

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
CURSO DE CIÊNCIA DA COMPUTAÇÃO

CLAUDIO JOSÉ CASTALDELLO BUSATTO

O que tá valendo?
Um sistema Web de recomendação de
eventos

Trabalho de Graduação.

Prof. Dr. Marcelo Soares Pimenta
Orientador

Porto Alegre, dezembro de 2013.

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

Reitor: Prof. Carlos Alexandre Netto

Vice-Reitor: Prof. Rui Vicente Oppermann

Pró-Reitor de Graduação: Prof. Sérgio Roberto Kieling Franco

Diretor do Instituto de Informática: Prof. Luís da Cunha Lamb

Coordenador do Curso de Ciência da Computação: Prof. Raul Fernando Weber

Bibliotecário-Chefe do Instituto de Informática: Alexander Borges Ribeiro

AGRADECIMENTOS

Em primeiro lugar, agradeço a minha família por todo o apoio e incentivo durante toda esta etapa importante em minha vida, em especial nos momentos difíceis, nos quais pensei em desistir.

Em segundo lugar, a minha namorada Patrícia Strack por todo o amor, apoio, carinho e compreensão, principalmente nos momentos em que precisei me ausentar.

Agradeço à Universidade Federal do Rio Grande do Sul pela minha formação, aos professores do Instituto de Informática por todos os ensinamentos e conselhos que certamente levarei para o resto de minha vida, em especial ao meu orientador Prof. Dr. Marcelo Soares Pimenta pela confiança em mim depositada e por todo o auxílio na realização deste trabalho.

Por fim, agradeço a todos os amigos que fiz durante a graduação, em especial à “Gurizada”, pela parceria nas madrugadas de estudo, churrascos na casa do Potter e clássicos jogos de futebol QCB versus Void.

Um agradecimento especial aos colegas de time do QCB, campeão do DACOMPeonato 2009/1. Vamos em busca de mais títulos!

SUMÁRIO

LISTA DE ABREVIATURAS E SIGLAS.....	7
LISTA DE FIGURAS	8
LISTA DE QUADROS.....	10
RESUMO	11
1 INTRODUÇÃO.....	13
2 TRABALHOS RELACIONADOS	15
2.1 Qual é a boa.....	15
2.2 Curtindo POA	16
2.3 Funplace	16
2.4 Eventful	17
3 CONCEITOS UTILIZADOS	18
3.1 Evento.....	18
3.2 Visualização da Informação.....	18
3.2.1 Treemapping.....	18
3.3 Sistemas de recomendação.....	19
3.3.1 Dados explícitos e implícitos	20
3.3.2 Saída de um sistema de recomendação	21
3.3.3 Identificação do usuário	21
3.3.4 Técnicas de recomendação.....	21
3.3.4.1 Recomendação colaborativa.....	22
3.3.4.2 Recomendação demográfica	22
3.3.4.3 Recomendação baseada em conteúdo	22
3.3.4.4 Recomendação baseada em utilidade.....	23
3.3.4.5 Recomendação baseada em conhecimento.....	23
3.3.4.6 Sistemas de recomendação híbridos.....	23

3.4 Coletores de Documentos	24
3.5 Model-View-Controller	24
3.6 Ferramentas utilizadas	25
3.6.1 Web.....	25
3.6.2 MySQL.....	25
3.6.3 Javascript.....	25
3.6.4 HTML.....	26
3.6.4.1 DOM - Modelo de Objetos para Documentos.....	26
3.6.5 CSS.....	27
3.6.6 Framework Yii.....	27
3.6.6.1 A implementação do MVC utilizada no framework Yii.....	28
3.6.7 Facebook API.....	29
4 SISTEMA “O QUE TÁ VALENDO?”	30
4.1 Projeto do sistema	30
4.1.1 Requisitos do sistema	30
4.1.2 User Stories	30
4.1.3 Requisitos Não-funcionais	32
4.2 Arquitetura do Sistema	32
4.2.1 Módulo de Web crawling.....	33
4.2.2 Módulo de identificação e integração ao Facebook.....	35
4.2.3 Banco de Dados.....	36
4.2.3.1 Entidade User.....	37
4.2.3.2 Entidade User_Likes.....	38
4.2.3.3 Entidade User_Clicks.....	38
4.2.3.4 Entidade Event.....	38
4.2.3.5 Entidade Event_Type.....	38
4.2.3.6 Entidade Places.....	38
4.2.3.7 Entidade User_Rating_Places.....	39
4.2.3.8 Entidade Recommendation_Weights.....	39
4.2.4 Módulo de recomendação.....	39
4.2.4.1 A ideia.....	39
4.2.4.2 Captura de dados explícitos e implícitos.....	39
4.2.4.3 Avaliações e Pesos.....	40
4.2.4.4 Cálculo de recomendação.....	42
4.2.4.5 Visualização Personalizada.....	47
4.2.5 Framework Yii.....	49
4.2.5.1 Modelos.....	49
4.2.5.2 Visões.....	49
4.2.5.3 Controladores.....	50
5 USO E NAVEGAÇÃO NO SISTEMA	51
5.1 Tela para usuários	51
5.1.1 Aba "Eventos recomendados"	51
5.1.2 Aba "Ver todos os eventos".....	52
5.1.3 Aba "Locais"	53
5.1.4 Abas "Festas", "Shows" e "Teatros".....	55
5.1.5 Botão de Login via Facebook.....	55
5.2 Telas de gerência do sistema	56
5.2.1 Aba "Minerar Novos Eventos"	57

5.2.2	Aba "Alterar Peso das Recomendações"	59
5.2.3	Abas "Cadastrar Eventos" e "Cadastrar Locais"	60
5.2.4	Abas "Editar Eventos", "Editar Locais" e "Editar Usuários"	61

6 CONCLUSÕES.....64

6.1 Resultados..... 64

6.2 Limitações..... 64

6.3 Trabalhos futuros 65

REFERÊNCIAS66

LISTA DE ABREVIATURAS E SIGLAS

AJAX	Asynchronous Javascript and XML
API	Application Interface Programming
BD	Banco de Dados
CSS	Cascade Style Sheet
DB	Database
DOM	Document Object Model
HTML	HyperText Markup Language
MVC	Model-View-Controller
PHP	PHP Hypertext Preprocessor
UFRGS	Universidade Federal do Rio Grande do Sul
URL	Uniform Resource Locator
W3C	World Wide Web Consortium

LISTA DE FIGURAS

Figura 2.1: Exibição dos eventos no site “Qual é a boa”.	15
Figura 2.2: Exibição dos eventos no site “Curtindo POA”.	16
Figura 2.3: Exibição dos eventos no site “Funplace”.	17
Figura 3.1: Exemplo de um <i>treemap</i> .	19
Figura 3.2: Exemplo de avaliação explícita utilizada no site netflix.com.	21
Figura 3.3: Exemplo de recomendação baseada em conteúdo utilizada no netflix.com.	23
Figura 3.4: Recomendação híbrida.	24
Figura 3.5: Visão geral do modelo MVC.	25
Figura 3.6: Representação da árvore DOM a partir do código HTML de uma tabela.	27
Figura 3.7: Implementação do modelo MVC feita pelo framework Yii.	28
Figura 3.8: Representação do workflow em uma aplicação utilizando o framework Yii.	29
Figura 4.1: Visão geral da arquitetura do sistema.	33
Figura 4.2: Trecho do código HTML de um evento no site “Curtindo POA”.	34
Figura 4.3: Trecho do código HTML de outro evento no site “Curtindo POA”.	34
Figura 4.4: Exemplo do algoritmo de obtenção de dados no site “Curtindo POA”.	35
Figura 4.5: Trecho de código HTML do site “Curtindo POA”.	35
Figura 4.6: Widget de autenticação com o Facebook.	36
Figura 4.7: Diagrama Entidade-Relacionamento do sistema proposto.	37
Figura 4.8: Exemplo de como a entidade User_Likes armazena a informação das páginas curtidas no Facebook por um novo usuário do sistema.	40
Figura 4.9: Exemplo de avaliações realizadas por um usuário registrado.	40
Figura 4.10: Exemplo de saída dos cálculos de recomendação dos critérios definidos.	46
Figura 4.11: Saída com a soma dos valores de todos os critérios, ordenada de forma decrescente.	47
Figura 4.12: Trecho de código responsável por buscar os eventos recomendados ao usuário.	47
Figura 4.13: Retângulos utilizados para destacar os eventos recomendados.	48
Figura 4.14: Etapa final da recomendação, a exibição ao usuário.	48
Figura 4.15: Estrutura dos Modelos do sistema.	49
Figura 4.16: Estrutura de arquivos do componente Visão.	49
Figura 4.17: Estrutura das classes Controladores.	50
Figura 5.1: Tela inicial, com a aba “Eventos recomendados” selecionada.	52
Figura 5.2: Exibição das informações do evento em uma modal.	52
Figura 5.3: Aba “Ver todos os eventos”.	53
Figura 5.4: Visualização da aba locais por um usuário não-registrado.	54
Figura 5.5: Visualização da aba locais por um usuário registrado.	54
Figura 5.6: Pop-up de confirmação da avaliação.	55
Figura 5.7: Aba “Teatros”.	55
Figura 5.8: Pop-up de confirmação da concessão dos dados.	56
Figura 5.9: Opção, disponível a administradores, de gerência do sistema.	56
Figura 5.10: Aba “Minerar Novos Eventos”.	57
Figura 5.11: Exemplo dos dados obtidos do site “Qual é a boa?”.	58
Figura 5.12: Exemplo de evento que ocorrerá em um local ainda não cadastrado no sistema.	58
Figura 5.13: Modal para cadastro de novo local dentro da interface “Minerar Novos Eventos”.	59
Figura 5.14: Aba “Alterar Pesos das Recomendações”.	60
Figura 5.15: Aba de cadastro.	61
Figura 5.16: Aba “Editar Eventos”.	62
Figura 5.17: Opção “Visualização”.	62

Figura 5.18: Opção de “Edição”	63
Figura 5.19: Pop-up de confirmação na opção “Exclusão”	63

LISTA DE QUADROS

Quadro 3.1: Dados que podem ser capturados de maneira implícita.	20
Quadro 4.1: Critérios levados em consideração para diferentes usuários no sistema.	41
Quadro 4.2: Processo de cálculo de recomendação utilizando a análise de acessos.	42
Quadro 4.3: Processo de cálculo de recomendação utilizando a avaliação de locais.	44
Quadro 4.4: Processo do cálculo de recomendação utilizando a análise das páginas curtidas no Facebook.	45

RESUMO

Os sistemas *Web* de divulgação de eventos, em sua grande maioria, apenas informam os eventos agendados, não levando em consideração os gostos e preferências dos usuários. Isso acaba resultando em uma maior dificuldade para o usuário em encontrar os eventos que lhe são interessantes. O projeto aqui apresentado tem como objetivo desenvolver um sistema *Web* que busque obter, de grandes portais, a agenda cultural de Porto Alegre e recomendar as atividades que sejam do interesse de cada usuário. Neste trabalho serão abordados os conceitos de sistemas de recomendação, coletores de documentos, visualização da informação, bem como apresentadas as tecnologias utilizadas no desenvolvimento do sistema e suas aplicações. Por fim, serão exibidas telas da versão atual do sistema, juntamente com a explicação do seu uso, com o objetivo de demonstrar o funcionamento do mesmo.

Palavras-Chave: *Web*, internet, sistemas de recomendação, coletor de documentos.

“O que tá valendo?”
A web event recommendation system

ABSTRACT

Web recommender systems, in general, only inform scheduled events, not considering likes and preferences of users. This behavior results in a major difficulty to users to find their preferred events. The project presented here has the objective to show a web system that tries to acquire, from big websites, the cultural schedule of Porto Alegre and recommend the activities that will be interesting to each specific user. In this work we will show concepts of recommendation systems, crawlers, information visualization, as well as present the technologies used in the development of the system and its applications. Finally, screens of the current version of the system will be displayed along with explanations of its use, in order to demonstrate the system operation.

Keywords: Web, internet, recommendation systems, crawlers.

1 INTRODUÇÃO

A maneira como as pessoas se informam sobre o que está acontecendo na cidade, inclusive em relação à agenda de eventos culturais, mudou muito através dos tempos. Atualmente, não é mais possível pensar em propagação de informação sem automaticamente pensar em internet, em redes sociais. No cenário atual, Porto Alegre possui 1,1 milhão de usuários conectados, o que equivale a 73% da população da cidade¹. Os meios de divulgação *broadcast*, onde um veículo (como televisão, jornais, revistas) dissemina a informação para um grande grupo de pessoas, estão sendo substituído por meios *multicast*, nos quais qualquer usuário conectado à internet, além de receber informações, se torna um possível disseminador de informação, podendo receber e enviar dados para a rede. Esse aspecto acaba acarretando em uma enxurrada de informações diárias, em convites e divulgação de eventos que acabam não sendo de interesse de grande parte dos usuários atingidos.

Este trabalho consiste no projeto e desenvolvimento de um sistema intitulado “O que está valendo?”, que é um sistema *Web* de divulgação e recomendação dos eventos culturais que estão acontecendo em Porto Alegre. Ele se difere dos portais de divulgação tradicionais pela tentativa de aproximar o usuário do conteúdo que, provavelmente, vá ser de seu interesse, através de utilização de conceitos de recomendação e da utilização de diferentes formas de exibição do conteúdo. Além disso, é implementado um processo de coleta de dados onde as informações sobre os eventos a serem divulgados são extraídas dinamicamente dos principais portais de eventos e de sites de casas de shows, previamente definidos.

Os principais portais relacionados à divulgação e recomendação de eventos serão apresentados no capítulo 2. Em cada portal haverá uma descrição de suas características fundamentais e de como é feita a recomendação, caso haja, e a apresentação dos dados de eventos para os usuários.

Na realização deste trabalho, é utilizada a técnica de *Treemapping* para a visualização da informação dos eventos recomendados, conceitos de coleta de documentos na extração de dados de sites externos e sistemas de recomendação, que serão descritos no capítulo 3, juntamente com a apresentação das ferramentas utilizadas no desenvolvimento.

A arquitetura do sistema, bem como a aplicação dos conceitos apresentados e a utilização das ferramentas de desenvolvimento, serão demonstradas no capítulo 4.

¹ Dados de <http://www.slideshare.net/gurudocorretor/02-midias-sociais-no-mercado-imobiliario-porto-alegre-mariana-ferronato-vivareal>. Acesso em: novembro de 2013.

No capítulo 5, serão apresentadas as telas do sistema, com o objetivo de demonstrar o seu funcionamento. As conclusões, as limitações atuais, bem como sugestões de melhorias futuras, serão abordadas no capítulo 6.

2 TRABALHOS RELACIONADOS

Realizou-se uma pesquisa em sistemas que, assim como o “O que está valendo?”, tem como objetivo recomendar e/ou divulgar eventos para o público, através da internet. Em cada sistema buscou-se apresentar as principais características, formas de exibição das informações e como é realizada a recomendação, quando ela existe.

2.1 Qual é a boa

O site <http://www.qualeaboa.com.br> provê aos usuários informações sobre shows, festas e filmes em cartaz, em Porto Alegre. Apesar de não utilizar nenhuma forma de recomendação, possui um sistema de busca, com o qual os eventos podem ser filtrados por categoria, cidade ou local.

As informações são cadastradas por administradores do site e os eventos são exibidos, como pode ser visto na figura 2.1, em forma de uma tabela ordenada cronologicamente, onde cada evento apresenta uma imagem para divulgação, um título, um local, uma data em que ele ocorrerá e um link que, ao ser acessado, possibilita a obtenção de mais informações (como o endereço do local, sua localização em um mapa e, em alguns casos, informações sobre preço e reservas).

Figura 2.1: Exibição dos eventos no site “Qual é a boa”.

2.2 Curtindo POA

O site <http://www.curtindopoa.com.br> é uma iniciativa da Prefeitura Municipal de Porto Alegre para criar a agenda cultural da cidade, de forma colaborativa, e divulgar os eventos que estão acontecendo na cidade. O portal possui informações sobre atividades culturais, esportivas, turísticas e educacionais.

“Curtindo POA” utiliza integração com o *Facebook* para cadastro e acesso ao sistema. São os próprios usuários do sistema que cadastram os eventos que, após aprovação da moderação, são divulgados no site. Ele possibilita filtrar os eventos por data e horário, por tipo de evento (cinema, festas, shows, entre outros), pelo bairro em que ocorrerão e pelo custo.

Eventos que estão acontecendo na data e hora de acesso, ou iniciam nas próximas horas, possuem destaque na tela inicial. Os demais eventos são exibidos em um mosaico de miniaturas de imagens, como visto na figura 2.2, com informações básicas, ordenados de forma cronológica, possibilitando ao usuário registrado divulgá-lo em sua rede social. O site possui uma seção onde alguns eventos escolhidos (normalmente grandes shows) possuem destaque permanente.



Figura 2.2: Exibição dos eventos no site “Curtindo POA”.

