IMPLEMENTAÇÃO

Este tópico versa sobre a fase de implementação do sistema especificado no tópico anterior, descrevendo as técnicas e ferramentas utilizadas para tanto.

3.3.1 FERRAMENTAS UTILIZADAS

O sistema foi desenvolvido em linguagem de programação PHP, o ambiente utilizado foi o Adobe Dreamweaver CS3 Trial. O Dreamweaver foi utilizado para desenvolver todas as camadas do sistema incluindo o design (HTML + CSS), a programação (arquivos PHP) e os códigos JavaScript. Também se fez uso do Adobe Flash CS3 para confecção dos banners de mídia. O servidor de testes utilizado foi o Apache (1.3.33) com o PHP instalado (4.3.10). O SGBD utilizado foi o MySQL (4.1.9), e para administrar o SGBD foi usado o PhpMyAdmin (2.6.1).

O WebRatio foi utilizado somente para a confecção das telas que fazem a modelagem do sistema. Todo o código do sistema foi implementado pelo autor e não gerado pela aplicação como possível.

O Tiny MCE (MOXIECODE, 2008) foi utilizado como editor HTML para o administrador poder alterar o conteúdo do sistema diretamente pela área administrativa. O TINY MCE é desenvolvido em JavaScript e é livre para uso. É possível personalizar o editor, assim, escondendo ou mostrando somente determinadas funções operadas por ele. A Figura 17 demonstra a aparência do editor dentro do sistema.

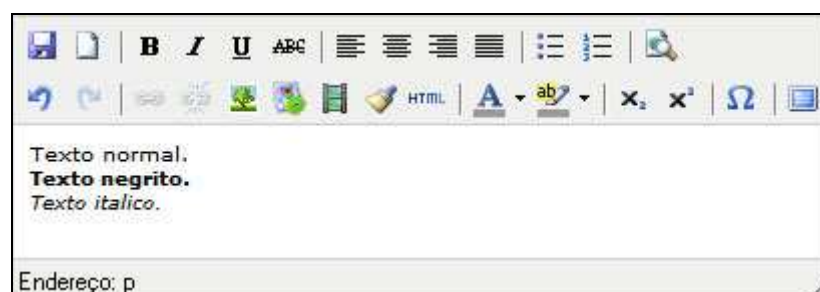


Figura 17 – Tiny MCE

3.3.2 ASPECTOS IMPORTANTES DA IMPLEMENTAÇÃO

Nesta seção será discutida um pouco de cada um dos diferentes aspectos e técnicas reunidas para elaborar este projeto.

3.3.2.1 QUESTÕES DE USABILIDADE

Foi retirado dos ensinamentos e estatísticas de Jakob Nielsen e seu grupo os principais itens navegacionais na concepção do layout do sistema. Alguns pontos relatados por ele que fizeram parte da implementação do sistema:

- a) logomarca no canto superior esquerdo;
- b) menu localizado abaixo da logomarca na lateral esquerda;
- c) cores contrastantes e que facilitam a leitura, letras pretas em fundo branco;
- d) disposição de conteúdo intuitivo e de fácil acesso;
- e) tipologia de fonte da web 2.0 (*Trebuchet MS*);
- f) *clean URL*.

A Figura 18 mostra a tela inicial do *Front* do sistema, onde é possível distinguir a posição da logomarca, menu, cores e a tipografia utilizada. Quesitos importantes segundo Jakob Nielsen.



Figura 18 - Usabilidade no sistema

O sistema também se utiliza do *ModRewrite* do servidor Apache para redirecionar internamente as páginas web. Esta funcionalidade elimina o uso de variáveis na URL. Além

de deixar as URLs fixas, melhora e facilita o indexamento das páginas do sistema por *web spiders*⁹ de sites de pesquisa como o Google.

O Quadro 4 demonstra uma URL normal com o parâmetro “cod” sendo passado com valor igual a “5”. Já na Quadro 5 pode-se ver o uso de *Clean URL*, o endereço demonstra exatamente qual é o seu conteúdo.

```
http://www.baixagastronomia.com/boteco.php?cod=5
```

Quadro 4 - URL normal com passagem de parâmetro

```
http://www.baixagastronomia.com/boteco/5-Rei-do-Lanche-O.html
```

Quadro 5 - Clean URL

No Quadro 6 está o código inserido no arquivo *.htaccess* na raiz do sistema. Arquivo este que é lido pelo *ModRewrite* do servidor para interpretar o redirecionamento interno das páginas.

```
RewriteEngine On

#Noticias
RewriteRule ^noticia/pagina/(.*)$ index.php?codPagina=$1 [L]
RewriteRule ^noticia/(.*)$ index.php?codNoticia=$1 [L]

#Institucional
RewriteRule ^quemsomos institucional.php?cod=1 [L]
RewriteRule ^oblog institucional.php?cod=2 [L]

#Botecos
RewriteRule ^botecopesquisa botecopesquisa.php [L]
RewriteRule ^boteco/(.*)$ boteco.php?urlBoteco=$1 [L]
RewriteRule ^botecomaps/(.*)$ botecomaps.php?urlBoteco=$1 [L]
RewriteRule ^botecomaps botecomaps.php [L]

#Receitas
RewriteRule ^receitas/(.*)$ receita.php?urlReceita=$1 [L]
RewriteRule ^receitas receita.php [L]

#Contato
RewriteRule ^contato contato.php [L]
```

Quadro 6 - Código do *ModRewrite*

3.3.2.2 AJAX

Foi utilizado dos conceitos de Ajax para implementar alguns recursos dentro do sistema, como é o caso dos comboBoxes de cidade-estado. O usuário ao selecionar o estado tem o comboBox de cidades preenchido com as cidades do estado selecionado.

No Quadro 7 está o código embutido no comboBox de estado que quando modificado

⁹ É uma aplicação de software que tem por objetivo varrer a web indexando as páginas pelas suas urls e links.

(*onChange*) ativa o processo de popular o *comboBox* de cidades.

```
<select name="estado" size="1" id="estado"
onchange="javascript:ajaxPopulaCombo(this.value, '../include/xml_cidade_ajax.php',
document.form.cidade);">
```

Quadro 7 - Select Field de estado

Já Quadro 8 no está o código do *comboBox* de cidades, que nada precisa a não ser a ID que foi passado pelo código em estado.

```
<select name="cidade" id="cidade">
```

Quadro 8 - Select Field de cidade

No Quadro 9 está descrita a criação do Objeto *XMLHttpRequest*

```
function GetXmlHttpRequest() {
    var ajax = null;
    try {
        ajax = new ActiveXObject("Microsoft.XMLHTTP");
        //alert("Microsoft.XMLHTTP");
        return ajax;
    } catch(e) {
        try {
            ajax = new ActiveXObject("Msxml2.XMLHTTP");
            //alert("Msxml2.XMLHTTP");
            return ajax;
        } catch(ex) {
            try {
                ajax = new XMLHttpRequest();
                //alert("XMLHttpRequest");
                return ajax;
            } catch(exc) {
                alert("Esse não suporta Ajax");
                return ajax;
            }
        }
    }
}
```