2.3 Funplace

O site <http://www.funplace.com.br> é completamente integrado ao *Facebook*. Ele realiza a busca e a extração das informações de todos os eventos localizados nas capitais do Brasil e cadastrados na rede social para alimentar a sua base de dados. Mesmo assim, os usuários registrados também podem cadastrar manualmente os seus próprios eventos, a fim de utilizar o sistema para divulgá-los.

“Funplace” possui diversas formas de filtrar os eventos, entre eles é possível selecionar eventos por cidade, por percentual de homens e mulheres que confirmaram presença através da rede social e pelo local onde seus contatos no *Facebook* confirmaram presença. Além disso, é possível obter as informações básicas sobre o evento e visualizar fotos e comentários de outros usuários sobre os locais.

Os eventos são exibidos em um mosaico de imagens e informações sobre o mesmo, como visto na figura 2.3, ordenados por data e número de pessoas confirmadas no evento, através da rede social. Vale salientar que, pelo fato de utilizar integração com o *Facebook*, as ações de um usuário registrado no sistema são propagadas para a rede social do mesmo, ou seja, ao confirmar presença em um evento através do site, automaticamente esse evento será adicionado à rede social do usuário.

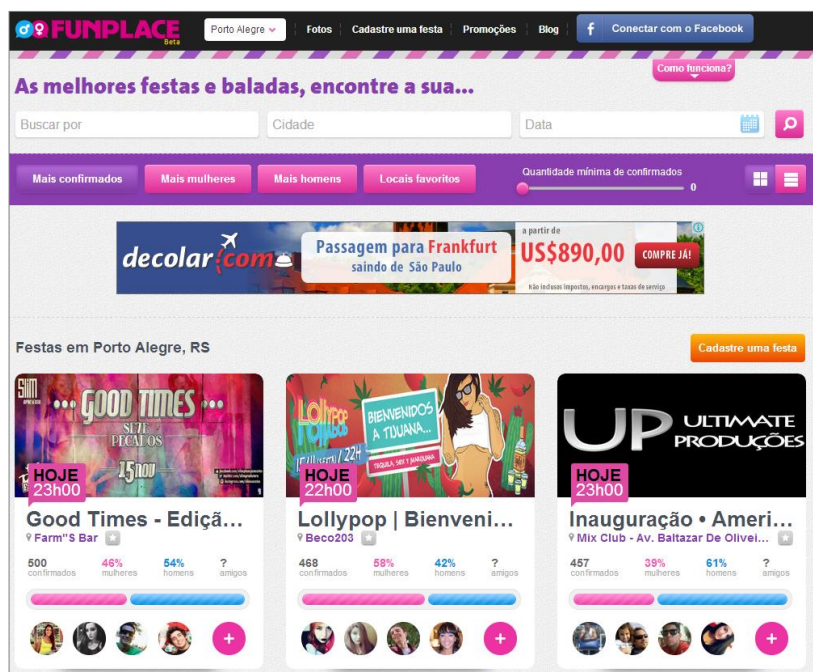


Figura 2.3: Exibição dos eventos no site “Funplace”.

2.4 Eventful

O site <http://www.eventful.com> busca divulgar e recomendar eventos utilizando a geolocalização do usuário como principal característica. Cada evento possui uma reputação no sistema e os eventos são separados por diferentes categorias, como esportes, shows, festas, entre outros. O sistema recomenda os eventos com melhor reputação e que estão acontecendo em uma região próxima ao usuário. É possível, caso o usuário deseje, definir a busca de eventos para uma cidade específica, diferente da detectada pelo sistema.

Um usuário pode se registrar no sistema, para isso é necessário realizar um cadastro, com o qual é feita uma pesquisa de interesses que tem como objetivo traçar um perfil e utilizar essas informações obtidas para refinar o processo de recomendação. É possível aos usuários registrados cadastrar seus próprios eventos para serem divulgados. Isso é, inclusive, incentivado pelo sistema.

Esse portal ainda não possui uma grande popularidade no Brasil, sendo assim não existem muitos eventos cadastrados no país.

3 CONCEITOS UTILIZADOS

Neste capítulo será apresentado a definição de evento, os conceitos de visualização da informação (com abordagem ao método de *treemapping*), de sistemas de recomendação e de coletores de documentos, a fim de facilitar a compreensão do funcionamento geral do sistema.

3.1 Evento

Segundo Antenor Nascentes, em seu Dicionário da Língua Portuguesa, a palavra *evento*, originária do latim *eventus*, é definida como: “Acontecimento, fato, ocorrência”. (NASCENTES, 1988, p. 265).

No sistema proposto, é considerado como um evento uma atividade cultural (festas, shows e peças de teatro) com um título, um local, uma data, uma imagem para divulgação e um link que, ao ser acessado, traga mais detalhes sobre o mesmo.

3.2 Visualização da informação

A visualização da informação é o estudo de representações visuais de dados abstratos com o objetivo de reforçar a cognição humana. “Técnicas de representação e interação visual tiram proveito da ampla largura de banda entre o olho humano e a mente que permite aos usuário ver, explorar e entender uma grande quantidade de informação de uma vez.” (THOMAS; COOK, 2005, p. 4).

Thomas e Cook (2005) ainda acrescentam que todas as pesquisas em visualização da informação têm como objetivo desenvolver novas abordagens, onde se consiga transmitir informação de maneira mais intuitiva.

Segundo Vliegen (2006), o uso de visualização da informação torna possível uma análise de maior quantidade de informação em comparação com apresentações utilizando tabelas, provendo, também, contexto ao resultado de uma busca, por tornar as discrepâncias mais visíveis.

3.2.1 Treemapping

Visualizações baseadas em área existem há décadas. Na arte, mosaico é uma famosa técnica que utiliza lâminas retangulares de diversos tamanhos para criar representações de imagens reais.

Sob essa ótica, os *treemaps* foram introduzidos por Johnson e Shneiderman (1991) como uma proposta de um método de visualização interativa para apresentação de informações hierárquicas. Um *treemap* mapeia dados para retângulos 2D onde informações com maior importância são exibidas em retângulos maiores na tela e

informações com menor importância, conseqüentemente, em retângulos que ocupam espaços menores, construção essa que pode ser vista na figura 3.1.

A principal característica de uma *treemap* é a sua construção recursiva, o que permite que ela seja estendida para uso em dados hierárquicos em qualquer número de níveis.

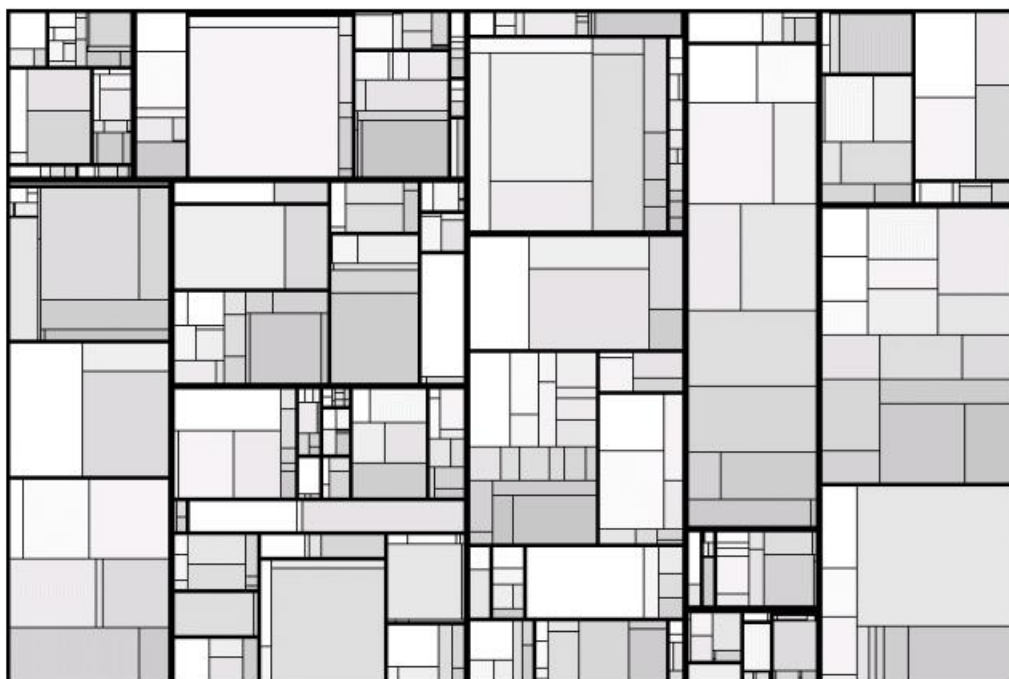


Figura 3.1: Exemplo de um *treemap*.

Fonte: Bederson, Shneiderman e Wattenberg (2002).

3.3 Sistemas de recomendação

A internet alcançou, em novembro de 2013, a marca de mais de 190 milhões de páginas ativas². A grande quantidade de informação disponível e a contínua expansão da rede aumenta a dificuldade dos usuários em encontrar a informação desejada. O'Donovan e Smyth (2005) apontam os sistemas de recomendação como uma das ferramentas que devem ser utilizadas na tentativa de solucionar esse problema.

Sistema de recomendação tradicional é definido por Resnick e Varian como “um sistema em que pessoas provêm recomendações como entradas as quais são agregadas pelo sistema e direcionadas aos recipientes apropriados” (RESNICK; VARIAN, 1997, p. 56)

Segundo Burke (2002), atualmente, o termo possui uma conotação mais ampla, descrevendo sistema de recomendação como qualquer sistema que produza uma recomendação individualizada como saída ou possua o efeito de guiar o usuário em um caminho personalizado para objetos interessantes ou úteis em uma larga quantidade de possibilidades. Especifica ainda que um sistema de recomendação precisa ter: (i) dados

² Dados de <http://news.netcraft.com/>. Acesso entre agosto e novembro de 2013.

prévios (informações que o sistema possui antes do processo de recomendação iniciar), (ii) dados de entrada (as informações acrescentadas por cada usuário ao sistema para o auxílio da recomendação) e (iii) um algoritmo que combine os dados prévios e dados de entrada e gere como saída uma recomendação.

3.3.1 Dados explícitos e implícitos

Os dados de entrada são informações fornecidas pelo usuário de maneira explícita (como na avaliação de um local, por exemplo) ou de forma implícita (como no registro de sua navegação pelo site), a fim de que se possa, com esses dados, entender o que o usuário gosta e fazer uso disso no processo de recomendação.

Segundo Nichols (1998), existem diversos dados que podem ser capturados de maneira implícita. As ações e os tipos de uso para isso, são apresentados no Quadro 3.1.

Quadro 3.1: Dados que podem ser capturados de maneira implícita.

Ação	Notas
Comprar	Comprar um novo item
Avaliar	Avaliar ou recomendar
Uso repetitivo	Ex.: busca pelo mesmo assunto
Salvar/Imprimir	Salvar um documento
Deletar	Deletar um item
Referenciar	Citar ou referenciar um item
Responder	Responder a um item
Marcar	Adicionar à lista de interesses
Examinar/Ler	Ex.: ler todo um artigo
Considerar	Ex.: ler somente o resumo de um artigo
Ter uma visão rápida	Ex.: ler somente o título de um artigo
Associar	Relacionar itens aos que estão sendo examinados
Consultar	Fazer associações entre os termos da consulta

Fonte: traduzida e adaptada de Nichols (1998).

Uma das formas mais utilizadas de obter-se dados explícitos é através da avaliação de itens. Essa avaliação, feita pelo usuário, pode ser binária (se aquela informação interessa ou não ao usuário) ou valorada (por meio de notas). Grandes sistemas normalmente utilizam uma combinação desses dois tipos de avaliação. A figura 3.2 é um exemplo de como pode ser realizada essa coleta explícita de dados. Na figura é possível ver que o usuário pode avaliar (de 1 a 5 estrelas, sendo 5 a melhor nota) o seu

interesse pelo item e/ou marcar a opção “Não tenho interesse”, caso o item recomendado, e provavelmente itens semelhantes, não sejam de seu interesse.



Figura 3.2: Exemplo de avaliação explícita utilizada no site *Netflix.com*.

3.3.2 Saída de um sistema de recomendação

Segundo Vozalis e Margaritis (2003), o último passo em um sistema de recomendação deve ser a produção de uma *predição* (um valor que represente a importância de um item para um dado usuário) ou uma *recomendação*, expressa em uma lista de N itens, na qual o usuário deve gostar da maioria.

3.3.3 Identificação do usuário

Segundo Reategui e Cazella (2005), para que seja possível recomendar a um usuário de maneira personalizada, é vital se ter conhecimento sobre quem ele é. Logo, torna-se necessário identificar o usuário no momento em que ele acessa o sistema. Na internet, é proposto que isso seja realizado de duas formas:

- **Identificação no servidor:** disponibiliza ao usuário uma área de cadastro com informações pessoais e essas informações ficam armazenadas em um banco de dados no servidor. Este mecanismo provê maior precisão na identificação do usuário conectado.
- **Identificação no cliente:** normalmente feito através do uso de *cookies*, onde as informações do usuário que serão utilizadas pelo sistema de recomendação são armazenadas no navegador do mesmo. Este método assume que a máquina é sempre utilizada pela mesma pessoa. É um mecanismo mais simples do que a identificação no servidor, porém menos confiável, pois pessoas com dois perfis de preferências completamente distintos podem utilizar o sistema através da mesma máquina e, neste caso, serão considerados apenas um usuário.

Assim que o usuário é identificado, é possível coletar dados explícitos e implícitos sobre ele e, desta forma, gerar e manter o seu perfil de preferências.

3.3.4 Técnicas de recomendação

Existem diversas técnicas de recomendação que podem ser aplicadas. Burke (2002) define cinco principais, elas são descritas a seguir:

3.3.4.1 *Recomendação colaborativa*

É provavelmente a técnica mais utilizada e a que apresenta as tecnologias mais consolidadas. Sistemas de recomendação colaborativos buscam prever a utilidade de um item a um usuário específico utilizando como base itens já avaliados por outros usuários (BURKE, 2002).

Reategui e Cazella (2005) afirmam que a essência desta técnica está na troca de experiências entre as pessoas que possuem interesses em comum. Usando as informações de outros usuários é possível recomendar itens mesmo que esses não sejam semelhantes aos já avaliados pelo usuário.

Como os sistemas de recomendação colaborativos dependem apenas das avaliações feitas, o usuário precisa avaliar muitos itens até que o sistema consiga definir um perfil de preferências e, até que isso aconteça, as recomendações não serão muito precisas. Por outro lado, a especialização do perfil do usuário acaba fazendo com que só sejam feitas recomendações a itens já conhecidos pelo usuário, ou seja, um usuário que não frequente festas eletrônicas, nunca receberá uma recomendação para um evento desse tipo.

3.3.4.2 *Recomendação demográfica*

A recomendação demográfica busca categorizar os usuários, tendo como base alguns atributos pessoais extraído através de preenchimento de formulários (como sexo, idade, endereço, escolaridade) e realiza recomendações com base em classes demográficas definidas. O perfil do usuário é criado através da classificação dos usuários em estereótipos que representam as características de uma classe de usuários (BURKE, 2002).

3.3.4.3 *Recomendação baseada em conteúdo*

Na recomendação baseada em conteúdo busca-se fazer uma recomendação em função do conteúdo de determinado item. Sistemas como *Netflix.com* aplicam esta técnica para recomendar seus filmes, utilizando como conteúdo de análise características como o diretor do filme, os atores envolvidos, o gênero do filme, entre outros.

Para que a recomendação seja possível é necessário que se encontre associações entre os itens. Por exemplo, um filme possui diversos atores e esses atores estão presentes também em outros filmes, sendo assim, pode-se inferir que usuários que gostaram de um filme com um determinado ator podem querer assistir seus outros trabalhos disponíveis.

Na figura 3.3 se vê que, devido ao fato do usuário ter assistido dois filmes baseados em livros, o sistema *Netflix.com* passa a recomendar novos filmes que possuem um conteúdo semelhante, ou seja, também são filmes com histórias baseadas em livros.



Figura 3.3: Exemplo de recomendação baseada em conteúdo utilizada no *Netflix.com*

3.3.4.4 Recomendação baseada em utilidade

Sistemas de recomendação baseados em utilidade realizam um cálculo da utilidade de cada item disponível para determinado usuário. Segundo Burke (2002), o grande problema está em como desenvolver uma função de utilidade que supra as necessidades de cada usuário especificamente. As informações do perfil do usuário formam uma função de utilidade e o sistema emprega a técnica de satisfação de restrições na busca da obtenção do melhor resultado.

No cálculo de utilidade, além de considerar as características dos itens cadastrados, o sistema pode acrescentar variáveis que não estão diretamente relacionadas aos mesmos como, por exemplo, prazo de entrega, localização do produto, forma de pagamento, entre outros (BURKE, 2002).

Como a técnica de recomendação baseada em utilidade não se utiliza do histórico de avaliações dos usuários, ela não apresenta a limitação, presente na recomendação colaborativa, de que novos usuários cadastrados no sistema não recebem recomendações precisas até possuírem um número definido de avaliações. Entretanto, exige-se um grande nível de interação com o sistema para que ele consiga construir uma função de utilidade que precise as preferências do usuário com sucesso (BURKE, 2002).

3.3.4.5 Recomendação baseada em conhecimento

A recomendação baseada em conhecimento é feita através de “inferências das preferências do usuário e suas necessidades através de conhecimento estruturado de forma funcional” (REATEGUI; CAZELLA, 2005, p. 3). O sistema utiliza o conhecimento sobre o usuário para inferir quais devem ser as suas preferências.

Adomavicius e Tuzhilin (2005) apontam como vantagem, no uso desta técnica, o aumento de precisão e a falta das limitações intrínsecas dos sistemas colaborativos e baseados em conteúdo. Entretanto, o principal problema em sistemas baseados em conhecimento é, justamente, adquirir o conhecimento.

3.3.4.6 Sistemas de recomendação híbridos

Sistemas de recomendação híbridos nada mais são do que a combinação de mais de uma técnica de recomendação. Seu objetivo é explorar os pontos fortes de cada técnica

a fim de minimizar as limitações apresentadas no uso de apenas uma técnica (BURKE, 2002).

Reategui e Cazella (2005) definem sistemas de recomendação híbridos como a combinação da recomendação colaborativa e da recomendação baseada em conteúdo, onde os pontos fortes de cada uma dessas técnicas eliminam as fraquezas que elas apresentam isoladamente, como pode ser visto na figura 3.4.

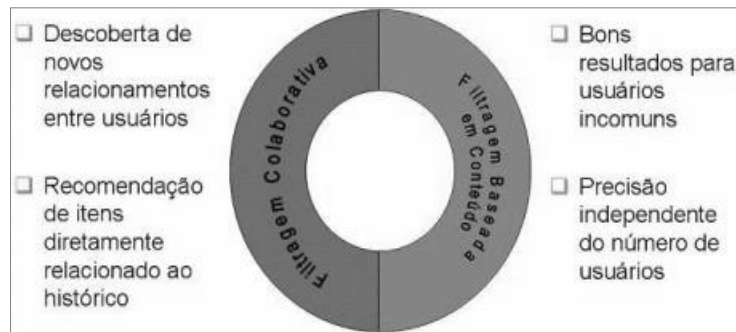


Figura 3.4: Recomendação híbrida.
Fonte: Reategui e Cazella (2005, p. 20)

3.4 Coletores de Documentos

Os coletores de documentos (ou *Web crawlers*), são softwares desenvolvidos com o intuito de percorrerem estruturas de *hyperlinks* de páginas *Web* a fim de armazenar automaticamente os dados das páginas referenciadas por estes links para uma posterior análise e utilização da informação obtida (LIU, 2011).

Segundo Liu (2011), os coletores de documentos podem ser classificados como *universal crawlers*, onde todas as páginas devem ser armazenadas (sem levar em consideração os seus conteúdos) e *preferential crawlers*, que somente armazena as páginas relacionadas a certas categorias e assuntos.

3.5 Model-View-Controller

O conceito de *Model-View-Controller* (Modelo-Visão-Controlador, em português) foi introduzido por Krasner e Pope (1988) como um modelo para estabelecer um conjunto de componentes necessários para o processo de desenvolvimento de *software* e para prover aos desenvolvedores um conjunto de componentes do sistema que tornasse fácil se desenvolver aplicações gráficas que pudessem ser reutilizáveis.

Para desenvolver uma aplicação utilizando arquitetura MVC deve-se fatorar o sistema em três camadas: uma camada que represente um modelo de domínio específico da aplicação, denominada **modelo**; uma camada que seja responsável pela forma como os dados são apresentado ao usuário final, denominada **visão**; e uma camada que trate da interação do usuário com a **visão** e o **modelo**, denominada **controlador**. (Krasner e Pope, 1988)

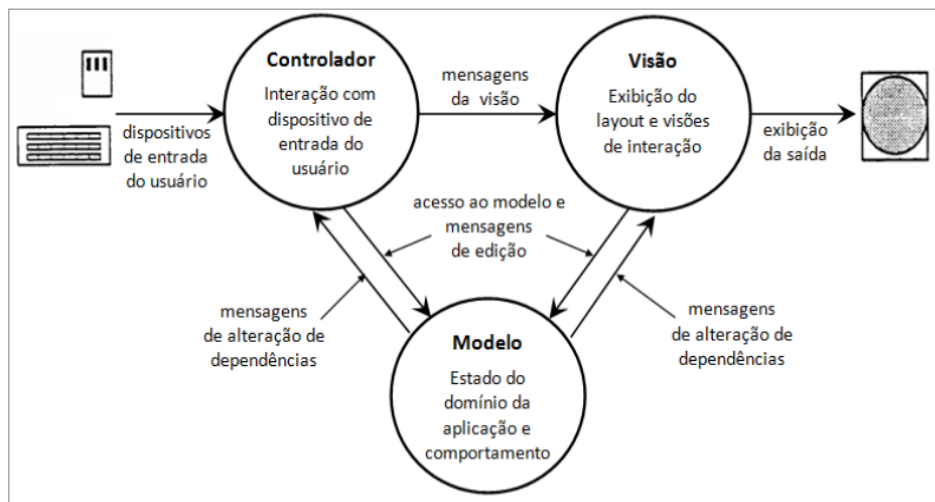


Figura 3.5: Visão geral do modelo MVC

Fonte: imagem traduzida e adaptada de Krasner e Pope (1988 apud DA SILVA, 2012 p. 30).

3.6 Ferramentas utilizadas

Esta seção apresentará as ferramentas utilizadas no desenvolvimento do sistema proposto, bem como a motivação para o seu uso.

3.6.1 Web

O sistema será desenvolvido para o ambiente *Web*, pois conforme apresentado no capítulo 1, quase toda a população de Porto Alegre já está conectada à internet, facilitando assim o acesso ao sistema. Além disso, o ambiente e as ferramentas utilizadas na visualização do conteúdo mantém a experiência de uso do sistema através de *smartphones* e *tablets*.

3.6.2 MySQL

Para a persistência dos dados foi utilizado o MySQL (<http://www.mysql.com/>), que é um sistema de gerenciamento de banco de dados relacional. Sua escolha se deu devido ao fato de ser um *software* gratuito e por já possuir integração nativa com o *framework* Yii.

Sua utilização foi feita juntamente com a ferramenta MySQL Workbench (<http://www.mysql.com/products/workbench/>) que provê uma interface gráfica ao MySQL, disponibilizada também de forma gratuita.

3.6.3 Javascript

O Javascript é uma linguagem de programação interpretada que descreve funções com o objetivo de interagir com o Modelo de Objeto de Documentos (DOM) de uma página *Web*. Essa interação provida pela linguagem permite a detecção de diversas ações realizadas pelo usuário, possibilitando assim uma melhor experiência na navegação.

Uma de suas vantagens é ser executada no *browser* do usuário, sem a necessidade de uma comunicação com um servidor remoto, trazendo assim um melhor tempo de resposta a algumas aplicações no sistema.

Atualmente, é largamente utilizada na técnica AJAX (*Asynchronous Javascript and XML*), técnica esta que implementa a criação de requisições assíncronas onde a requisição ao servidor e sua resposta são tratadas pelo sistema de forma que a navegação do usuário não seja prejudicada. Enquanto o sistema processa a requisição, o usuário pode continuar navegando e, ao terminar o processamento, o servidor responde trazendo apenas os dados necessários para a aplicação, que são tratados e exibidos ao usuário. É importante ressaltar que, em uma requisição normal, o servidor responderia trazendo todo o documento HTML para a exibição.

Diversas bibliotecas foram desenvolvidas a partir do Javascript. Uma delas, talvez uma das mais famosas, é a jQuery³. Essa biblioteca tem como principais objetivos a resolução da incompatibilidade do Javascript entre os diferentes navegadores existentes e prover uma sintaxe que torne o uso da linguagem mais simples e intuitivo. Além disso, a jQuery permite a criação de *plugins* sobre ela, possibilitando aos desenvolvedores criarem camadas de abstração para interações de mais baixo nível.

Neste trabalho, foram utilizados, além do Javascript e do jQuery, os *plugins* *isotope.js*⁴ e *isotope-perfectmasonry.js*⁵ na criação do *layout* empregando o conceito de *treemapping*.

3.6.4 HTML

O HTML é uma linguagem de marcação utilizada na apresentação e formatação de conteúdo de páginas *Web*. Utiliza *tags* (palavras entre parênteses angulares) como sintaxe de formatação do documento.

Um elemento HTML é formado por uma *tag*, atributos, valores e filhos (outros elementos HTML ou texto). Atualmente, todos os principais *browsers* interpretam documentos HTML.

3.6.4.1 DOM – Modelo de Objetos para Documentos

O DOM (*Document Object Model*) é especificado pela W3C e representa a forma como as *tags* de documentos HTML são organizadas e lidas por navegadores *Web*. Uma vez indexadas, estas marcações se transformam em uma árvore de nodos que mapeia a sua ordem e estrutura com relação aos elementos do documento, que pode ser visto na figura 3.6. O Modelo de Objetos para Documentos oferece, ainda, uma interface de programação de aplicações (API) para a manipulação dos nodos.

Segundo Tittel, o DOM “permite a programas e *scripts* acessar e atualizar dinamicamente o conteúdo, a estrutura e o estilo de documentos.” (TITTEL, 2003, p. 83-84)

³ Disponível em: <http://jquery.com/>

⁴ Disponível em: <http://isotope.metafizzy.co/>

⁵ Disponível em: <https://github.com/zonear/isotope-perfectmasonry>

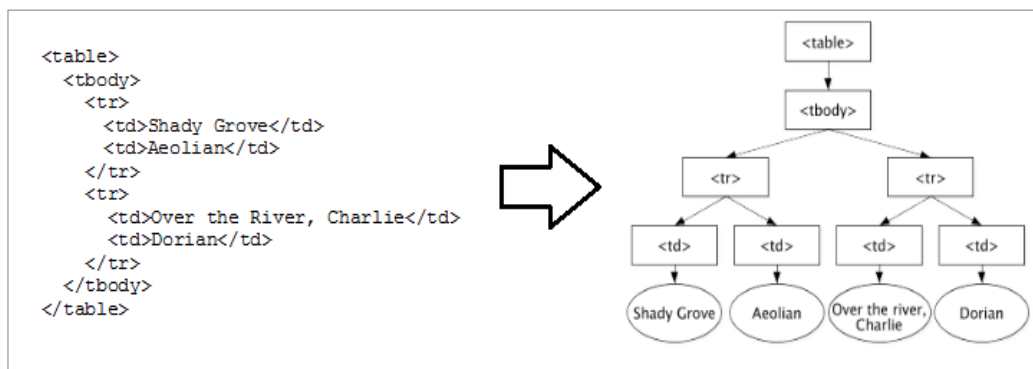


Figura 3.6: Representação da árvore DOM a partir do código HTML de uma tabela.

Fonte: Document Object Model (DOM) Level 3 Core Specification (W3C, 2013)

3.6.5 CSS

O CSS (*Cascade Style Sheet*) é uma linguagem que tem como objetivo definir o estilo da apresentação de documentos escritos com linguagens de marcação, como o HTML, provendo a separação entre o formato e o conteúdo do documento.

Um arquivo CSS consiste em uma folha de estilos que define a forma de apresentação de um documento, identificando os elementos nos quais serão aplicados o estilo através de *tags*. Isso facilita a alteração do *layout* de um sistema pois, quando necessário for, só é preciso alterar em um local o que se propagará para todo o sistema.

No desenvolvimento do sistema proposto foi utilizado o Bootstrap⁶, um *framework* para desenvolvimento de interfaces disponibilizado gratuitamente. Uma das suas principais características está no uso de CSS responsivo, que garante a exibição do conteúdo da mesma forma em diferentes *browsers* e dispositivos (como *tablets* e *smartphones*). Além disso, por estar sendo utilizado atualmente em diversos sites, provê uma navegação mais intuitiva ao usuário, mesmo em seu primeiro acesso.

3.6.6 Framework Yii

O PHP vem sendo usado para o desenvolvimento de websites dinâmicos há mais de 10 anos. Inicialmente, todos os websites PHP eram escritos com código PHP intercalado com HTML na mesma página. Isto funcionava muito bem, pois fornecia uma resposta rápida, e para scripts simples aparentemente isso é tudo o que era necessário. O PHP tornou-se mais popular nas versões 3 e 4, e, inevitavelmente, aplicações cada vez maiores passaram a ser escritas em PHP. Rapidamente tornou-se óbvio que misturar código PHP com HTML não era uma solução de longo prazo para websites maiores. Os problemas tornaram-se evidentes: manutenção e capacidade de extensão. Embora o PHP misturado com HTML permita resultados extremamente rápidos, a longo prazo fica difícil continuar atualizando o website. (ALLEN; LO; BROWN, 2009, p. 3)

Winesett (2012) afirma que a utilização de *frameworks* de desenvolvimento *Web* impulsiona a criação de novas aplicações pelo fato de disponibilizar ao desenvolvedor o núcleo do sistema e diversas características comuns em aplicações *Web*. “Um *framework* moderno, flexível e extensível se tornou uma ferramenta essencial para o

⁶ Disponível em: <http://getbootstrap.com/2.3.2/>

desenvolvedor *Web*, tanto quanto a linguagem de programação por si só." (WINESETT, 2012, p. 9)

O *framework* utilizado para o desenvolvimento do "O que tá valendo?" foi o Yii, na versão 1.1.13⁷. A escolha do Yii se deveu ao fato de ser um *software* livre, disponível em versão estável, licenciado pelo *BSD License* (que consente ao usuário o direito de uso gratuito do *software*, inclusive com fins comerciais), contando com uma documentação disponível bastante completa e uma comunidade de usuários ativa.

Diferentemente de alguns *frameworks* de desenvolvimento *Web*, o Yii exige apenas um servidor *Web* com suporte à PHP 5.1.0, não necessitando instalação de nenhum serviço exclusivo para a sua execução. Para utilizá-lo, basta descompactar os seus arquivos no servidor e acessá-lo através do navegador.

O *framework* possui o conceito de extensão, onde as *features* disponíveis podem ser incrementadas através do uso de componentes desenvolvidos pela comunidade e disponibilizados através do site oficial. Dentre uma das *features* disponíveis no pacote padrão do Yii, um destaque deve ser dado ao gerador automático de código, chamado *Gii*. Com ele, é possível gerar modelos, controladores e formulários com apenas alguns cliques do *mouse*.

Ao longo deste trabalho serão descritas mais características do Yii e sua utilização.

3.6.6.1 A implementação do MVC utilizada no framework Yii

Além das três camadas definidas na arquitetura MVC (modelo, visão e controlador), o *framework* Yii acrescenta, em sua implementação, um controlador específico, chamado *Application*, responsável por encaminhar as requisições do usuário para o controlador apropriado. Também acrescenta à arquitetura uma camada denominada *widget*. Um *widget* é um componente autocontido utilizado, normalmente, para exibir elementos complexos na interface, auxiliando à reusabilidade de código (YII, 2013).

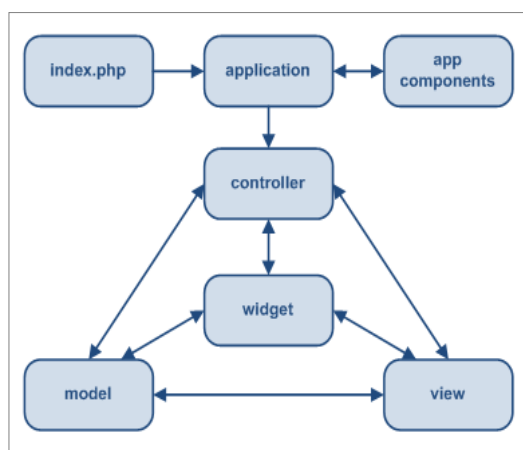


Figura 3.7: Implementação do modelo MVC feita pelo *framework* Yii.