Quadro 9 - Criação do Objeto *XMLHttpRequest*

Já no Quadro 10 está descrita a função chamada pela modificação do *comboBox* de estado. Ela de início instancia um objeto *XMLHttpRequest* como descrito acima pelo Quadro 9.

```
function ajaxPopulaCombo(codigo, url, combo) {
    url = url+"?cod="+codigo;
    var ajax = GetXmlHttpRequest();
    combo.options.length = 1;
    ajax.open("GET", url, true);
    ajax.setRequestHeader("Content-Type", "application/x-www-form-urlencoded");
    ajax.onreadystatechange = function () {
        if (ajax.readyState == 0) { //Nao inicializado
            combo.options[0] = new Option('Problemas na inicializaçao','0');
        } else if (ajax.readyState == 1) { //Carregando
            combo.options[0] = new Option('Carregando...', '0');
        } else if (ajax.readyState == 4) { //Completado
            var dataArray = ajax.responseXML.getElementsByTagName("item");
            if(dataArray.length > 0) {
                combo.options[0] = new Option('Selecione', '');
                for(var i=0; i<dataArray.length; i++) {
                    var xmlNode = dataArray[i];
                    var codigo =
xmlNode.getElementsByTagName("codigo")[0].firstChild.nodeValue;
```

```
        var descricao =
xmlNode.getElementsByTagName("descricao")[0].firstChild.nodeValue;
        combo.options[i+1] = new Option(descricao, codigo);
    }
    } else { //caso o XML volte vazio, printa a mensagem abaixo
        combo.options[0] = new Option('Nenhum item cadastrado','0');
    }
    }
}
}
ajax.send(null);
}
```

Quadro 10 - Função ajaxPopulaCombo

3.3.2.3 GOOGLE MAPS API

Outro aspecto importante do sistema é a sua integração com a API do Google Maps. Ao cadastrar cada “Boteco”, o usuário-administrador fornece a real latitude e longitude do “Boteco”, que pode ser coletado através de um GPS¹⁰, ou localizado através do site do Google Maps.

Assim cadastrado o “Boteco” há uma página chamada de “Boteco Maps” onde o usuário pode facilmente identificar todos os estabelecimentos cadastrados e ter acesso à sua página com as restantes informações, isso sem sair do sistema em momento algum.

A Figura 19 demonstra a utilização dessa ferramenta pelo usuário final.

¹⁰ Global Positioning System: Aparelho que aponta em números a sua posição no globo terrestre.

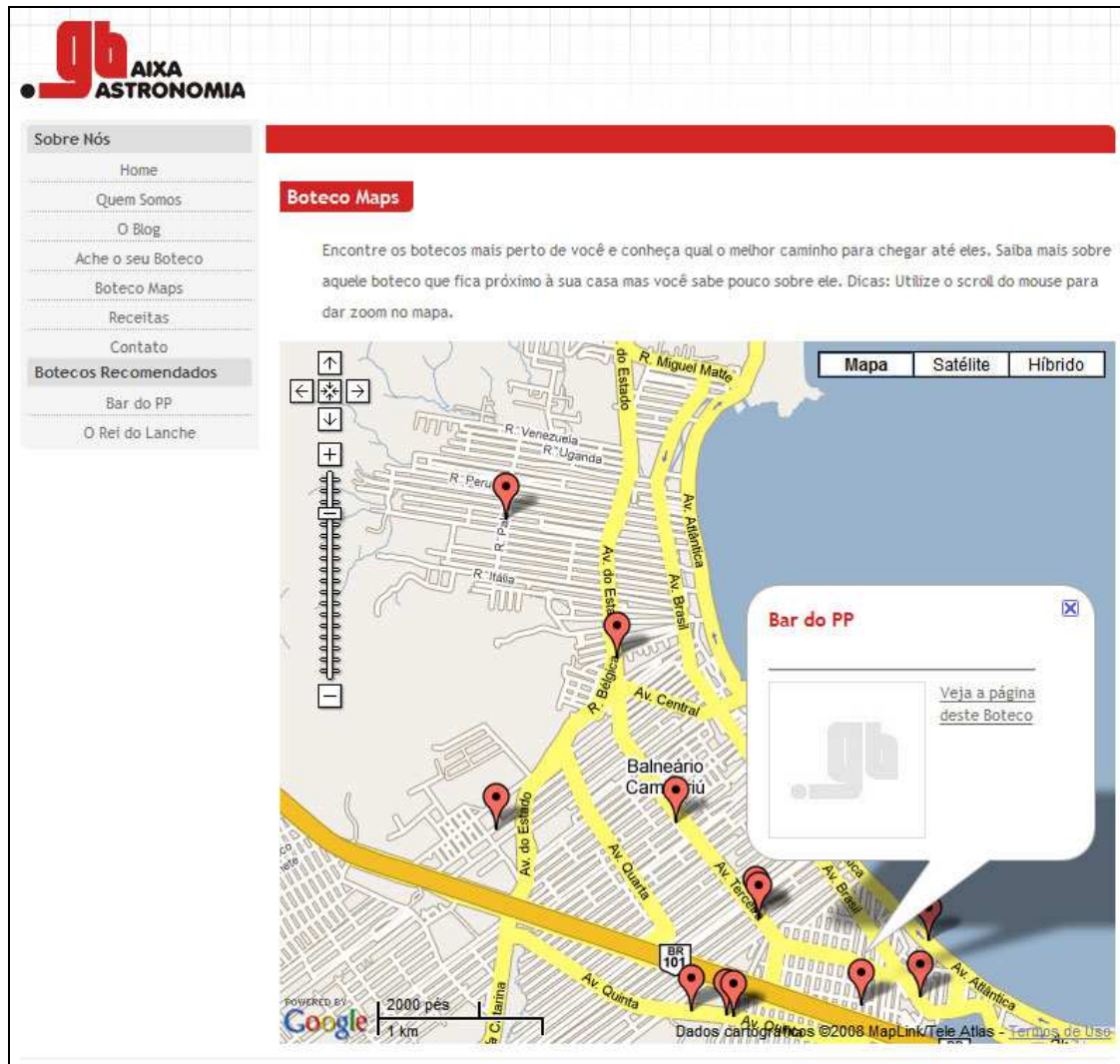


Figura 19 - Utilização do Boteco Maps

Para se obter tal funcionalidade foi gerado uma lista em XML, exemplificada no Quadro 11, gerada dinamicamente do banco de dados para ser carregada pela API do Google Maps.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<markers>
  <marker codigo="1" titulo="Bar do PP" lat="-27.00459500" lon="-48.62488700"
url="/boteco/1-Bar-do--PP.html" nivel="" selected="0" />
  <marker codigo="16" titulo="Bar do Miro" lat="-26.97816200" lon="-
48.64663400" url="/boteco/16-Bar-do-Miro.html" nivel="" selected="0" />
  <marker codigo="6" titulo="Coqueiros, Lanchonete" lat="-26.99986800" lon="-
48.63113700" url="/boteco/6-Coqueiros-Lanchonete.html" nivel="" selected="0" />
  <marker codigo="8" titulo="Egídio, Bar do" lat="-27.27634500" lon="-
48.85011800" url="/boteco/8-Egídio-Bar-do.html" nivel="" selected="0" />
</markers>
```

Quadro 11 - Dados XML gerados para o Google Maps API

Na página em questão foi adicionado um IFRAME do Google Maps e o código *JavaScript* que carrega o XML de dados para inserir no mapa a localização de cada “Boteco”. Este código está descrito no Quadro 12, a seguir.