Fonte: The Definitive Guide (YII, 2013)

A figura 3.8 apresenta o *workflow* típico de uma aplicação MVC no Yii. O usuário faz uma requisição para uma dada *URL*, o servidor *Web* cria uma instância de *Application* que determina e cria uma instância para o controlador apropriado. Esse

⁷ Disponível em: <http://www.yiiframework.com/>

controlador determina a ação a ser executada. A ação lê dados do modelo e, após, renderiza uma visão com esses dados. A visão lê e exibe os atributos do modelo, executa alguns *widgets* (caso existam) e embute em um *layout*, o resultado do processamento. (YII, 2013)

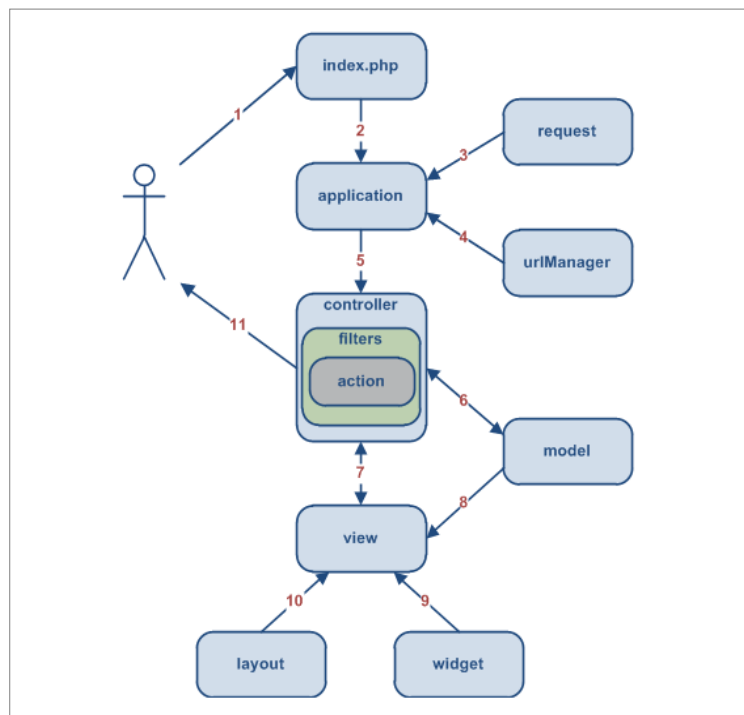


Figura 3.8: Representação do *workflow* em uma aplicação utilizando o *framework* Yii.

Fonte: The Definitive Guide (YII, 2013)

3.6.7 Facebook API

O *Facebook* disponibiliza gratuitamente, na seção de desenvolvimento em seu site oficial (<http://developers.facebook.com>), uma interface de programação de aplicativos (API). Ela provê uma série de funções para a criação de aplicativos utilizando informações da rede social e de seus usuários.

Uma das funcionalidades providas pela API é a possibilidade da utilização de um perfil no *Facebook* para o cadastro e autenticação de usuários em aplicativos externos. Para tanto, é necessário criar no *Facebook*, em sua seção de desenvolvimento, um aplicativo que será responsável por fazer requisições à rede social, entregando as respostas ao sistema que o utiliza. Na criação desse aplicativo, declara-se quais informações do usuário ele desejará ter acesso.

No sistema proposto, foi utilizada uma extensão do *framework* chamada *Facebook-opengraph* que realiza essa troca de mensagens entre o *framework* Yii e a aplicação criada no *Facebook* através de uma caixa preta.

4 SISTEMA “O QUE TÁ VALENDO?”

Nos capítulos anteriores foram discutidas as características de um sistema de recomendação e vistos aspectos relevantes do funcionamento dos principais sistemas de divulgação de eventos atuais.

Neste capítulo será proposto um sistema com o objetivo de obter eventos automaticamente de grandes portais (através do conceito de coletores de documentos), centralizando-os e divulgando-os para os usuários de maneira personalizada, utilizando os conceitos de recomendação. Primeiramente, serão apresentadas as funcionalidades necessárias ao sistema e, em seguida, será apresentada a arquitetura do sistema, descrevendo com detalhes o funcionamento de cada um de seus componentes.

4.1 Projeto do sistema

Nesta seção serão definidos os requisitos necessários para a criação do sistema de recomendação de eventos proposto.

4.1.1 Requisitos do sistema

A engenharia de *software* define como requisitos os objetivos que um sistema deve atender e as restrições que devem ser satisfeitas. Em outras palavras, ela define aquilo que o sistema deve fazer e como se comportar em determinadas situações.

Os requisitos funcionais necessários ao sistema proposto serão apresentados através de *user stories*. Segundo Mike Cohn, em seu site oficial⁸, uma *user story* é definida como uma descrição simples de uma *feature* do projeto descrita sobre o ponto de vista da pessoa que irá utilizá-la. Normalmente, é feita sobre o *template*: “Como um <tipo do usuário>, eu quero <descrição do objetivo> para que <descrição do motivo>.”

4.1.1.1 User Stories

Nesta seção serão descritos os requisitos do sistema, através de *user stories*. A modelagem proposta prevê três perfis de usuário: o usuário não-registrado (que não está registrado no sistema), usuário registrado (que está registrado sistema) e usuário administrador (além de ser um usuário registrado, possui o atributo de administrador). Quando citado usuário, sem especificar um tipo, corresponde a uma *feature* disponível aos três perfis. Cada item abaixo corresponde a uma *user story*:

- História 1: “Como usuário, eu quero poder visualizar os eventos atualmente cadastrados, para que possa me informar sobre a agenda cultural.”

⁸ <http://www.mountangoatsoftware.com/agile/user-stories>

- História 2: “Como usuário administrador, eu quero gerenciar os eventos (incluir, editar e excluir), para que possa administrar as informações disponíveis para os usuários do sistema.”
- História 3: “Como usuário administrador, eu quero gerenciar os locais (incluir, editar e excluir), para que possa administrar as informações disponíveis para os usuários do sistema.”
- História 4: “Como usuário administrador, eu quero gerenciar os usuários do sistema (editar e excluir), para que possa administrar os usuários cadastrados no sistema.”
- História 5: “Como usuário administrador, eu quero buscar e adicionar novos eventos automaticamente, através de sites previamente definidos no sistema, para que simplifique o processo de adicionar novos eventos ao sistema.”
- História 6: “Como usuário, eu quero visualizar os eventos de maneira personalizada, de forma que o sistema recomende aqueles que me serão mais interessantes para que facilite a minha escolha.”
- História 7: “Como um usuário registrado, eu quero poder avaliar os locais cadastrados no sistema, onde ocorrem os eventos, para que possa informar quais possuo preferência.”
- História 8: “Como um usuário, eu quero poder visualizar os locais cadastrados no sistema, onde ocorrem os eventos, para que possa conhecer novos locais.”
- História 9: “Como um usuário, eu quero poder visualizar a média das avaliações dos locais cadastrados no sistema, onde ocorrem os eventos, para que possa saber o que os outros usuários pensam sobre os locais.”
- História 10: “Como um usuário, eu quero poder acessar a página oficial de um local, para que possa obter mais informações sobre ele.”
- História 11: “Como um usuário, eu quero visualizar o cartaz de divulgação de um evento ao selecioná-lo, para que possa obter mais informações sobre ele.”
- História 12: “Como um usuário, eu quero uma opção que me redirecione à página original de um evento, para que eu possa obter mais informações sobre ele.”
- História 13: “Como um usuário, eu quero me cadastrar no sistema utilizando meu perfil do *Facebook*, para que não precise preencher formulários e memorizar uma nova senha.”
- História 14: “Como um usuário, eu quero me logar no sistema utilizando meu perfil do *Facebook*, para simplificar o processo de autenticação.”
- História 15: “Como um usuário, eu quero poder filtrar os eventos disponíveis no sistema pelo seu tipo, para que possa visualizar as opções disponíveis de diferentes maneiras.”

4.1.1.2 Requisitos Não-Funcionais

Segundo Sommerville (2011), um requisito não-funcional reflete uma necessidade do sistema não relacionada diretamente ao serviço que ele entrega aos seus usuários. Pode ser considerado um requisito não-funcional a confiabilidade esperada, o tempo de resposta exigido, entre outros, por exemplo.

Os requisitos abaixo listados, definem pontos de projeto, como as plataformas, as ferramentas e tecnologias utilizadas para o desenvolvimento do sistema.

- O desenvolvimento *server-side* deve ser feito em PHP, por ser uma linguagem estabelecida e com uma grande comunidade ativa, com o auxílio do *framework* Yii, a fim de auxiliar na padronização de código e manutenção do sistema.
- O desenvolvimento *client-side* deve ser feito em HTML, CSS e Javascript por serem tecnologias sem custo e com suporte a grande maioria dos navegadores atuais.
- O *layout* do sistema deve utilizar, sempre que possível, componentes do Bootstrap, a fim de se manter um padrão na exibição dos dados.
- O sistema deve rodar em um servidor Apache.
- O sistema deve utilizar o banco de dados MySQL por ser uma tecnologia sem custo e com integração nativa com o *framework* Yii.
- O sistema deve possuir uma interface de apresentação de fácil utilização e rápida aprendizagem.

4.2 Arquitetura do sistema

Nesta seção será apresentada a arquitetura do sistema, bem como será explicado o funcionamento de cada um dos seus componentes.

A arquitetura do sistema, conforme visto na figura 4.1, tem no *framework* Yii, definido no capítulo 3, a sua base de funcionamento, sendo ele o responsável por toda a interação entre os diversos componentes presentes no sistema. O componente de “*Web crawling*” tem como objetivo extrair informações de sites definidos e adicionar novos eventos ao sistema. O componente de “Autenticação com o *Facebook*” é responsável por realizar o cadastro e *login* de usuários utilizando, para isso, os dados do perfil da rede social, através do uso da *Facebook* API (definida na seção 3.6.2). O componente de “Banco de Dados” é responsável por manter a persistência dos dados que serão utilizados pelo componente de “Recomendação”, que por sua vez realiza a busca do histórico de uso do sistema e, através de diversos critérios e cálculos, tenta priorizar os eventos que serão mais interessantes àquele usuário específico.

A seguir será feita uma descrição detalhada do funcionamento de cada um dos componentes do sistema.

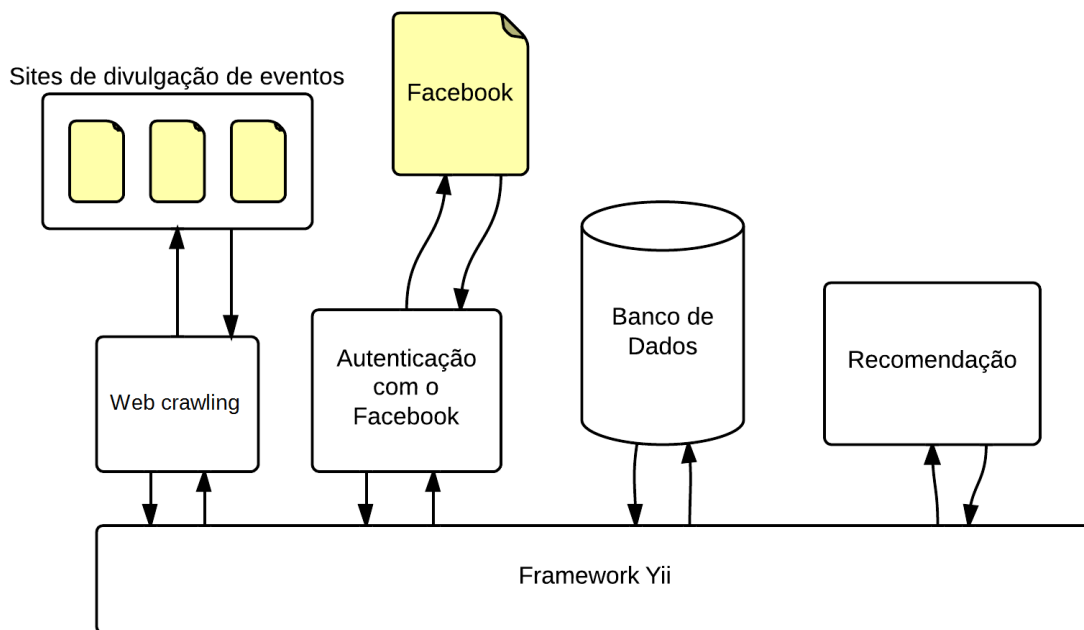


Figura 4.1: Visão geral da arquitetura do sistema.

Fonte: imagem criada pelo autor.

4.2.1 Módulo de Web crawling

Conforme definido no capítulo 3, coletores de documentos, ou *crawlers*, consistem na armazenagem de dados de páginas da *Web*. No sistema proposto, o módulo de *Web crawling* tem como objetivo extrair a informação sobre eventos agendados em sites pré-definidos.

Esse processo é implementado no sistema com o auxílio da extensão SimpleHTMLDOM⁹, uma classe PHP que implementa um *parser* para o Modelo de Objetos para Documentos, permitindo, com isso, a busca de conteúdo através da pesquisa por *tags* HTML.

Cada site presente na internet possui um documento HTML único. Entretanto, normalmente é possível identificar padrões que se repetem em duas páginas diferentes de um mesmo site. As figuras 4.2 e 4.3 exibem trechos de código HTML de eventos diferentes do site “Curtindo POA”, descrito no capítulo 2. Apesar de se tratar de dois eventos diferentes, é possível analisar que nos dois casos o *link* para a imagem de divulgação de cada evento está dentro de uma *tag* `<div>` com o atributo *class* igual a *photo*. Sendo assim, é possível assumir que, para o site “Curtindo POA”, os *links* das imagens de exibição dos eventos estarão disponíveis sempre sob esse padrão.

⁹ Disponível em: <http://www.yiiframework.com/extension/simplehtmldom>

```

<div class="photo">

<p>Classificação: 14 anos</p>
<p class="url"><a href="http://www.opinioao.com.br/eventos/jota-quest/" target='_blank'><i class='ico link'></i> Acessar link do evento</a></p>
<p class="url"></a></p>
<p class="url"></a></p>

<div class="author">
<h3>Indicado por</h3>
<div class="content">
<a href="https://www.facebook.com/camila.surian/" title="Camila Surian" target="_blank">
Camila Surian </a>
</div>
</div>
</div>
</div>

```

Figura 4.2: Trecho do código HTML de um evento no site “Curtindo POA”.

```

<div class="photo">

<p>Gravação do Happy Hour Devassa da Ipanema FM, nesta quinta às 19h no Pão com Linguíça. (Rua Santa Catarina, 390)</p>
<p class="url"><a href="https://www.facebook.com/TonhoCroccoNoNoKingRecords" target='_blank'><i class='ico link'></i> Acessar link do evento</a></p>
<p class="url"></a></p>
<p class="url"></a></p>

<div class="author">
<h3>Indicado por</h3>
<div class="content">
<a href="https://www.facebook.com/camila.surian/" title="Camila Surian" target="_blank">
Camila Surian </a>
</div>
</div>
</div>
</div>

```

Figura 4.3: Trecho do código HTML de outro evento no site “Curtindo POA”.

Para realizar a obtenção de dados utilizando a técnica proposta neste sistema, é preciso identificar as *tags* que contém as informações necessárias para a definição de um evento no sistema (definido no capítulo 3).

Internamente, os sites em que serão realizados esse processo são representados, cada um, por um modelo. O modelo possui um método principal chamado *minera*, que cria uma instância da extensão SimpleHTMLDOM que, por sua vez, é aplicada a um método para retornar um objeto contendo o DOM de uma *URL* solicitada.

Conforme se pode analisar na figura 4.4, tendo o objeto que possui o DOM (representado no código por *\$html*), é possível realizar a busca das *tags* HTML que contém as informações que são necessárias para o sistema. No exemplo apresentado, os *links* (que são representados em HTML utilizando *tag* `<a>`) dos eventos agendados no site “Curtindo POA” deverão estar dentro de uma *tag* `<h4>` que, por sua vez, estarão dentro de uma *tag* `<div>` com o atributo *class* igual a *events*. A informação que se procura e, os elementos utilizados na sua busca podem ser vistos no trecho do código HTML da página citada, destacados na figura 4.5. Posteriormente, para a continuidade do processo, é obtido o DOM de cada um dos *links* a fim de se obter as demais informações necessárias.

```

public function minera()
{
    Yii::import('application.extensions.SimpleHTMLDOM.SimpleHTMLDOM');

    $events = array();

    $simpleHTML = new SimpleHTMLDOM;
    $url = "http://curtindopoa.com.br/eventos";
    $html = $simpleHTML->file_get_html($url);

    foreach($html->find('div[class=events] h4 a') as $k => $element):
        $events[$k]["link"] = "http://www.curtindopoa.com.br".$element->href;
    endforeach;

    for($i=0; $i<count($events); $i++):
        $html = $simpleHTML->file_get_html($events[$i]["link"]);
        $events[$i]["name"] = $this->getEventName($html);
        $events[$i]["date"] = $this->getEventDate($html);
        $events[$i]["place"] = $this->getEventPlace($html);
        $events[$i]["image"] = $this->getEventImage($html);
    endfor;

    return $events;
}

```

Figura 4.4: Exemplo do algoritmo de obtenção de dados no site “Curtindo POA”.

```

<div class="evento">
<ul class="clearfix event-list">
<li class="event musica">
<span class="start-hour">20/11 16:00</span>
<div class="content">
<a href="/eventos/4361/sarau-york"></a> <div class="detail">
<h4><a href="/eventos/4361/sarau-york">SARAU YORK</a></h4>
<div class="social-stats">
<span class="likes">13</span>
<span class="joins">1</span>
</div>
</div>
</li>

```

Figura 4.5: Trecho de código HTML do site “Curtindo POA”.

A saída do módulo de *Web crawling* é um *array* contendo as informações de todos os eventos extraídos, que são enviados a uma interface de gerência para a aceitação, ou não, de cada novo evento obtido.

O processo aqui demonstrado, apesar de simples, é bastante eficaz para a obtenção de dados automaticamente. Entretanto, por ser completamente dependente dos padrões definidos na busca, caso haja uma alteração na maneira em que os dados são exibidos no HTML do site fonte, torna-se necessária a correção das *tags* de busca, para a volta do funcionamento do processo para o referido site.

4.2.2 Módulo de identificação e integração ao *Facebook*

Seguindo uma das necessidades de um sistema de recomendação, a identificação do usuário (definido no capítulo 3), esse módulo tem como objetivo criar o cadastro de um novo usuário (ou realizar a identificação de um usuário prévio) no sistema, utilizando um perfil do *Facebook*.

Atualmente, essa é uma técnica empregada em diversos sistemas na *Web* pela praticidade (pois o cadastro é feito com apenas alguns cliques, sem o preenchimento de formulários) e pela possibilidade da obtenção, por parte do sistema, de informações que o usuário disponibiliza em sua rede social.

O *framework* Yii possui uma extensão chamada *Facebook-opengraph*¹⁰ que encapsula a API do *Facebook*. Para a sua utilização é necessário somente a criação e definição de uma aplicação na seção de desenvolvimento do *Facebook* (conforme citado no capítulo 3). A extensão utiliza a aplicação definida para realizar a comunicação entre o *Facebook* e o sistema. Quando o usuário realiza o *login* pela primeira vez no sistema proposto, utilizando o perfil do *Facebook*, essa aplicação é instalada em seu perfil.

O *Facebook-opengraph* encapsula diversas funcionalidades úteis, entre elas está um *widget*, como visto na figura 4.6, que realiza o processo de autenticação na rede social e renderiza um botão de *login* para a exibição.

```
<?php $this->widget('ext.yii-facebook-opengraph.plugins.LoginButton', array(
    'show_faces' => false,
    'width' => '100px',
    'scope' => 'email, user_likes',
    'text' => 'Login via facebook',
    'on_login' => 'window.location.replace("http://localhost/TCC/index.php/site/login");'
)); ?>
```

Figura 4.6: *Widget* de autenticação com o *Facebook*.

O processo completo de autenticação e cadastro de um usuário no sistema proposto é dividido em duas etapas: a que ocorre no *Facebook* e a implementada pelo sistema.

Quando o usuário clica no botão para efetuar o *login*, o processo de autenticação é iniciado na rede social e, caso seja bem sucedido, a extensão cria um objeto com os dados do *Facebook* do usuário e chama a ação de autenticação definida pelo sistema. O processo de autenticação realizado no *Facebook* pela extensão é uma caixa preta, maiores informações podem ser obtidas em <http://developers.facebook.com>.

O sistema armazena somente uma informação do usuário: o *hash* de seu identificador único na rede social. O Yii possui um componente chamado *UserIdentity* que é responsável pela autenticação de usuários no sistema. O componente realiza a busca do *hash* do identificador único do usuário no *Facebook* na base de dados e, caso não seja encontrado é adicionado um novo registro à base de dados referente ao novo usuário. Posteriormente, é criada uma instância de *UserIdentity* para esse novo usuário. Caso contrário, somente é instanciada a classe *UserIdentity* para o *hash* dado.

O Yii implementa o *design pattern GoF Singleton* no seu método nativo de *login*. Ao utilizar-se esse método (que recebe um *UserIdentity* e um tempo de duração da credencial) o sistema realiza a autenticação, e caso haja sucesso, cria uma instância do usuário, garantindo que em todos os outros pontos do sistema onde o objeto usuário for instanciado ele será recuperado de sua instância original até que seja chamado o método, também nativo do *framework*, de *logout* ou termine o tempo definido de duração da credencial.

4.2.3 Banco de Dados

A recomendação de eventos é baseada na identificação, no histórico de navegação e na avaliação dos usuários do sistema. Para tanto, torna-se necessária a persistência

¹⁰ Disponível em: <http://www.yiiframework.com/extension/facebook-opengraph/>

desses dados, bem como o simples armazenamento das informações de eventos e locais que são exibidos aos usuários.

Conforme definido, foi utilizado o MySQL para o gerenciamento da base de dados. A figura 4.7 apresenta o diagrama Entidade-Relacionamento do sistema proposto.

As entidades e seus atributos são descritos nas subseções seguintes.

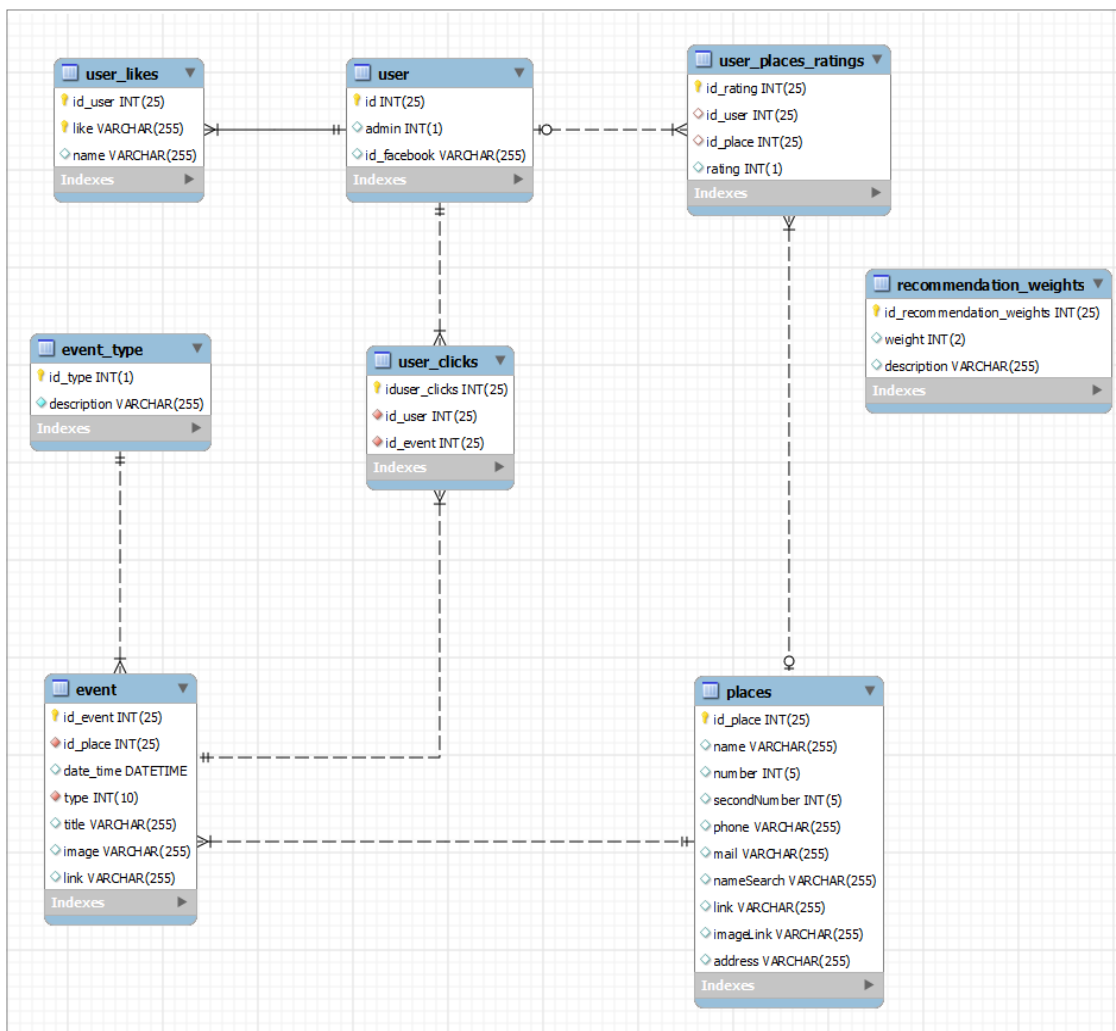


Figura 4.7: Diagrama Entidade-Relacionamento do sistema proposto.

4.2.3.1 Entidade User

A entidade *User* (usuário, em português) armazena o *hash* do identificador único de usuário no *Facebook* para os usuários cadastrados no sistema. Cada linha desta tabela possui, além disso, um identificador único (que identifica o usuário dentro do sistema proposto) e uma *flag* que determina se o usuário é um usuário administrador (1) ou um usuário registrado (0).

A tabela é construída dinamicamente a partir da entrada de novos usuários no sistema. A única forma possível de criar um novo usuário se dá através do botão “*Login via Facebook*”.

4.2.3.2 Entidade *User_Likes*

Ao realizar o primeiro *login* no sistema é feita uma busca das páginas que o usuário curtiu no *Facebook*. A entidade *User_likes* possui uma chave primária composta com o id do usuário (que é, também, uma chave estrangeira para a tabela *User*) e o identificador único da página no *Facebook*. O campo *name* armazena o nome da página, dado a ser utilizado na recomendação.

4.2.3.3 Entidade *User_Clicks*

Seguindo a ideia de obtenção de dados implícitos, descrito na seção 3.3.1, a tabela *User_Clicks* é responsável pela persistência de todas as interações de usuários com os eventos no sistema.

Uma linha da tabela corresponde a: um identificador único do clique, o identificador do usuário (0, caso não seja um usuário registrado) e o identificador do evento clicado. Novas entradas são incluídas toda vez que o usuário clicar para visualizar as informações de um evento nas diferentes interfaces, conforme será explicado no próximo capítulo.

4.2.3.4 Entidade *Event*

A entidade *Event* (evento, em português) armazena as informações dos eventos cadastrados no sistema. Cada evento é constituído de um identificador único, um identificador que se refere a um local cadastrado no sistema, uma data e hora, um identificador que se refere a um dos tipos de evento disponibilizados no sistema, um título, um link para uma imagem de divulgação e o *link* de onde foi obtido as suas informações.

4.2.3.5 Entidade *Event_Type*

O sistema atualmente dá suporte a três tipos de eventos: festas, shows e teatros. Cada linha na tabela possui um identificador do tipo do evento e a sua descrição.

A tabela *Event_Type* atualmente armazena somente essas três tuplas, entretanto a sua existência torna o sistema apto ao acréscimo do suporte a novos tipos de eventos, caso desejado.

4.2.3.6 Entidade *Places*

Somente são exibidos, aos usuários, eventos cujo local de acontecimento esteja cadastrado no sistema. A tabela *Places* (locais, em português) armazena as informações dos locais. Cada local possui um identificador único, um nome, um endereço, um número da casa, um complemento (caso seja necessário), um telefone, um e-mail para contato, o *link* do seu site oficial, um *link* para uma imagem de exibição e um conjunto de variações do nome do local.

Sites de divulgação diferentes podem utilizar mais de um nome para se referir ao mesmo local. Um exemplo disso é o “Auditório Araújo Vianna”, que é referenciado em alguns sistemas como “Oi Araújo Vianna”. O atributo *nameSearch* soluciona esse problema armazenando as variações de nomes relacionados ao mesmo local, separados por vírgula, campo este que é utilizado na verificação e busca de um local no sistema.

4.2.3.7 Entidade *User_Rating_Places*

Os usuários logados no sistema podem avaliar os locais cadastrados, informação esta utilizada como um dos parâmetros no cálculo de recomendação. Cada linha nesta tabela é composta por um identificador único, o identificador do usuário que avaliou, o identificador do local avaliado e uma avaliação numérica sendo 1 a pior nota e 5 a melhor nota.

4.2.3.8 Entidade *Recommendation_Weights*

O sistema se utiliza de diversos critérios para calcular a recomendação ao usuário. É possível definirmos pesos diferentes para cada uma desses critérios a fim de calibrarmos a recomendação conforme a necessidade.

Cada registro na tabela *Recommendation_Weights* possui um identificador, um nome para o critério e o seu peso, sendo 0 o menor valor e 10 o maior valor.

A alteração dos pesos somente é feita pelo administrador, em uma interface disponibilizada apenas para ele.

4.2.4 Módulo de recomendação

O módulo de recomendação é fruto de uma combinação de funcionalidades executadas pelo *framework* Yii e de consultas realizadas sobre o histórico de uso do sistema na base de dados.

4.2.4.1 A ideia

O módulo de recomendação foi desenvolvido baseado na ideia de que as pessoas costumam frequentar e se interessar por eventos que aconteçam em locais do seu agrado. Em outras palavras, o primeiro critério de escolha para a decisão de qual evento participar, dentre as opções disponíveis, é onde ele irá ocorrer.

O sistema proposto busca reconhecer quais são esses locais de maior interesse do usuário registrado, identificar quais são os locais onde está havendo maior incidência de procura por eventos em um contexto geral (mais acessados, levando em conta todos os usuários do sistema) e, a partir de um processo de comparação e cálculo entre essas e informações, recomendar os eventos que estarão acontecendo na cidade.

4.2.4.2 Captura de dados explícitos e implícitos

Seguindo a definição de sistemas de recomendação apresentada no capítulo 3, é necessário se obter dados do usuário na tentativa de capturar, através disto, os seus gostos e preferências. Para tal, foi implementada a captura de dados explícitos e implícitos no sistema proposto.

Os dados implícitos são capturados nas seguintes situações:

- No primeiro *login* do usuário no sistema, é realizado uma busca e armazenamento das páginas que ele curtiu no *Facebook*, como pode ser visto na figura 4.8. Esse processo é realizado através de um método, que o autor adicionou à extensão *Facebook-opengraph*, que realiza uma requisição das páginas curtidas pelo usuário na rede social diretamente à *Facebook* API, trata a resposta recebida e retorna um *array* com o identificador único das página que foram curtidas e o seu nome de exibição.

	id_user	like	name
	351	107767729245735	Engenheiros do Hawaii
	351	110617212300500	+44
	351	110715645608956	Eric Clapton Autobiography
	351	112182752127936	Tequila Baby
	351	112203125458831	That Thing You Do!

Figura 4.8: Exemplo de como a entidade *User_Likes* armazena a informação das páginas curtidas no *Facebook* por um novo usuário do sistema.

- Ao clicar para visualizar os detalhes de um evento. Toda vez que um usuário, seja ele registrado no sistema ou não, clica para obter mais informações de um evento (processo descrito no próximo capítulo) é feito, juntamente à requisição AJAX que obtém os dados do evento a serem exibidos, um novo registro na entidade *User_Clicks* informando qual o identificador do usuário (0, caso não seja um usuário registrado) e o identificador do evento acessado.

Os dados explícitos são capturados na seguinte situação:

- Ao avaliar um local. Conforme visto na figura 4.9 (e descrito no próximo capítulo), um usuário registrado pode avaliar os locais onde ocorrem os eventos cadastrados no sistema conforme o seu interesse sobre os mesmos, atribuindo de 1 (representando pouco interesse) a 5 estrelas (representando muito interesse).

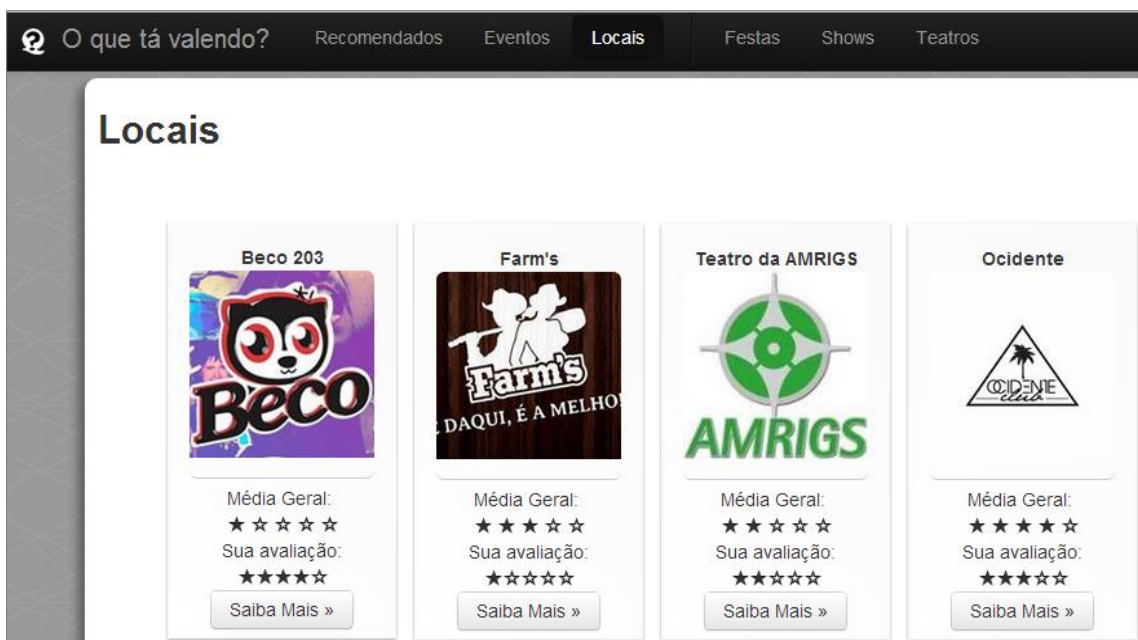


Figura 4.9: Exemplo de avaliações realizadas por um usuário registrado.