```
function load() {
```

```

if (GBrowserIsCompatible()) {
    var map = new GMap2(document.getElementById("GMap"));
    map.addControl(new GScaleControl());
    map.addControl(new GLargeMapControl());
    map.addControl(new GMapTypeControl());
    map.enableScrollWheelZoom(true);
    map.setCenter(new GLatLng(0,0), 13);

GDownloadUrl("<?=>URL_PATH?>include/xml_boteco_maps.php?codBoteco=<?=>$codBoteco?>",
function(data, responseCode) {
    var xml = GXml.parse(data);
    var markers = xml.documentElement.getElementsByTagName("marker");
    var bounds = new GLatLngBounds();
    var boundsOnSelected = new GLatLngBounds();
    for (var i = 0; i<markers.length; i++) {
        var codigo = markers[i].getAttribute("codigo");
        var titulo = markers[i].getAttribute("titulo");
        var url = markers[i].getAttribute("url");
        var nivel = markers[i].getAttribute("nivel");
        var selected= markers[i].getAttribute("selected");
        var point = new GLatLng(parseFloat(markers[i].getAttribute("lat")),
parseFloat(markers[i].getAttribute("lon")));
        bounds.extend(point);
        if(selected == 1) {
            boundsOnSelected.extend(point);
        }
        var marker = createMarker(point, codigo, titulo, url, nivel);
        map.addOverlay(marker);
    }
    if(!boundsOnSelected.isEmpty()) {
        map.setCenter(boundsOnSelected.getCenter(),
map.getBoundsZoomLevel(boundsOnSelected)-1);
    } else {
        map.setCenter(bounds.getCenter(), map.getBoundsZoomLevel(bounds));
    }
    });
}
}
//Função para criar o marcador de cada Boteco no mapa
function createMarker(point, codigo, titulo, url, nivel) {
    var marker = new GMarker(point);
    //O que aparece dentro do balão do Boteco
    var html = '<div style="width:190px;"><h3>' + titulo + '</h3><hr><div
style="float:left; margin-right:10px; border:1px solid #dfdfdf; padding:5px;"></div><a href="' +
url + '">Veja a página deste Boteco</a></div>';

    GEvent.addListener(marker, 'mouseover', function() {
        marker.openInfoWindowHtml(html);
    });
    return marker;
}
window.onload = load;
window.onunload=GUnload();

```

Quadro 12 – Código para mostrar os Botecos no Mapa

3.3.2.4 GOOGLE ANALYTICS

O requisito RF10 diz que o sistema de mídias deve ter um relatório de visualização das mídias integrado ao Google Analytics. O Google trata como se fossem visualizações de

páginas esta contagem. A Figura 20 demonstra esta integração. O código necessário para que isto ocorra pode ser observado no Quadro 13. A separação é feita através de barras e as palavras entre elas formam a informação por completo. “Ads” é a designação interna para “Mídias”, “V” para “Visualizações”, “C” para “Cliques”. Por exemplo: houveram 339 visualizações no “Banner Full”, da empresa “Conti”, campanha “Insitucional, mídia “Conti Full”.

```
pageTracker._trackPageview('/Ads/V/Full/Conti/Instituional/Conti Full')
```

Quadro 13 - Código Tracker do Google Analytics

Desempenho do conteúdo			
Exibições de página [?] 679 % do total do site: 59,72%	Exibições de página únicas [?] 304 % do total do site: 49,35%	Tempo na página [?] 00:00:46 Média do site: 00:00:34 (35,41%)	Taxa de rejeição [?] 33,33% Média do site: 2,54% (1.211,11%)
URL	Exibições de página ↓	Exibições de página únicas	
1. /Ads/V/Full/Conti/Instituional/Conti Full	339	113	
2. /Ads/V/Full/Capa/Conti/Instituional/Conti Full	164	106	
3. /Ads/V/Full/Boteco/Conti/Instituional/Conti Full	80	34	
4. /Ads/V/Full/Boteco.Pesquisa/Conti/Instituional/Cont...	33	15	
5. /Ads/V/Full/Boteco.Maps/Conti/Instituional/Conti Full	31	17	
6. /Ads/V/Full/Instituional/Conti/Instituional/Conti Full	22	13	
7. /Ads/V/Full/Contato/Conti/Instituional/Conti Full	8	4	
8. /Ads/C/Full/Boteco/Conti/Instituional/Conti Full	1	1	
9. /Ads/C/Full/Conti/Instituional/Conti Full	1	1	
Localizar URL: <input type="text" value="contendo"/> <input type="text" value="ads"/> <input type="button" value="Ir"/>			

Figura 20 - Google Analytics

3.3.2.5 PESQUISA AVANÇADA DE BOTECOS

Conforme especificação, cada “Boteco” poderá estar cadastrado em várias categorias de classificação, por exemplo, ele poderá ter um “bom atendimento”, ter “mesa de sinuca”, “canha de bocha”, “cerveja gelada”, etc.

O usuário, ao fazer uma pesquisa avançada pode escolher pelos “Botecos” que contenham somente os atrativos que ele deseja.

A Figura 21 ilustra o modelo de dados respectivo.

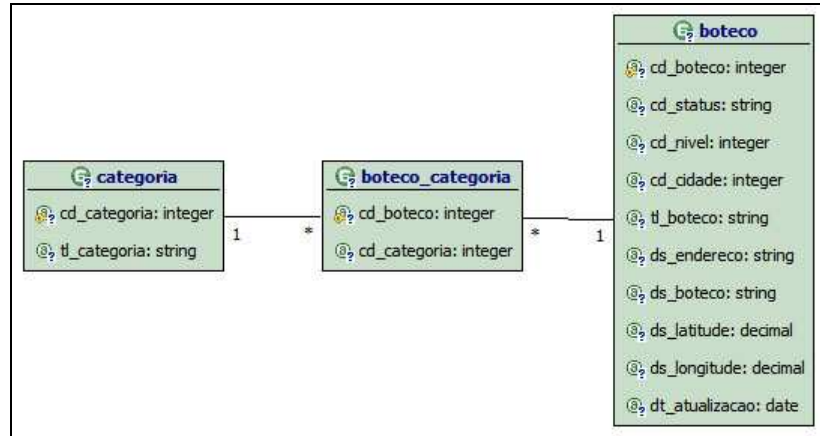


Figura 21 - Relação de dados Boteco Categoria

Esta pesquisa poderia ser obtida facilmente pelo comando INTERSECT de qualquer SGBD, porém, foi descoberto que a versão utilizada pelo MySQL para este sistema, assim como a maioria das versões de MySQL entre os servidores pela internet não oferece suporte para este comando.

O Quadro 14 ilustra um comando SQL utilizado então para a pesquisa. Em negrito está destacado a respectiva intersecção. Nota que a cada categoria adicionada à pesquisa, novo SUBSELECT é adicionado à query.

```

SELECT bo.cd_boteco,
       bo.tl_boteco,
       ci.tl_cidade,
       es.sg_estado,
       ni.tl_nivel
FROM boteco bo,
     cidade ci,
     estado es,
     status st,
     nivel ni
WHERE bo.cd_cidade = ci.cd_cidade AND
      ci.cd_estado = es.cd_estado AND
      bo.cd_status = st.cd_status AND
      bo.cd_nivel = ni.cd_nivel AND
      bo.cd_boteco IN
      (SELECT cd_boteco
       FROM boteco_categoria
       WHERE cd_categoria = 4 AND
          cd_boteco IN
          (SELECT cd_boteco
           FROM boteco_categoria
           WHERE cd_categoria = 2 AND
              cd_boteco IN
              (SELECT cd_boteco
               FROM boteco_categoria
               WHERE cd_categoria = 5 AND
                  cd_boteco IN
                  (SELECT cd_boteco
                   FROM boteco_categoria
                   WHERE cd_categoria = 1
                     )))
      )))
ORDER BY bo.tl_boteco

```

Quadro 14 - Comando SQL de pesquisa de Boteco

3.4 OPERACIONALIDADES DA IMPLEMENTAÇÃO

Nesta seção será demonstrado o funcionamento da implementação através da inclusão de um boteco, de fotos para este boteco e a criação de uma notícia.

3.4.1 INCLUSÃO DE UM BOTEÇO

Uma vez na tela inicial do Admin, conforme Figura 22, o usuário-admin irá selecionar no menu lateral o item “BOTEÇO”.

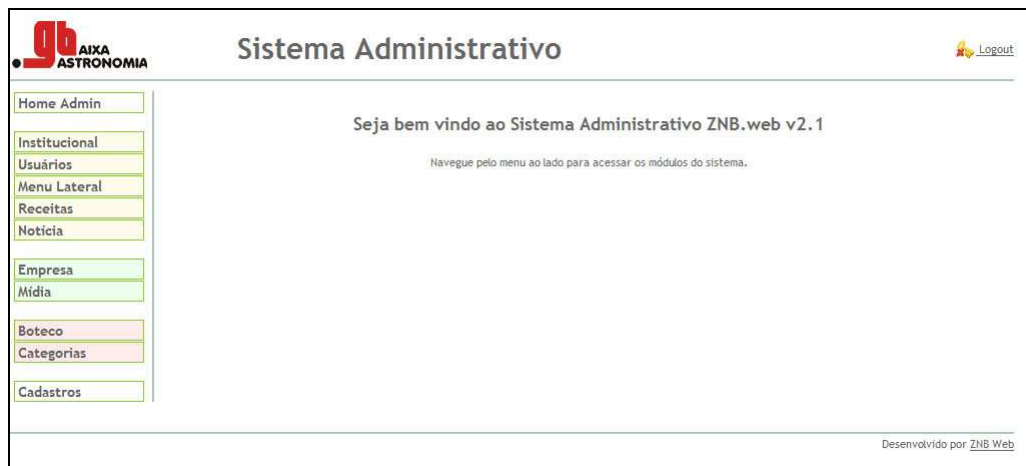


Figura 22 - Tela de entrada do Admin

Selecionado o item, o usuário tem a opção de pesquisar por um “boteco” já cadastrado para fazer modificações, inclusão de fotos, vídeos e personalidades, ou pode incluir um novo “Boteco” clicando no botão incluir no topo da página. A Figura 23 demonstra esta tela do sistema.

Para editar, incluir fotos, vídeos e personalidades para um boteco, basta clicar no item correspondente à mesma linha do nome do mesmo.

The screenshot shows the 'Sistema Administrativo' interface. At the top left is the logo for 'AIXA ASTRONOMIA'. The main header is 'Sistema Administrativo' with a 'Logout' link on the right. A navigation menu on the left lists: Home Admin, Institucional, Usuários, Menu Lateral, Receitas, Notícia, Empresa, Mídia, Boteco, Categorias, and Cadastros. The main content area is titled 'BOTEÇOS' and features an 'INCLUIR' button. Below this is a search form 'PESQUISAR BOTEÇO' with fields for 'Código', 'Nome' (containing 'bar'), 'Estado' (a dropdown menu), and 'Cidade' (a dropdown menu), followed by a 'Pesquisar' button. A table displays a list of bars with columns: TÍTULO, CIDADE/UF, STATUS, EDITAR, FOTOS, VIDEOS, and PERSON. The table contains six rows of data. At the bottom, a 'HELP' section provides instructions: 'Para incluir um novo Boteco, clique em "INCLUIR NOVO". Para editar um Boteco ou apagá-lo, basta clicar "EDITAR" no lado direito de seu título.' The footer indicates 'Desenvolvido por ZNB Web'.

TÍTULO	CIDADE/UF	STATUS	EDITAR	FOTOS	VIDEOS	PERSON.
Bar do Miro	Balneário Camboriu/SC	Operante				
Bar do PP	Balneário Camboriu/SC	Operante				
Egídio, Bar do	Sao Joao Batista/SC	Operante				
Imperatriz, Bar	Balneário Camboriu/SC	Operante				
Roque's Bar	Balneário Camboriu/SC	Operante				
São Paulo, Bar e Lanchonete	Balneário Camboriu/SC	Operante				

Figura 23 - Tela da relação de Botecos cadastrados

Clicando no link “Incluir”, a página de inclusão/alteração dos dados de um Boteco é mostrada, conforme Figura 24. Nesta tela o administrador informa os seguintes campos:

- título: o nome do “Boteco”;
- estado: o estado do “Boteco”;
- cidade: a respectiva cidade do “Boteco”;
- endereço: o endereço do “Boteco”;
- descrição: um campo descritivo onde o administrador irá preencher com as principais qualidades, informações sobre o respectivo “Boteco”. Este campo faz uso do Tiny MCE¹¹. Um editor HTML open-source desenvolvido em *JavaScript*;
- status: comboBox de seleção do estado do “Boteco” atualmente. Operante, em reforma ou fechado;
- nível: comboBox de seleção do nível social do estabelecimento, este sendo definido pelo administrador;
- categorias: seleção de quais as categorias o respectivo “Boteco” se enquadra.
- latitude: número real da latitude do estabelecimento;
- longitude: número real da longitude do estabelecimento;
- arquivo: imagem com a logomarca do “Boteco”.
- botão Adicionar: após o preenchimento dos dados, o botão Adicionar dá o submit

¹¹ Plataforma independente para a web baseada em *JavaScript* e HTML para edição de campos tipo TextArea. (<http://tinymce.moxiecode.com/>)

no formulário para inclusão dos dados no banco de dados;

m) botão Voltar: retorna à tela anterior do sistema, a listagem dos “botecos” cadastrados.

The screenshot displays the 'Sistema Administrativo' interface. On the left is a navigation menu with options like 'Home Admin', 'Institucional', 'Usuários', 'Menu Lateral', 'Receitas', 'Noticia', 'Empresa', 'Mídia', 'Boteco', 'Categorias', and 'Cadastros'. The main content area is titled 'BOTECO' and contains a form with the following fields and options:

- Codigo:** 22
- Título:** Bareco
- Estado:** SC
- Cidade:** Blumenau
- Endereço:** Rua Joinville, 436
- Descrição:** A rich text editor containing the text: "Excelente Boteco Universitário. Pra começar falando da gastronomia nada melhor que falar da enorme porção de batata-frita de 1kg. Acompanhada ao topo por bacon frito e queijo mussarela. Excelente petisco".
- Status:** Operante
- Nível:** A
- Checkboxes:**
 - Cerveja Gelada
 - Bom Papo
 - Jogo do Bicho
 - Mesa de Sinuca
 - Atendimento de Primeira
 - Cancha de Bocha
- Latitude:** -26,90616000 (Ex: -27.23541267)
- Longitude:** -49,08463000 (Ex: -47.23763467)
- Arquivo:** A text input field with a 'Procurar...' button and '(100x?px)' text.

At the bottom of the form are three buttons: 'Alterar' (green), 'Deletar' (red), and 'Voltar' (green). The footer of the page reads 'Desenvolvido por ZNB Web'.

Figura 24 - Tela de inclusão/alteração de um "Boteco"

3.4.2 INCLUSÃO DE FOTOS PARA UM BOTECO

Na tela de relação de “botecos” cadastrados, conforme Figura 23, o usuário-admin pode optar por adicionar fotos para um determinado estabelecimento. Para tal, basta clicar no ícone correspondente na linha do “boteco” na qual ele deseja inserir fotos.

A Figura 25 demonstra a tela de fotos cadastradas para um determinado estabelecimento.



Figura 25 - Tela de fotos cadastradas para um determinado "boteco"

Para alterar uma foto, basta clicar no ícone de edição na linha correspondente à foto que se deseja alterar.