4.2.4.3 Avaliações e Pesos

Através das informações obtidas e dos dados disponíveis no sistema, é possível definirmos sete critérios que serão levados em consideração no cálculo da recomendação dos eventos, são eles:

- Eventos acessados por qualquer usuário, com qualquer data de acontecimento e data de acesso ao sistema.
- Eventos acessados por qualquer usuário que ocorrerão posteriormente à data de acesso ao sistema.
- Eventos acessados pelo usuário registrado com qualquer data de acontecimento e data de acesso ao sistema.
- Eventos acessados pelo usuário registrado que ocorrerão posteriormente à data de acesso ao sistema.
- Eventos acessados por qualquer usuário que ocorrerão na data de acesso ao sistema.
- Eventos acessados pelo usuário registrado que ocorrerão na data de acesso ao sistema.
- Avaliação dos locais realizada por qualquer usuário.
- Avaliação dos locais realizada pelo usuário registrado.
- Análise das páginas curtidas pelo usuário registrado no *Facebook*.

Um usuário administrador pode definir pesos, de 1 a 10, para cada um dos critérios de análise utilizados, podendo assim priorizar, ou não, no cálculo aqueles que forem interessante.

O quadro 4.1 mostra os critérios utilizados na recomendação para diferentes tipos de usuário no sistema, onde UNR representa um usuário não-registrado, UR um usuário registrado e UA um usuário administrador.

Quadro 4.1: Critérios levados em consideração para diferentes usuários no sistema.

Critério	UNR	UR	UA
Eventos acessados por qualquer usuário, com qualquer data de acontecimento e data de acesso ao sistema.	Sim	Sim	Sim
Eventos acessados por qualquer usuário que ocorrerão posteriormente à data de acesso ao sistema.	Sim	Sim	Sim
Eventos acessados pelo usuário registrado com qualquer data de acontecimento e data de acesso ao sistema.	Não	Sim	Sim
Eventos acessados pelo usuário registrado que ocorrerão posteriormente à data de acesso ao sistema.	Não	Sim	Sim
Eventos acessados por qualquer usuário que ocorrerão na data de acesso ao sistema.	Sim	Sim	Sim

Eventos acessados pelo usuário registrado que ocorrerão na data de acesso ao sistema.	Não	Sim	Sim
Avaliação dos locais realizada por qualquer usuário.	Sim	Sim	Sim
Avaliação dos locais realizada pelo usuário registrado.	Não	Sim	Sim
Análise das páginas curtidas pelo usuário registrado no <i>Facebook</i> .	Não	Sim	Sim

4.2.4.4 Cálculo de recomendação

A partir da ideia, dos dados obtidos e dos critérios definidos é possível definir um cálculo que transforme essas informações em uma recomendação.

O cálculo de recomendação implementado pode ser dividido em quatro passos: o cálculo dos critérios relacionados a acessos, o cálculo dos critérios relacionados a avaliação de locais, o cálculo dos critérios da análise das páginas curtidas no *Facebook* e a soma dos valores obtidos. A descrição aqui apresentada contempla todos os tipos de usuário no sistema. No caso de um usuário não registrado, é necessário desconsiderar os critérios que não estão definidos para o seu perfil (visto no quadro 4.1).

O cálculo da recomendação, utilizando os critérios de análise de acessos, é descrito no quadro 4.2, onde é informado o critério e o processo utilizado na obtenção do valor de importância de cada local.

Quadro 4.2: Processo de cálculo de recomendação utilizando a análise de acessos.

Critério	Processo
Eventos acessados por qualquer usuário, com qualquer data de acontecimento e data de acesso ao sistema.	<ol style="list-style-type: none"> 1. É obtido o total de acessos separados por local, através dos eventos acessados. 2. É obtido o total de eventos acessados. 3. É feito o cálculo do percentual de acessos que cada local obteve. 4. O percentual é multiplicado pelo peso definido para o critério. 5. O valor resultante é armazenado em um <i>array</i> bidimensional indexado pelo identificador do local e o identificador do critério.
Eventos acessados por qualquer usuário que ocorrerão posteriormente à data de acesso ao sistema.	<ol style="list-style-type: none"> 1. É obtido o total de acessos separados por local, através dos eventos acessados que ocorrerão após a data de acesso ao sistema. 2. É obtido o total de acessos nos eventos que ocorrerão após a data de acesso. 3. É feito o cálculo do percentual de acessos que cada local

sistema.	<p>obteve.</p> <p>4. O percentual é multiplicado pelo peso definido para o critério.</p> <p>5. O valor resultante é armazenado em um <i>array</i> bidimensional indexado pelo identificador do local e o identificador do critério.</p>
Eventos acessados pelo usuário registrado com qualquer data de acontecimento e data de acesso ao sistema.	<p>1. É obtido o total de acessos feitos pelo usuário registrado separados por local, através dos eventos acessados por ele.</p> <p>2. É obtido o total de eventos acessados pelo usuário registrado.</p> <p>3. É feito o cálculo do percentual de acessos que cada local obteve.</p> <p>4. O percentual é multiplicado pelo peso definido para o critério.</p> <p>5. O valor resultante é armazenado em um <i>array</i> bidimensional indexado pelo identificador do local e o identificador do critério.</p>
Eventos acessados pelo usuário registrado que ocorrerão posteriormente à data de acesso ao sistema.	<p>1. É obtido o total de acessos feitos pelo usuário registrado separados por local, através dos eventos acessados por ele que ocorrerão após a data de acesso ao sistema.</p> <p>2. É obtido o total de acessos realizados pelo usuário registrado nos eventos que ocorrerão após a data de acesso ao sistema.</p> <p>3. É feito o cálculo do percentual de acessos que cada local obteve.</p> <p>4. O percentual é multiplicado pelo peso definido para o critério.</p> <p>5. O valor resultante é armazenado em um <i>array</i> bidimensional indexado pelo identificador do local e o identificador do critério.</p>
Eventos acessados por qualquer usuário que ocorrerão na data de acesso ao sistema.	<p>1. É obtido o total de acessos separados por local, através dos eventos acessados que ocorrerão na data de acesso ao sistema.</p> <p>2. É obtido o total de acessos nos eventos que ocorrerão na data de acesso ao sistema.</p> <p>3. É feito o cálculo do percentual de acessos que cada local obteve.</p> <p>4. O percentual é multiplicado pelo peso definido para o critério.</p> <p>5. O valor resultante é armazenado em um <i>array</i> bidimensional indexado pelo identificador do local e o</p>

	identificador do critério.
Eventos acessados pelo usuário registrado que ocorrerão na data de acesso ao sistema.	<ol style="list-style-type: none"> 1. É obtido o total de acessos separados por local, através dos eventos acessados pelo usuário registrado que ocorrerão na data de acesso ao sistema. 2. É obtido o total de eventos que ocorrerão na data de acesso, acessados pelo usuário registrado. 3. É feito o cálculo do percentual de acessos que cada local obteve. 4. O percentual é multiplicado pelo peso definido para o critério. 5. O valor resultante é armazenado em um <i>array</i> bidimensional indexado pelo identificador do local e o identificador do critério.

O quadro 4.3 define o procedimento do cálculo do valor da importância de cada local, em função dos critérios de avaliação de locais. Para cada linha é definido o critério utilizado e o processo realizado no cálculo do valor da importância.

Quadro 4.3: Processo de cálculo de recomendação utilizando a avaliação de locais.

Critério	Processo
Avaliação dos locais realizada por qualquer usuário.	<ol style="list-style-type: none"> 1. É obtida a média das avaliações do local. 2. O valor é multiplicado por 20 e pelo peso do critério. 3. O valor resultante é armazenado em um <i>array</i> bidimensional indexado pelo identificador do local e o identificador do critério.
Avaliação dos locais realizada pelo usuário registrado.	<ol style="list-style-type: none"> 1. É obtida o valor da avaliação realizada pelo usuário registrado para o dado local. 2. O valor é multiplicado por 20 e pelo peso do critério. 3. O valor resultante é armazenado em um <i>array</i> bidimensional indexado pelo identificador do local e o identificador do critério.

Vale ressaltar que na segunda etapa do cálculo no processo de recomendação utilizando a avaliação de locais, o valor da avaliação é multiplicado por 20 para que haja equiparação com o valor máximo que é possível obter no processo de recomendação utilizando a análise de acessos. Com isso, nos dois casos, o valor máximo que pode ser atingido é de 100 pontos (desconsiderando-se os pesos).

O quadro 4.4 define o processo utilizado no cálculo da recomendação através da análise das páginas curtidas no *Facebook*.

Quadro 4.4: Processo do cálculo de recomendação utilizando a análise das páginas curtidas no *Facebook*.

Critério	Processo
Análise das páginas curtidas pelo usuário registrado no <i>Facebook</i> .	<ol style="list-style-type: none"> 1. É obtida da base de dados os registros das páginas que o usuário curtiu no <i>Facebook</i>. 2. Para cada local cadastrado no sistema é verificado se o usuário curtiu a sua página no <i>Facebook</i>. 3. Caso verdadeiro, retorna o valor 100. Caso contrário, o valor 0. 4. O valor é multiplicado pelo peso do critério. 5. O valor é armazenado em um <i>array</i> bidimensional indexado pelo identificador do local e o identificador do critério.

Assim como no processo anterior, o valor de 100, busca trazer igualdade entre o valor máximo que pode ser obtido entre os diferentes tipos de critérios.

Ao fim desses três passos, teremos um *array* indexado pelo identificador do local e os nove valores resultantes do cálculo de cada critério, como pode ser visto na figura 4.10.

```

array
(
  1 => array
    (
      'name' => 'Beco 203'
      'value1' => 7.6923076923077
      'value2' => 8.2644628099174
      'value3' => 7.7519379844961
      'value4' => 8.3333333333333
      'value8' => 1.2345679012346
      'value5' => 0
      'value6' => 0
      'value7' => 100
      'value9' => 0
    )
  2 => array
    (
      'name' => 'Clube Silêncio'
      'value1' => 3.0769230769231
      'value2' => 0
      'value3' => 3.1007751937984
      'value4' => 0
      'value8' => 0
      'value5' => 0
      'value6' => 0
      'value7' => 0
      'value9' => 0
    )
  3 => array
    (
      'name' => 'Dhomba'
      'value1' => 3.8461538461538
      'value2' => 0
      'value3' => 3.8759689922481
      'value4' => 0
      'value8' => 0
      'value5' => 0
      'value6' => 0
      'value7' => 0
      'value9' => 0
    )
  4 => array
    (
      'name' => 'Opinião'
      'value1' => 73.846153846154
      'value2' => 79.338842975207
      'value3' => 73.643410852713
      'value4' => 79.166666666667
      'value8' => 98.765432098765
      'value5' => 0
      'value6' => 0
      'value7' => 0
      'value9' => 0
    )
)

```

Figura 4.10: Exemplo de saída dos cálculos de recomendação dos critérios definidos.

O próximo passo é a soma dos valores de cada local e o reordenamento do *array* de forma decrescente, como pode ser visto na figura 4.11. O índice das primeiras posições do *array* representam os identificadores dos locais onde provavelmente haverá mais interesse do usuário sobre os eventos que lá ocorrerão.

```

array
(
  4 => array
    (
      'value' => 227.48956469887
    )
  1 => array
    (
      'value' => 160.4442456768
    )
  23 => array
    (
      'value' => 78.533094812165
    )
  12 => array
    (
      'value' => 13.088849135361
    )
  3 => array
    (
      'value' => 7.7221228384019
    )
)

```

Figura 4.11: Saída com a soma dos valores de todos os critérios, ordenada de forma decrescente.

O último passo no cálculo de recomendação implementado no sistema é a passagem dos identificadores de locais de maior pontuação ao método de ordenamento da consulta realizada à base de dados, que traz os eventos que ocorrerão na data selecionada pelo usuário.

No trecho de código exibido na figura 4.12, pode ser visto como é realizado este último passo, internamente: é feita uma busca para a obtenção de todos os registros do modelo *Event*, que ocorram na data selecionada pelo usuário, ordenando-os através dos seus locais de preferência. Por fim, os dados são enviados para a visão, que é responsável pela última etapa do algoritmo de recomendação no sistema.

```

$events = Event::model()->findAllByAttributes(
    array("date_time" => $date),
    array("order" => $recommended_places)
);

$this->render('_recommended_events', array('events' => $events, 'date' => $date));
}

```

Figura 4.12: Trecho de código responsável por buscar os eventos recomendados ao usuário.

4.2.4.5 Visualização Personalizada

A última etapa do processo de recomendação no sistema proposto é feita através de uma visão que implementa uma variação de um *treemap*, com apenas um nível. Conforme descrito no capítulo 3, a ideia básica da técnica de *treemapping* é conceder maior espaço na tela para itens que possuam maior importância no contexto.

O sistema proposto implementa 4 retângulos com diferentes tamanhos de base e altura que são utilizados para dar destaque para os eventos recomendados. A figura 4.13, apresenta as figuras geométricas utilizadas e é possível visualizar suas relações de tamanho e altura.

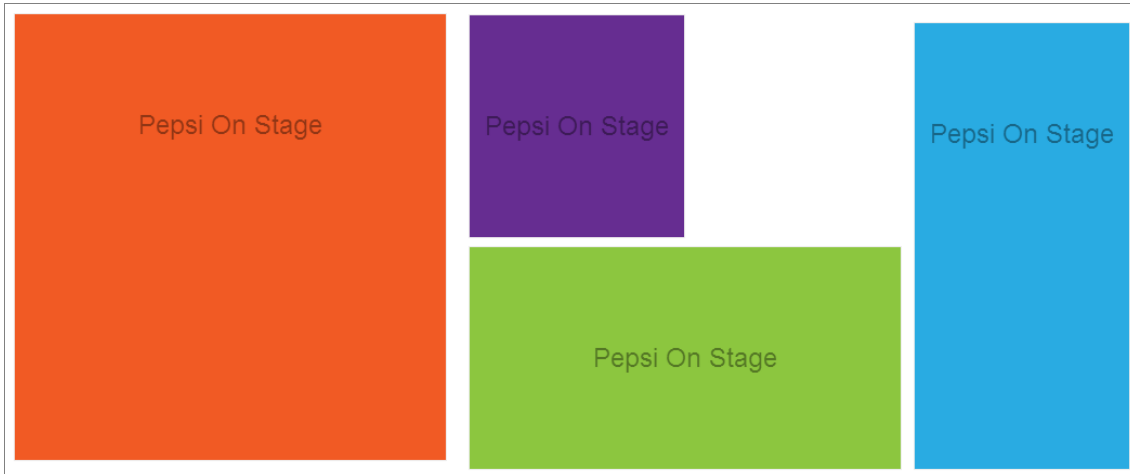


Figura 4.13: Retângulos utilizados para destacar os eventos recomendados.

Os três locais que possuem maior valor de importância para o usuário são anexados ao retângulo laranja, azul e verde, respectivamente. Os demais locais, são anexados ao retângulo roxo e, randomicamente, há o anexo de locais com um valor de importância não tão elevado aos retângulos azul e verde, na tentativa de atrair visualmente o usuário para uma opção que, provavelmente ele dispensaria, mas que pode ser uma oportunidade de conhecer locais novos. Esse comportamento pode ser visto na figura 4.14, onde podemos notar que o evento que ocorrerá no Opinião possui o maior valor de importância, seguido pelo Beco 203 e pelo Clube Silêncio.