Para incluir uma nova foto basta clicar em “Incluir nova foto”. A Figura 26 demonstra a tela de inclusão de nova imagem. Neste tela o usuário-admin deve preencher os seguintes campos:

- legenda: o texto que faz relação com a foto, sua legenda;
- imagem: arquivo com extensão jpg à ser procurado na máquina do usuário-admin.

Caso o tamanho da imagem for maior do que o tamanho (altura e largura em pixels) especificado ao lado do campo, ela será reduzida para o tamanho respectivo. Uma amostra de 50x50 pixels também será gerada a partir da imagem original.



Figura 26 - Tela de inserção de foto para determinado "boteco"

A tela mostrada para o usuário final, com todos os dados do “Boteco” inclusive fotos, vídeos e personalidades, pode ser vista na Figura 27. Não havendo cadastros de fotos, vídeos

ou personalidades, os mesmos não aparecerão para o usuário final.

The screenshot shows the website interface for 'Bar do PP'. At the top left is the logo 'gb AIXA ASTRONOMIA'. A navigation menu on the left includes links for 'Sobre Nós', 'Home', 'Quem Somos', 'O Blog', 'Ache o seu Boteco', 'Boteco Maps', 'Receitas', 'Contato', and 'Botecos Recomendados'. The main content area features a red header for 'Bar do PP' with its address: 'Endereço: Rua 3300, 800' and 'Cidade: Balneário Camboriú/SC'. Below this is a 'Descrição' section detailing the bar's history and offerings. A list of items available at the bar is shown under the heading 'No Bar do PP tem', including 'Atendimento de Primeira', 'Bom Papo', 'Cerveja Gelada', 'Jogo do Bicho', 'Lanches', 'Petiscos', and 'TV (Jogos / Novela)'. At the bottom, there is a 'Fotos do Bar do PP' section with three small images. The footer contains a copyright notice for 2008 and a link to 'ZNE Web'.

Figura 27 – Tela do “boteco” para o usuário final

3.4.3 INCLUSÃO DE NOTÍCIAS

O módulo de inclusão de notícias para a capa do sistema pode ser encontrado a partir do menu lateral sob o link de “Notícias”. A Figura 28 demonstra a relação de todas as notícias cadastradas até o momento.

Para alterar uma nova notícia previamente cadastrada, basta clicar no ícone na linha correspondente ao título da notícia. Já para incluir uma nova notícia, basta clicar no link “Incluir nova”.

Sistema Administrativo

Logout

Home Admin

Institucional

Usuários

Menu Lateral

Receitas

Notícia

Empresa

Mídia

Boteco

Categorias

Cadastros

NOTÍCIAS

INCLUIR NOVA

CODIGO	TITULO	USUÁRIO	DATA-HORÁRIO	EDITAR
24	Tá saindo buchada no PP	Rion Brattig Correia	06/06/2008 - 20:55:10	
23	Via Crucis quinafeirina	Marcos Romano	06/06/2008 - 10:56:04	
22	Bar da Frau	Fábio Antônio Lancini	03/06/2008 - 21:38:24	
21	Novos botecos adicionados	Rion Brattig Correia	28/05/2008 - 23:29:41	
20	E começa a brincadeira...	Fábio Antônio Lancini	28/05/2008 - 01:36:01	
19	Seja Bem Vindo	Rion Brattig Correia	23/05/2008 - 09:35:04	

HELP

Para incluir um novo notícia, clique em "INCLUIR NOVA".
Para editar um notícia ou apagá-la, basta clicar "EDITAR" no lado direito de seu título.

Desenvolvido por ZNB Web

Figura 28 - Tela de notícias cadastradas

A Figura 29 mostra os campos de preenchimento para uma nova notícia.

- usuário: o usuário que está postando a notícia;
- título: o título da notícia;
- sub título: o sub título da notícia;
- descrição: campo descritivo da notícia. Este campo também faz uso do Tiny MCE.

Sistema Administrativo

Logout

Home Admin

Institucional

Usuários

Menu Lateral

Receitas

Notícia

Empresa

Mídia

Boteco

Categorias

Cadastros

NOTÍCIAS

Usuário: Rion Brattig Correia

Título: Seja Bem Vindo

Sub Título: Surge o mais novo guia da Baixa Gastronomia de Botecos do Vale do Itajaí.

Descrição:

Neste guia, você poderá achar com muita facilidade aquele Boteco perto de você. Escolha os atrativos que você estava procurando, como uma boa cerveja, um bom papo e claro, os petiscos mais deliciosos que só os melhores botecos possuem.

Fique a vontade, utilize nossa ferramenta de [procura](#) por cidade ou atrativo, ou navegue direto pelo [Boteco Maps](#) para encontrar de acordo com a sua região.

Endereço:

Data de Publicação: 12/07/2008 - 22:00:45

Alterar Deletar Voltar

Figura 29 - Tela de cadastro de nova notícia

O resultado da notícia incluída, na visão do usuário final pode ser vista na Figura 30.

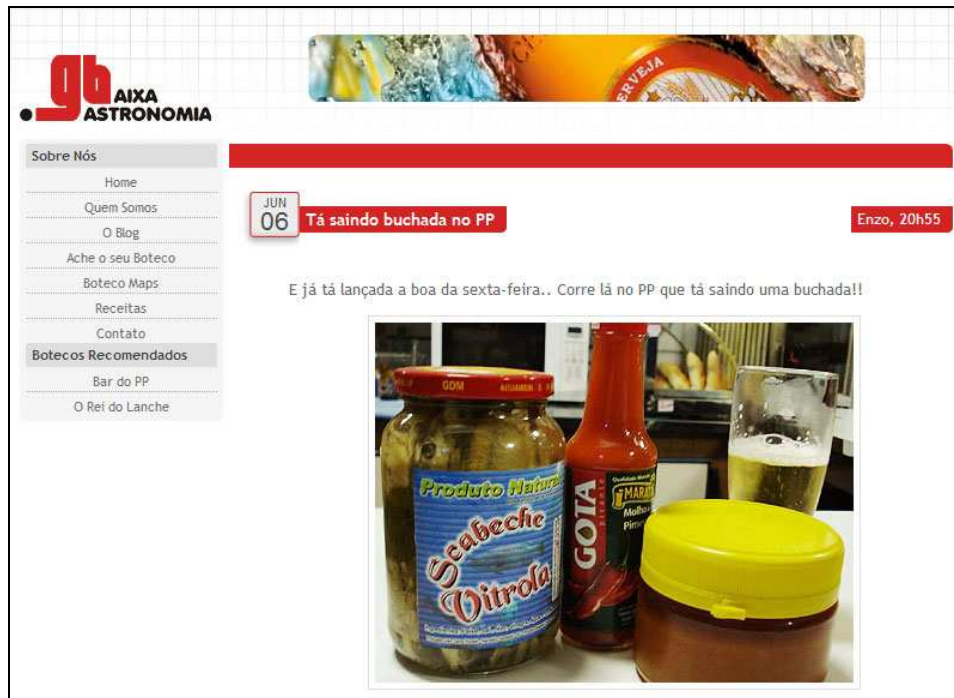


Figura 30 - Tela de notícias para o usuário final

3.5 RESULTADOS E DISCUSSÕES

O sistema apresentado neste trabalho permite o cadastramento e divulgação da Baixa Gastronomia no Vale do Itajaí na Internet. Está presente em diversos estabelecimentos conhecidos popularmente como “Botecos”.

Um dos diferenciais para este trabalho é a regionalização do conteúdo. Isto não significa que ele não irá poder crescer o suficiente por todo um estado, ou por vários. Mas ter de uma forma ampla, todos os estabelecimentos onde se possa saborear a Baixa Gastronomia, algo que é deixado de lado por gigantes como o Guia Quatro Rodas.

A agregação com o poderoso sistema de mapas do Google Maps e a facilidade que se tem para localizar e saber importantes pontos de cada estabelecimento é também um ponto importante no sistema.

O sistema também foi desenvolvido segundo o padrão especificado pelo W3C, desconsiderando o sistema de notícias em que os usuários podem inserir conteúdo não válido. O restante do sistema está em conformidade segundo o validador de páginas web da W3C.

Outro importante diferencial é o uso de uma ferramenta CASE voltada especialmente para a web para desenvolver o modelo de dados e de hiperlinks. Mostrar nos modelos como será a navegação do sistema, algo relativamente novo e que não é ainda muito difundido.

Este trabalho, em comparação com o Guia Quatro Rodas (2007), aborda de forma mais regional e de melhor abrangência os estabelecimentos. Também possui as notícias que colocam de forma opinativa e descritiva os achados e detalhes obtidos no “Boteco”. Possui também uma navegação mais intuitiva e fácil para o usuário final, além de incluir a fácil localização dos estabelecimentos por meio do Boteco Maps, algo não abrangida pelo Guia Quatro Rodas.

Com relação ao trabalho de Locatelli (2003), foi mostrado no trabalho atual como deve ser feita modelagem de um sistema real usando a WebML, algo que não foi mostrado pelo trabalho anterior. No atual houve, a partir dos requisitos obtidos, toda a especificação do sistema para somente assim o desenvolvimento do sistema proposto.

Já em relação ao trabalho de Gracik (2005), foi feita a modelagem em um ambiente totalmente voltado ao desenvolvimento web como é o caso da WebML, ao contrário do trabalho do Gracik em que a modelagem do sistema foi feito utilizando a UML com uma extensão para poder atender às especificações da web.

4 CONSIDERAÇÕES FINAIS

Ao findar este trabalho, pode-se concluir que os objetivos traçados inicialmente foram alcançados com sucesso. O sistema permite a divulgação dos ambientes gastronômicos de nossa região de uma forma fácil e sem esforço, usando as questões de usabilidade propostas.

Este trabalho irá fazer a sua parte na divulgação de nossa cultura tão amplamente variada. Irá trazer o que o brasileiro tem de melhor, que é a criatividade. Irá mostrar o que realmente está acontecendo nesta gastronomia simples, regional, mas de muito respeito. Servirá como base de conhecimento aos principais e melhores “Botecos” da região.

Usar as novas tecnologias emergentes sempre será um desafios para os programadores e desenvolvedores futuros. O Google está, dia-a-dia, trazendo séries de APIs que visam facilitar, e muito, a vida desses programadores. Estas APIs são *open-source* e estão completamente documentadas, o que facilita seu entendimento. Imprescindível poder contar com ferramentas tão avançadas com apenas uma simplicidade de códigos para fazer a ligação entre a ferramenta e o sistema proposto.

A WebML utilizada na especificação do sistema também se mostrou bastante significativa em seu papel. Percebeu-se que a sua utilização pode baixar o tempo necessário para a compreensão da totalidade do sistema pelos programadores, pois demonstra de modo fácil e direto como deve ser a navegação do sistema e como os dados devem fluir antes de qualquer código começar a ser feito.

Com relação à utilização do Ajax no sistema, notou-se um aumento significativo na usabilidade e na rapidez do sistema. Não houve grandes problemas para sua implementação que se mostrou bastante simples e com muitos recursos disponíveis na internet para buscar soluções aos problemas encontrados.

4.1 EXTENSÕES

Várias ampliações podem ser vistas para o futuro do sistema. Assim que ele obtiver certo número de visitas/dia, um sistema de comentários nas notícias poderá ser incluído, para que os visitantes possam deixar recados para as notícias que foram adicionadas.

Também um sistema de votação aberto ao público mediante cadastro poderá ser

colocado a disposição para se obter uma visão não apenas dos administradores do sistema, mas sim de todas as pessoas que estão diretamente envolvidas neste contexto.

Um módulo de *login* com controle de níveis de usuários também pode ser visto, para que haja a disponibilidade de abertura do sistema administrativo do sistema para usuários com permissão apenas para gerar conteúdo, notícias para o sistema. De Colunistas, estes usuários poderiam ser chamados.

Migrar o sistema para uma linguagem de programação mais robusta do que o PHP também pode ser uma extensão deste projeto. Algumas das linguagens sugeridas são o ASP.NET e o JSP (*Java Server Pages*).

REFERÊNCIAS BIBLIOGRÁFICAS

- BENELI, Claudemir. **Bonde**. Paraná, 2007. Disponível em: <http://www.bonde.com.br/colunistas/colunistasd.php?id_artigo=2035>. Acesso em: 27 set. 2007.
- CERI, Estefano; FRATERNALI, Piero; BONGIO, Aldo. **Web Modeling Language: a modeling language for designing Web Sites**. Dipartimento di Elettronica e Informazione, Politecnico di Milano, 2000. Disponível em: <<http://webml.elet.polimi.it/webml/page1.do>>. Acesso em 21 de abr. de 2008.
- FERRAZ, Ronaldo. **Construindo sites com padrão Web**. [São Paulo], 2003. Disponível em: <<http://kb.reflectivesurface.com/br/artigos/sitesComPadroesWeb/conteudo>>. Acesso em: 1 maio 2007.
- GOOGLE INC, **Google Maps: O que é o Google Maps**. Mountain View, CA, 2008. Disponível em <<http://maps.google.com.br/support/bin/answer.py?answer=7060&topic=10778>>. Acesso em: 07 Abr. 2008.
- _____. **Google Maps API**. Disponível em: <<http://code.google.com/intl/pt-BR/apis/maps/documentation/index.html>>. Acesso em: 11 jun 2008.
- GUIA QUATRO RODAS. A gente vai antes para você ir melhor. São Paulo. 2007.
- GRACIK, José Eduardo. **Sistema de Informações Executivas Baseado em Data WareHouse Aplicado a Cooperativa Central de Crédito**. 2005. 85f. Trabalho de Conclusão de Curso (Bacharelado em Ciências da Computação) - Centro de Ciências Exatas e Naturais, Universidade Regional de Blumenau, Blumenau.
- HAMA, Lia. Baixa Gastronomia. **Vida Simples**, São Paulo, v. 1, n. 25, 2005. Disponível em: <http://vidasimples.abril.com.br/subhomes/comer/comer_237950.shtml>. Acesso em: 29 abr. 2008.
- LOCATELLI, Márcio H. **Engenharia de software para o desenvolvimento de webapps e as metodologias OOADM e WEBML**. 2003. 57 f. Monografia (Especialista em Ciências da Computação) Programa de Pós-graduação em Ciências da Computação, Universidade Federal de Santa Catarina, Florianópolis.
- MOXIECODE. **Tiny MCE**. Disponível em: <<http://tinymce.moxiecode.com>>. Acesso em: 12 jun 2008.
- MURRAY, Greg. **Asynchronous Javascript Technology and XML (Ajax) With the Java Platform**. [Santa Clara,CA,USA], 2006. Disponível em: <<http://java.sun.com/developer/technicalArticles/J2EE/AJAX/>>. Acesso em: 29 abr. 2008.

MYSQL. The world's most popular open source database. Disponível em: <<http://www.mysql.com/about/>>. Acesso em: 11 de jun 2008.

NIELSEN, Jakob; LORANGER, Hoa. **Usabilidade na Web: projetando websites com qualidade.** Rio de Janeiro: Elsevier, 2007.

PHP. Hypertext processor. 2007. Disponível em: <<http://br.php.net>>. Acesso em: 18 set. 2007.

TABLELESS. Web standards com mandioca e strogonoff. 2007. Disponível em: <<http://www.tableless.com.br>>. Acesso em: 17 set. 2007.

W3C. The World Wide Web Consortium. 2007. Disponível em: <<http://www.w3c.org>>.

MURRAY, Greg. Asynchronous Javascript Technology and XML (Ajax) With the Java Platform. **Sun Developer Network.** Santa Clara, CA. Estados Unidos da América. Disponível em: <<http://java.sun.com/developer/technicalArticles/J2EE/AJAX/>>. Acesso em: 29 abr. 2008.

ZINABRE. Zinabre Ponto Com Ltda Me. Balneário Camboriú, 2008. Disponível em <<http://www.zinabre.com>>. Acesso em: 10 jul 2008.

APÊNDICE A – Código usado para gerar o banco de dados do sistema