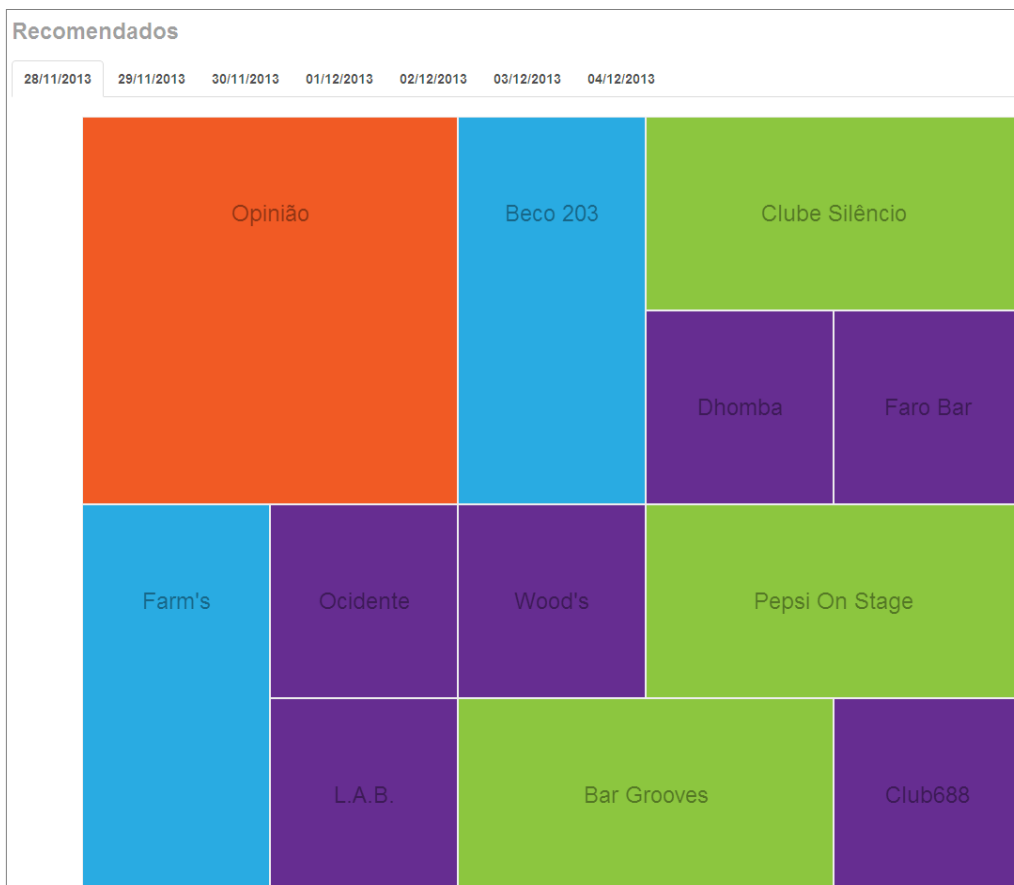


Figura 4.14: Etapa final da recomendação, a exibição ao usuário.

4.2.5 Framework Yii

O sistema proposto foi desenvolvido seguindo o padrão MVC, com o auxílio do *framework* Yii, ambos definidos no capítulo 3. As seções abaixo apresentam as classes e páginas criadas, sendo descrito o seu comportamento no desenvolvimento do sistema.

4.2.5.1 Modelos

A figura 4.15 exibe a estrutura de classes que representam os Modelos, no padrão MVC.

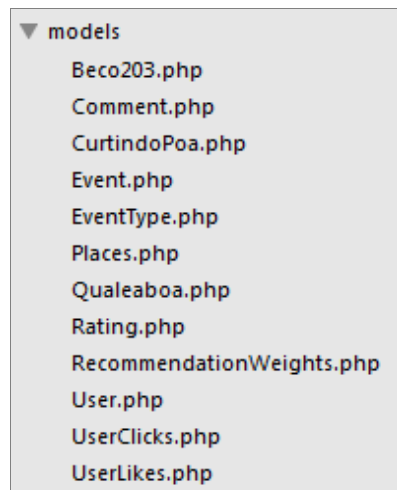


Figura 4.15: Estrutura dos Modelos do sistema.

Com exceção dos modelos que adquirem informação através do módulo de *Web crawling*, cada uma das classes de Modelo é a abstração de uma tabela na base de dados e realiza as operações através de métodos definidos. A classe *Event.php*, por exemplo, possui métodos responsáveis por consultar, editar, inserir e remover um evento na base de dados.

4.2.5.2 Visões

A figura 4.16 apresenta a estrutura de pastas com arquivos PHP que representam a Visão, no padrão MVC.

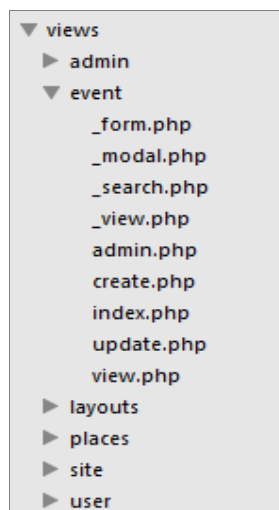


Figura 4.16: Estrutura de arquivos do componente Visão.

O componente Visão é representado por diversos arquivos PHP, cada arquivo representa uma tela diferente no sistema e está relacionado a um Controlador. Um exemplo pode ser descrito utilizando a visão *create.php*, na pasta *event*, que é responsável pela criação do formulário de cadastro de um evento no sistema, contendo os campos necessários para isto. Ao clicar no botão para salvar os dados é disparada uma ação no Controlador de eventos, *EventController.php*, que por sua vez deve utilizar os métodos de inserção da classe responsável pelo Modelo de eventos, *Event.php*.

Conforme dito no capítulo 3, toda vez que é renderizada uma visão, o *framework* Yii embute uma estrutura HTML criada e definida como *layout*. Normalmente, um *layout* é responsável pela estrutura HTML que deve ser utilizada em diversas visões, como por exemplo, o cabeçalho do site, o rodapé e o estilo onde é exibido o conteúdo. Entretanto, o *framework* possibilita que essa etapa seja ignorada em seu *workflow*, renderizando somente a estrutura HTML presente na visão selecionada. Visões que são renderizadas sem o *layout*, usualmente, possuem seus nomes precedido de *underline*, como, por exemplo, o *_form.php*, visto na figura 4.16.

4.2.5.3 Controladores

A figura 4.17 apresenta a estrutura de classes dos Controladores, um dos componentes do padrão MVC.

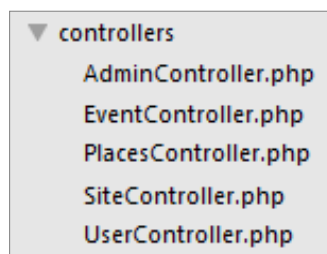


Figura 4.17: Estrutura das classes Controladores.

Cada um dos arquivos de classes PHP vistos na figura 4.17, representa um controlador no padrão MVC. Eles estão associados a visões exibidas ao usuário e são acionados a partir de ações executadas pelo mesmo, como o clique em um *link*, por exemplo.

Uma ação do usuário é relacionada a uma ação de um controlador, representada por um método na classe, que tem como objetivo identificar os dados necessários do Modelo e, após o processamento, encaminhar para o arquivo de visualização (a Visão) os dados gerados, para que sejam exibidos ao usuário.

5 USO E NAVEGAÇÃO NO SISTEMA

Este capítulo apresentará as telas do sistema desenvolvido, explicando o seu uso com o objetivo de demonstrar o funcionamento do sistema. O capítulo é dividido em duas grandes seções: tela para usuários e as telas de gerência do sistema.

As telas para usuários englobam as visões que os usuários têm no sistema.

As telas de gerência do sistema apresentam as visões que são exclusivas a um usuário administrador.

5.1 Tela para usuários

Ao entrar na *URL* onde está hospedado o sistema, a aba de eventos recomendados, exibida na figura 5.1, é a primeira tela vista pelo usuário.

É possível dividir a tela em duas partes: o cabeçalho e a divisória de conteúdo. No cabeçalho estão disponibilizadas as abas com opções de uso do sistema. Cada aba será descrita em uma seção, posteriormente. A divisória de conteúdo é o local onde são apresentados os dados referentes à aba escolhida pelo usuário. No primeiro acesso, a aba padrão selecionada é a de visualização dos eventos recomendados, logo, a divisória de conteúdo, nesse caso, apresenta a visão de eventos recomendados.

5.1.1 Aba “Eventos recomendados”

A visão de eventos recomendados possui sete abas que representam os dias da semana, onde a data em que o sistema está sendo acessado é a opção padrão. Cada vez que uma dessas abas é selecionada, é executado o processo de recomendação definido no capítulo 4.

Ao clicar em um dos locais recomendados, o fundo do sistema é escurecido e é aberta uma modal com a imagem de divulgação do evento selecionado, juntamente com as informações de data e local.

Conforme pode ser visto na figura 5.2, ao clicar sobre o cartaz de divulgação, o usuário é redirecionado à página original do evento, em uma nova janela no seu navegador, e assim ele pode obter mais dados, sem prejudicar a navegação no sistema proposto.

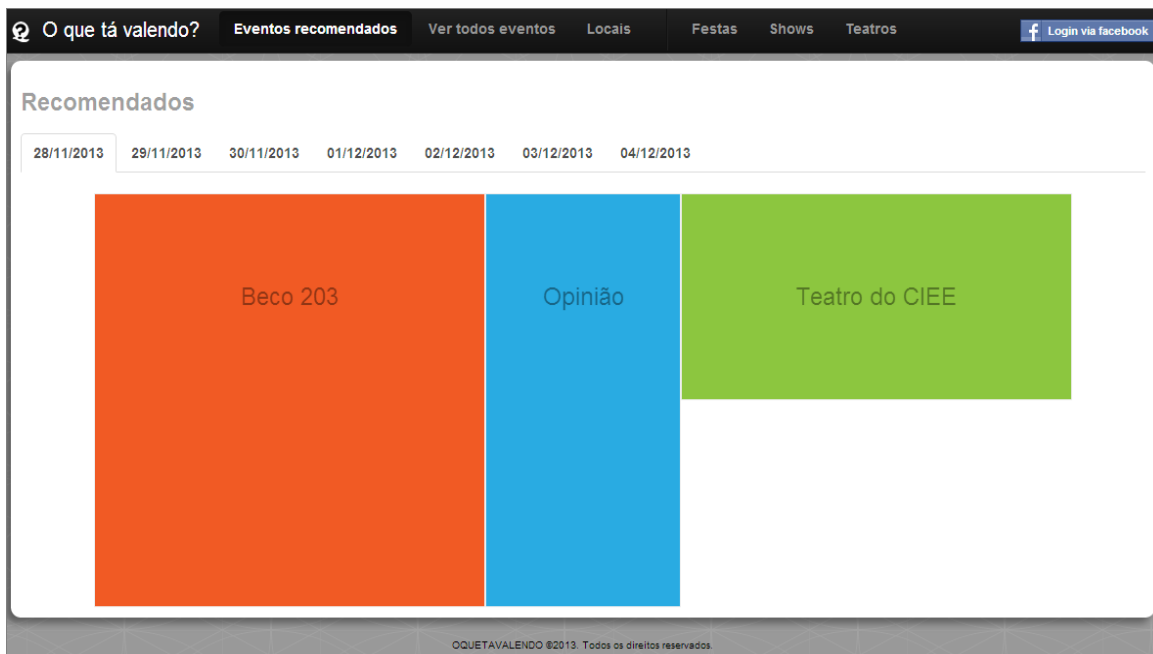


Figura 5.1: Tela inicial, com a aba “Eventos recomendados” selecionada.

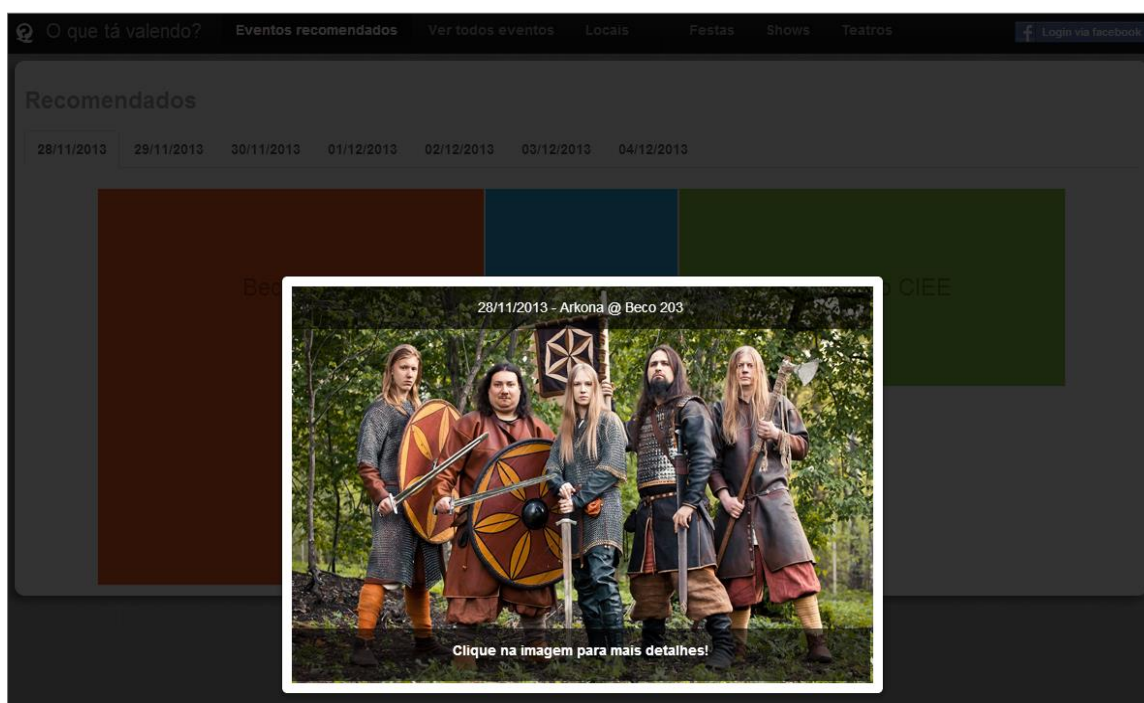


Figura 5.2: Exibição das informações do evento em uma modal.

5.1.2 Aba “Ver todos os eventos”

Ao selecionar a aba “Ver todos os eventos” a divisória de conteúdo é preenchida por todos os eventos cadastrados no sistema que estão para acontecer, em forma de um mosaico, onde cada evento exibe uma data, um local, uma miniatura de sua imagem de divulgação e o nome do evento, como visto na figura 5.3.

Como ocorre na aba de “Eventos recomendados”, ao clicar em algum dos eventos, é aberta a modal, vista na figura 5.2, para a exibição da imagem de divulgação no tamanho original e a possibilidade de obter mais detalhes.

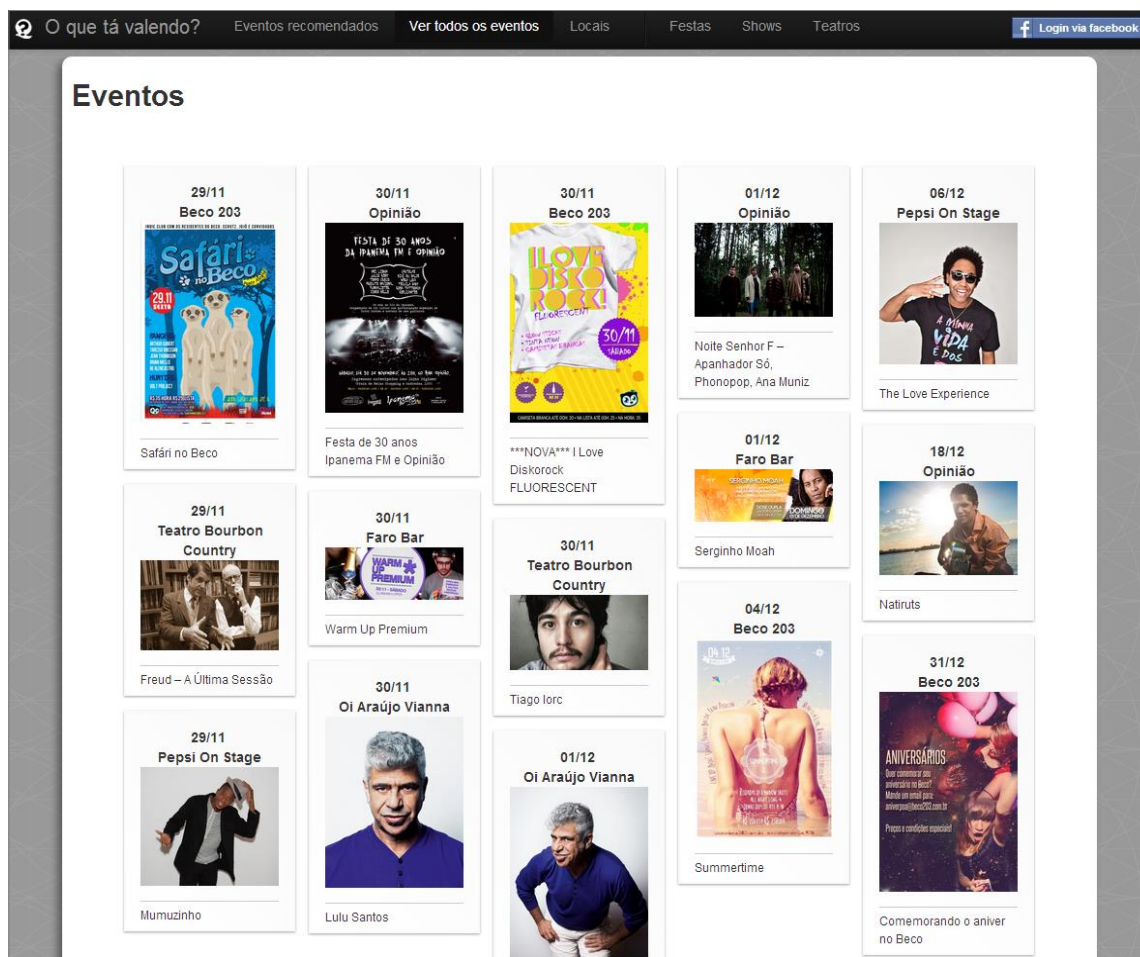


Figura 5.3: Aba “Ver todos os eventos”.