```

/* Estrutura da tabela `boteco` */

CREATE TABLE `boteco` (
  `cd_boteco` int(10) unsigned NOT NULL auto_increment,
  `cd_status` int(10) unsigned NOT NULL default '0',
  `cd_nivel` int(10) unsigned NOT NULL default '0',
  `cd_cidade` int(10) unsigned NOT NULL default '0',
  `tl_boteco` varchar(200) default NULL,
  `ds_endereco` varchar(250) default NULL,
  `ds_boteco` text,
  `ds_latitude` decimal(12,8) NOT NULL default '0.00000000',
  `ds_longitude` decimal(12,8) NOT NULL default '0.00000000',
  `dt_atualizacao` date NOT NULL default '0000-00-00',
  PRIMARY KEY (`cd_boteco`),
  KEY `cd_status` (`cd_status`),
  KEY `cd_nivel` (`cd_nivel`),
  KEY `cd_cidade` (`cd_cidade`)
) ENGINE=InnoDB;

/* Estrutura da tabela `boteco_categoria` */

CREATE TABLE `boteco_categoria` (
  `cd_boteco` int(10) unsigned NOT NULL default '0',
  `cd_categoria` int(10) unsigned NOT NULL default '0',
  PRIMARY KEY (`cd_boteco`,`cd_categoria`),
  KEY `cd_categoria` (`cd_categoria`)
) ENGINE=InnoDB;

/* Estrutura da tabela `cadastro` */

CREATE TABLE `cadastro` (
  `cd_cadastro` int(10) unsigned NOT NULL auto_increment,
  `cd_cidade` int(10) unsigned NOT NULL default '0',
  `nm_cadastro` varchar(250) default NULL,
  `ds_email` varchar(250) default NULL,
  `ds_telefone` varchar(14) default NULL,
  `dt_nascimento` date default NULL,
  `dt_cadastro` date default NULL,
  `flsexo` tinyint(3) unsigned default NULL,
  `fl_recnote` tinyint(3) unsigned default NULL, /* Recebe Notícia */
  `fl_recsms` tinyint(3) unsigned default NULL, /* Recebe SMS */
  PRIMARY KEY (`cd_cadastro`)
) ENGINE=InnoDB;

/* Estrutura da tabela `campanha` */

CREATE TABLE `campanha` (
  `cd_campanha` int(10) unsigned NOT NULL auto_increment,
  `cd_empresa` int(10) unsigned NOT NULL default '0',
  `tl_campanha` varchar(250) default NULL,
  PRIMARY KEY (`cd_campanha`)
) ENGINE=InnoDB;

/* Estrutura da tabela `categoria` */

CREATE TABLE `categoria` (
  `cd_categoria` int(10) unsigned NOT NULL auto_increment,
  `tl_categoria` varchar(250) default NULL,
  PRIMARY KEY (`cd_categoria`)
) ENGINE=InnoDB;

/* Estrutura da tabela `cidade` */

```

```

CREATE TABLE `cidade` (
  `cd_cidade` int(11) unsigned NOT NULL auto_increment,
  `tl_cidade` varchar(100) NOT NULL default '',
  `cd_estado` int(2) NOT NULL default '0',
  PRIMARY KEY (`cd_cidade`),
  KEY `cd_estado` (`cd_estado`)
) ENGINE=InnoDB;

/* Estrutura da tabela `empresa` */

CREATE TABLE `empresa` (
  `cd_empresa` int(10) unsigned NOT NULL auto_increment,
  `tl_empresa` varchar(250) default NULL,
  PRIMARY KEY (`cd_empresa`)
) ENGINE=InnoDB;

/* Estrutura da tabela `estado` */

CREATE TABLE `estado` (
  `cd_estado` int(3) NOT NULL auto_increment,
  `ds_estado` varchar(100) NOT NULL default '',
  `sg_estado` char(2) NOT NULL default '',
  PRIMARY KEY (`cd_estado`)
) ENGINE=InnoDB;

/* Estrutura da tabela `foto` */

CREATE TABLE `foto` (
  `cd_foto` int(10) unsigned NOT NULL auto_increment,
  `cd_boteco` int(10) unsigned NOT NULL default '0',
  `ds_legenda` varchar(250) default NULL,
  PRIMARY KEY (`cd_foto`),
  KEY `cd_boteco` (`cd_boteco`)
) ENGINE=InnoDB;

/* Estrutura da tabela `institucional` */

CREATE TABLE `institucional` (
  `cd_institucional` int(10) unsigned NOT NULL auto_increment,
  `tl_institucional` varchar(250) default NULL,
  `ds_institucional` text,
  PRIMARY KEY (`cd_institucional`)
) ENGINE=MyISAM;

/* Estrutura da tabela `menu_lateral` */

CREATE TABLE `menu_lateral` (
  `cd_menu` int(10) unsigned NOT NULL auto_increment,
  `cd_pai` int(10) unsigned NOT NULL default '0',
  `tl_menu` varchar(50) default NULL,
  `ds_url` varchar(250) default NULL,
  PRIMARY KEY (`cd_menu`)
) ENGINE=InnoDB;

/* Estrutura da tabela `midia` */

CREATE TABLE `midia` (
  `cd_midia` int(10) unsigned NOT NULL auto_increment,
  `cd_campanha` int(10) unsigned NOT NULL default '0',
  `cd_tipo` int(10) unsigned NOT NULL default '0',
  `cd_formato` int(10) unsigned NOT NULL default '0',
  `tl_midia` varchar(200) default NULL,
  `dt_midia` date default NULL,
  `ds_url` tinytext,
  `fl_midia` tinyint(3) unsigned default NULL,
  PRIMARY KEY (`cd_midia`),
  KEY `cd_tipo` (`cd_tipo`),
  KEY `cd_formato` (`cd_formato`),

```

```

    KEY `cd_campanha` (`cd_campanha`)
) ENGINE=InnoDB;

/* Estrutura da tabela `midia_formato` */

CREATE TABLE `midia_formato` (
  `cd_formato` int(10) unsigned NOT NULL auto_increment,
  `ds_formato` varchar(70) default NULL,
  `ds_extensao` varchar(10) default NULL,
  PRIMARY KEY (`cd_formato`)
) ENGINE=InnoDB;

/* Estrutura da tabela `midia_sessao` */

CREATE TABLE `midia_sessao` (
  `cd_midia` int(10) unsigned NOT NULL default '0',
  `cd_sessao` int(10) unsigned NOT NULL default '0',
  PRIMARY KEY (`cd_midia`,`cd_sessao`),
  KEY `cd_sessao` (`cd_sessao`)
) ENGINE=InnoDB;

/* Estrutura da tabela `midia_tipo` */

CREATE TABLE `midia_tipo` (
  `cd_tipo` int(10) unsigned NOT NULL auto_increment,
  `ds_tipo` varchar(100) default NULL,
  `nr_largura` int(10) default NULL,
  `nr_altura` int(10) default NULL,
  PRIMARY KEY (`cd_tipo`)
) ENGINE=InnoDB;

/* Estrutura da tabela `nivel` */

CREATE TABLE `nivel` (
  `cd_nivel` int(10) unsigned NOT NULL auto_increment,
  `tl_nivel` varchar(150) default NULL,
  PRIMARY KEY (`cd_nivel`)
) ENGINE=InnoDB;

/* Estrutura da tabela `noticia` */

CREATE TABLE `noticia` (
  `cd_noticia` int(10) unsigned NOT NULL auto_increment,
  `cd_usuario` int(10) unsigned NOT NULL default '0',
  `tl_noticia` varchar(250) default NULL,
  `st_noticia` varchar(250) NOT NULL default '',
  `ds_noticia` text,
  `dt_noticia` datetime default NULL,
  PRIMARY KEY (`cd_noticia`),
  KEY `cd_usuario` (`cd_usuario`)
) ENGINE=InnoDB;

/* Estrutura da tabela `personalidade` */

CREATE TABLE `personalidade` (
  `cd_personalidade` int(10) unsigned NOT NULL auto_increment,
  `cd_boteco` int(10) unsigned NOT NULL default '0',
  `nm_personalidade` varchar(250) default NULL,
  `ds_personalidade` text,
  PRIMARY KEY (`cd_personalidade`),
  KEY `cd_boteco` (`cd_boteco`)
) ENGINE=InnoDB;

/* Estrutura da tabela `receita` */

CREATE TABLE `receita` (
  `cd_receita` int(10) unsigned NOT NULL auto_increment,
  `tl_receita` varchar(250) default NULL,

```

```

    `ds_receita` text,
    `dt_receita` date default NULL,
    PRIMARY KEY (`cd_receita`)
) ENGINE=InnoDB;

/* Estrutura da tabela `sessao` */

CREATE TABLE `sessao` (
  `cd_sessao` int(10) unsigned NOT NULL auto_increment,
  `tl_sessao` varchar(50) default NULL,
  PRIMARY KEY (`cd_sessao`)
) ENGINE=InnoDB;

/* Estrutura da tabela `status` */

CREATE TABLE `status` (
  `cd_status` int(10) unsigned NOT NULL auto_increment,
  `tl_status` varchar(100) default NULL,
  PRIMARY KEY (`cd_status`)
) ENGINE=InnoDB;

/* Estrutura da tabela `usuario` */

CREATE TABLE `usuario` (
  `cd_usuario` int(10) unsigned NOT NULL auto_increment,
  `cd_nivel` tinyint(3) unsigned NOT NULL default '0',
  `nm_usuario` varchar(100) NOT NULL default '',
  `ds_email` varchar(150) NOT NULL default '',
  `ds_pseudonimo` varchar(100) NOT NULL default '',
  `ds_senha` varchar(20) NOT NULL default '',
  PRIMARY KEY (`cd_usuario`),
  UNIQUE KEY `ds_email` (`ds_email`),
  KEY `cd_nivel` (`cd_nivel`)
) ENGINE=InnoDB;

/* Estrutura da tabela `usuario_nivel` */

CREATE TABLE `usuario_nivel` (
  `cd_nivel` tinyint(3) unsigned NOT NULL default '0',
  `tl_nivel` varchar(250) NOT NULL default '',
  PRIMARY KEY (`cd_nivel`)
) ENGINE=InnoDB;

/* Estrutura da tabela `video` */

CREATE TABLE `video` (
  `cd_video` int(10) unsigned NOT NULL auto_increment,
  `cd_boteco` int(10) unsigned NOT NULL default '0',
  `tl_video` varchar(250) default NULL,
  `ds_link` varchar(50) default NULL,
  PRIMARY KEY (`cd_video`),
  KEY `cd_boteco` (`cd_boteco`)
) ENGINE=InnoDB;

/* Chave(s) estrangeira(s) para a tabela `boteco` */
ALTER TABLE `boteco`
  ADD CONSTRAINT `boteco_ibfk_7` FOREIGN KEY (`cd_status`) REFERENCES `status`
  (`cd_status`),
  ADD CONSTRAINT `boteco_ibfk_8` FOREIGN KEY (`cd_nivel`) REFERENCES `nivel`
  (`cd_nivel`),
  ADD CONSTRAINT `boteco_ibfk_9` FOREIGN KEY (`cd_cidade`) REFERENCES `cidade`
  (`cd_cidade`);

/* Chave(s) estrangeira(s) para a tabela `boteco_categoria` */
ALTER TABLE `boteco_categoria`
  ADD CONSTRAINT `boteco_categoria_ibfk_1` FOREIGN KEY (`cd_boteco`) REFERENCES
  `boteco` (`cd_boteco`),
  ADD CONSTRAINT `boteco_categoria_ibfk_2` FOREIGN KEY (`cd_categoria`) REFERENCES

```

```

`categoria` (`cd_categoria`);

/* Chave(s) estrangeira(s) para a tabela`cidade` */
ALTER TABLE `cidade`
  ADD CONSTRAINT `cidade_ibfk_1` FOREIGN KEY (`cd_estado`) REFERENCES `estado`
  (`cd_estado`);

/* Chave(s) estrangeira(s) para a tabela`foto` */
ALTER TABLE `foto`
  ADD CONSTRAINT `foto_ibfk_1` FOREIGN KEY (`cd_boteco`) REFERENCES `boteco`
  (`cd_boteco`);

/* Chave(s) estrangeira(s) para a tabela`midia`*/
ALTER TABLE `midia`
  ADD CONSTRAINT `midia_ibfk_1` FOREIGN KEY (`cd_tipo`) REFERENCES `midia_tipo`
  (`cd_tipo`),
  ADD CONSTRAINT `midia_ibfk_2` FOREIGN KEY (`cd_formato`) REFERENCES
  `midia_formato` (`cd_formato`);

/* Chave(s) estrangeira(s) para a tabela`midia_sessao` */
ALTER TABLE `midia_sessao`
  ADD CONSTRAINT `midia_sessao_ibfk_1` FOREIGN KEY (`cd_midia`) REFERENCES `midia`
  (`cd_midia`),
  ADD CONSTRAINT `midia_sessao_ibfk_2` FOREIGN KEY (`cd_sessao`) REFERENCES
  `sessao` (`cd_sessao`);

/* Chave(s) estrangeira(s) para a tabela`noticia` */
ALTER TABLE `noticia`
  ADD CONSTRAINT `noticia_ibfk_1` FOREIGN KEY (`cd_usuario`) REFERENCES `usuario`
  (`cd_usuario`);

/* Chave(s) estrangeira(s) para a tabela`personalidade` */
ALTER TABLE `personalidade`
  ADD CONSTRAINT `personalidade_ibfk_1` FOREIGN KEY (`cd_boteco`) REFERENCES
  `boteco` (`cd_boteco`);

/* Chave(s) estrangeira(s) para a tabela`usuario` */
ALTER TABLE `usuario`
  ADD CONSTRAINT `usuario_ibfk_1` FOREIGN KEY (`cd_nivel`) REFERENCES
  `usuario_nivel` (`cd_nivel`);

/* Chave(s) estrangeira(s) para a tabela`video` */
ALTER TABLE `video`
  ADD CONSTRAINT `video_ibfk_1` FOREIGN KEY (`cd_boteco`) REFERENCES `boteco`
  (`cd_boteco`);

```