5.1.3 Aba “Locais”

Ao ser selecionada a aba “Locais”, a divisória de conteúdo é preenchida por todos os locais cadastrados no sistema, em forma de um mosaico, onde cada local contém um nome, uma imagem de exibição, uma média, e um botão chamado “Saiba mais”, que redireciona o usuário para a página oficial do local, em uma nova aba do navegador.

A figura 5.4, apresenta a visão de um usuário não-registrado, ou seja, que não realizou o *login* no sistema. Como podemos ver, ele tem acesso à visualização da média das avaliações dos locais realizadas por usuários registrados, entretanto, não pode realizar avaliações. Na figura 5.5, temos a visão de um usuário registrado onde é possível notar o novo campo “Sua avaliação”, que possibilita ao usuário efetuar, ou editar, uma avaliação para os locais cadastrados.

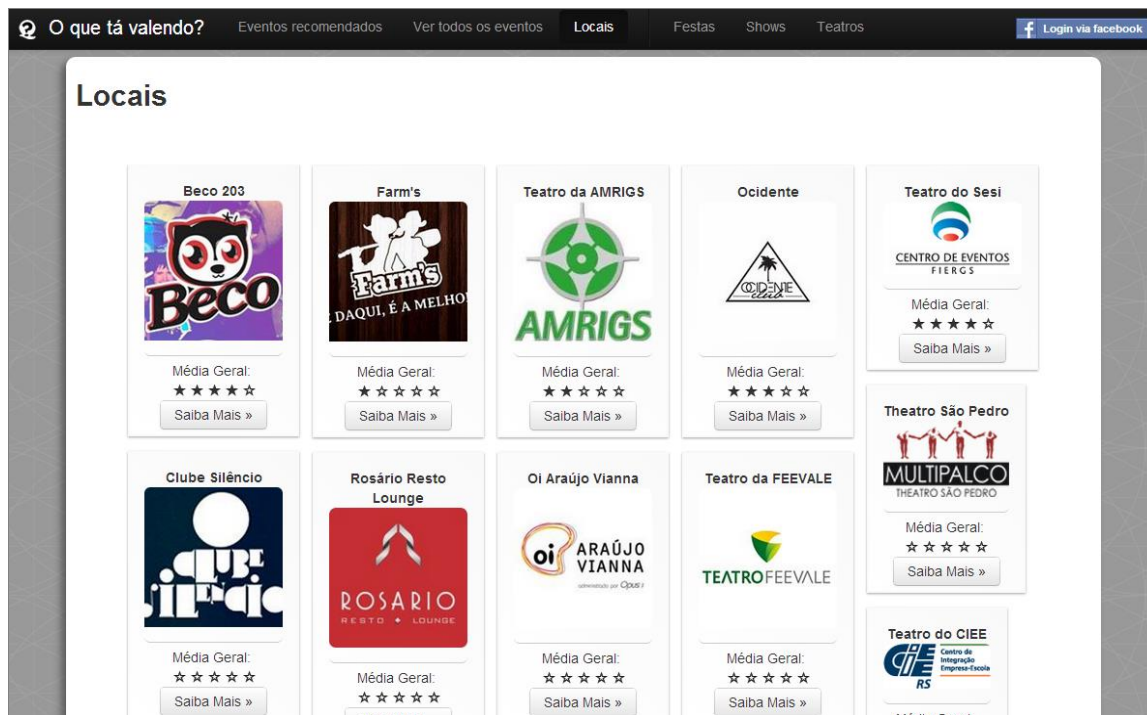


Figura 5.4: Visualização da aba locais por um usuário não-registrado.

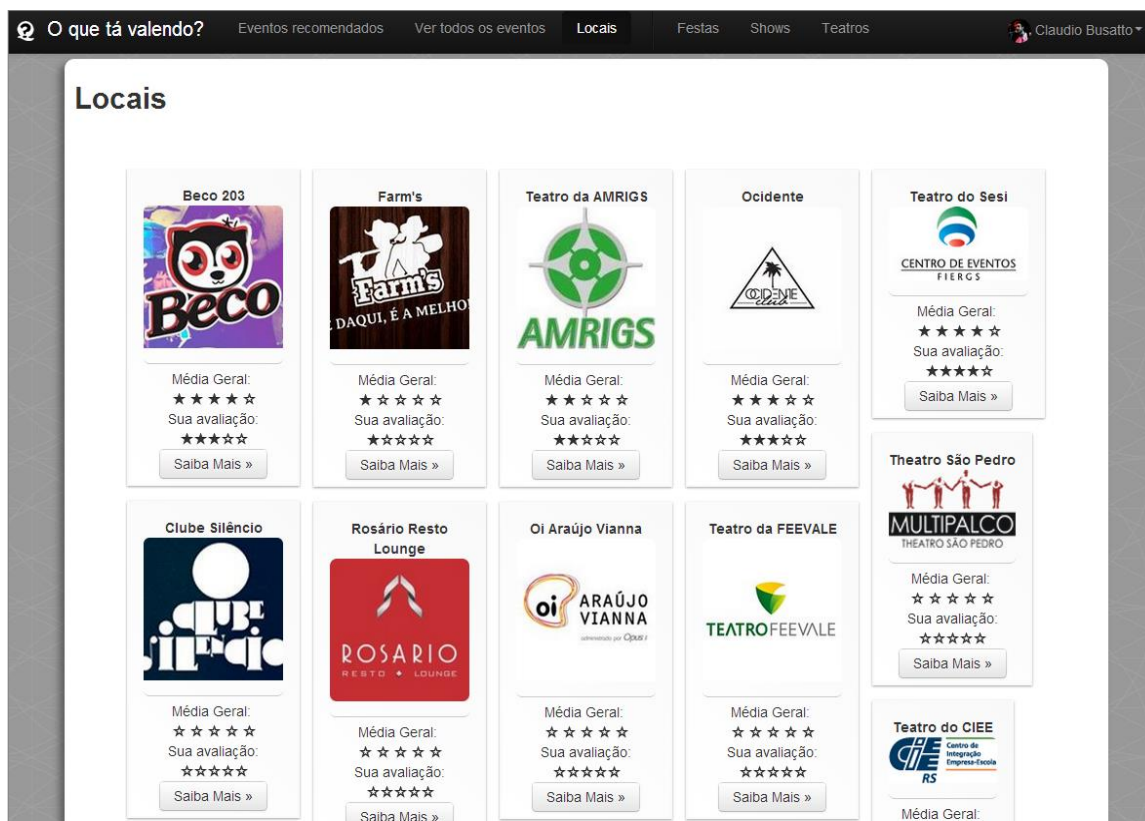


Figura 5.5: Visualização da aba locais por um usuário registrado.

Ao realizar uma avaliação, ou editar uma avaliação anterior, clicando em uma quantidade de estrelas, é disparada a ação no sistema. Caso tudo ocorra corretamente é aberto um *pop-up*, visto na figura 5.6, informando ao usuário sobre o sucesso da avaliação.

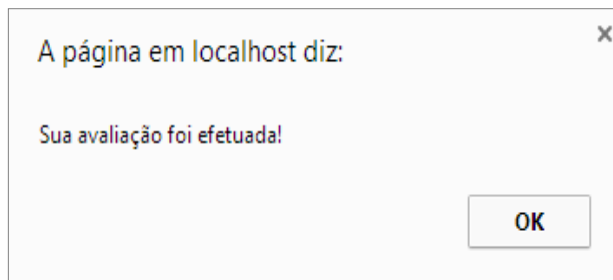


Figura 5.6: *Pop-up* de confirmação da avaliação.

5.1.4 Abas “Festas”, “Shows” e “Teatros”

As três abas (“Festas”, “Shows” e “Teatros”) possuem um comportamento similar à aba “Ver todos os eventos” com a única diferença de que os eventos exibidos na divisória de conteúdo são filtrados pelo tipo do evento. Em outras palavras, enquanto a aba “Ver todos os eventos” exhibe os eventos relacionados a festas, shows e teatros, a aba “Teatros”, exhibirá apenas os eventos com o tipo de evento definido como Teatro, como visto na figura 5.7. O mesmo se aplica às abas “Festas” e “Shows” que exibem apenas os eventos do tipo Festa e Show, respectivamente.

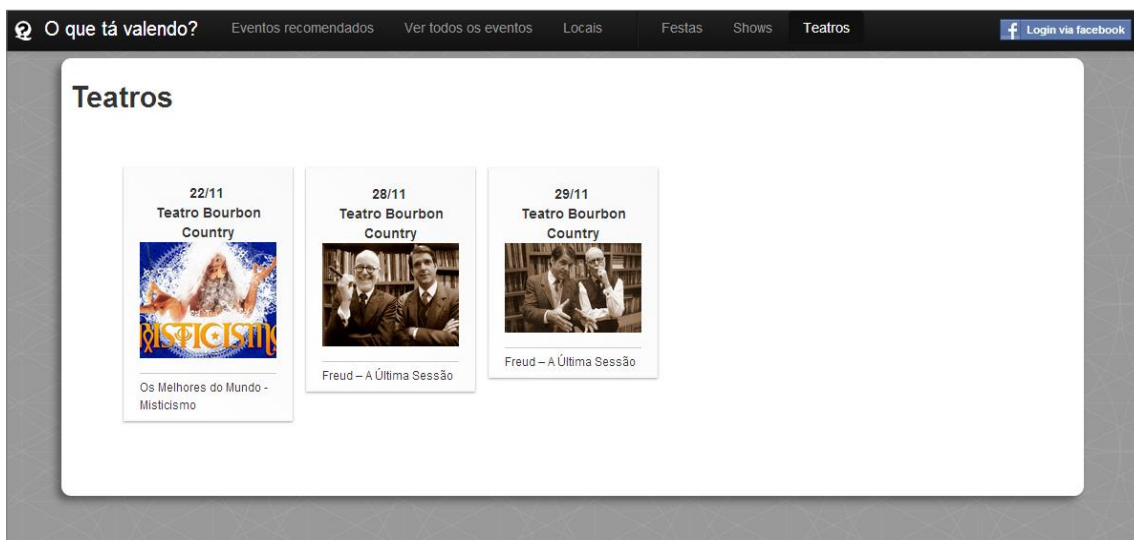


Figura 5.7: Aba “Teatros”.

5.1.5 Botão de Login via Facebook

Ao clicar pela primeira vez no botão “Login via Facebook” é aberto um *pop-up* perguntando se o usuário concede acesso ao aplicativo do sistema proposto a alguns dados privados do perfil na rede social, como visto na figura 5.8. O usuário só consegue se registrar no sistema caso conceda acesso a esses dados.

Ao confirmar, o local onde antes era exibido o botão de *login*, agora é substituído pelo nome do usuário e a sua imagem de exibição, ambos extraídos da rede social. Ao clicar sobre o nome, é aberto um menu de opções, como visto na figura 5.9.

É importante salientar que o protocolo de concessão de acesso aos dados só ocorre na primeira vez que é realizado o *login*, nas outras oportunidades o usuário já possui, em sua rede social, o aplicativo que realiza a autenticação instalado e o *Facebook* interpreta isso como uma manutenção da confiança no sistema.



Figura 5.8: *Pop-up* de confirmação da concessão dos dados.

5.2 Telas de gerência do sistema

Ao executar o *login*, e tiver o perfil de administrador, é oferecida a opção de “Gerenciar o sistema” ao usuário, como visto na figura 5.9. Ao selecioná-la o administrador é redirecionado para as telas de gerência do sistema, vistas na figura 5.10.

A interface de gerência do sistema possui um cabeçalho com seis abas: “Minerar Novos Eventos”, “Alterar Pesos das Recomendações”, “Cadastrar Eventos”, “Editar Eventos”, “Cadastrar Locais”, “Editar Locais” e “Editar Usuários”. Essas abas e suas funcionalidades serão descritas nas seções abaixo.

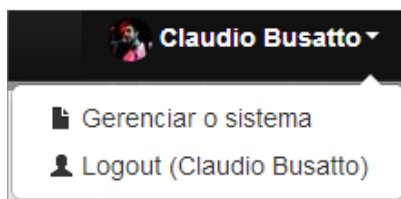


Figura 5.9: Opção, disponível a administradores, de gerência do sistema.

5.2.1 Aba “Minerar Novos Eventos”

Ao clicar na aba “Minerar Novos Eventos” são exibidos os sites que estão disponíveis para a realização do processo de *Web crawling* e, conseqüentemente, da obtenção dos seus eventos cadastrados. Atualmente, três sites estão definidos, como pode ser visto na figura 5.10.



Figura 5.10: Aba “Minerar Novos Eventos”.


Ao selecionar um dos sites disponibilizados, o módulo de *Web crawling* inicia a sua execução e, após o processamento, é exibida a lista de eventos obtidos, como pode ser visto na figura 5.11. Cada evento exibe uma data de realização, um título, uma imagem de divulgação e o local onde ele irá ocorrer. O administrador tem a opção de definir o tipo do evento antes de adicioná-lo à base de dados, clicando no botão “Adicionar Evento”. Caso o processo de adicionar o evento tenha sucesso, o botão clicado troca de cor para o verde e é exibida uma mensagem informando ao administrador que o evento foi adicionado com sucesso, estando a partir daquele momento disponível para a visualização dos outros usuários, através das Telas para usuários.

Admin @ O que tá valendo?

Minerar Novos Eventos Alterar Pesos das Recomendações Cadastrar Eventos Editar Eventos Cadastrar Locais Editar Locais Editar Usuários

Selecione de quais sites você deseja buscar eventos: Qual é a boa? Curtindo Poa Beco 203

28/11
Freud – A Última Sessão




Local: Teatro Bourbon Country

Festa

Adicionar Evento


28/11
Rocket Man – Elton John Tribute



Local: Teatro do CIEE

Evento adicionado!


29/11
Mumuzinho



Local: Pepsi On Stage

Evento adicionado!


30/11
Tiago Iorc



Local: Teatro Bourbon Country

Evento adicionado!


28/11
Open Bar da Patricia



Local: Faro Bar

Evento adicionado!

28/11
Jazz Trio




Local: Cult Pub

Festa

Adicionar Evento

29/11
Workstation




Local: Cult Pub

Festa

Adicionar Evento


30/11
Lulu Santos



Local: Oi Araújo Vianna

Evento adicionado!

30/11
Festa Khalifa (Skol Beach Fest 2013)



Villa Assumpção

Adicionar local

Figura 5.11: Exemplo dos dados obtidos do site “Qual é a boa?”

Caso um dos eventos exibidos esteja programado para ocorrer em um local que ainda não está cadastrado no sistema, é exibido um botão para “Adicionar Local”, como pode ser visto na figura 5.12. Ao clicar nesse botão, é aberta uma modal, que pode ser vista na figura 5.13, para o cadastro do novo local sem precisar sair da aba atual, melhorando assim a usabilidade do sistema.



Figura 5.12: Exemplo de evento que ocorrerá em um local ainda não cadastrado no sistema.

Admin @ O que tá valendo?

Minerar Novos Eventos | Alterar Pesos das Recomendações | Cadastrar Eventos | Editar Eventos | Cadastrar Locais | Editar Locais | Editar Usuários

Cadastrar Local ✕

Campos com * são obrigatórios.

Nome *

Variações do nome (separados por vírgula) *

Link para o site

Link para imagem de exibição

E-Mail

Endereço

Número

Complemento

Telefone

Adicionar Local

Seleção de quais sites v

28/11
Freud – A Última Sessão
Local: Teatro Bourbon Country
Evento adicionado!

28/11
Open Bar da Patricia
Local: Faro Bar
Evento adicionado!

28/11
Arkona

29/11
Mumuzinho
Beira do Guaíba
Adicionar local

Beco 203

30/11
Lulu Santos
Local: Oi Araújo Vianna
Evento adicionado!

30/11
Festa de 30 anos Ipanema FM e Opinião
FESTA DE 30 ANOS DA IPANEMA FM E OPINIÃO

Figura 5.13: Modal para cadastro de novo local dentro da interface “Minerar Novos Eventos”.

5.2.2 Aba “Alterar Peso das Recomendações”

Ao acessar a aba “Alterar Peso das Recomendações” é exibida a lista de critérios utilizados no cálculo de recomendação, uma barra de rolagem horizontal e o valor numérico do peso atual relacionado ao critério, conforme pode ser visto na figura 5.14.

Pode-se aumentar ou diminuir o peso de um critério no cálculo de recomendação arrastando a barra de rolagem para a esquerda (diminuir) ou para a direita (aumentar).

Ao final da alteração dos pesos, ao clicar no botão “Salvar novos pesos”, persiste-se esses novos valores na base de dados e o cálculo de recomendação passa a ser feito utilizando os valores, agora, redefinidos.

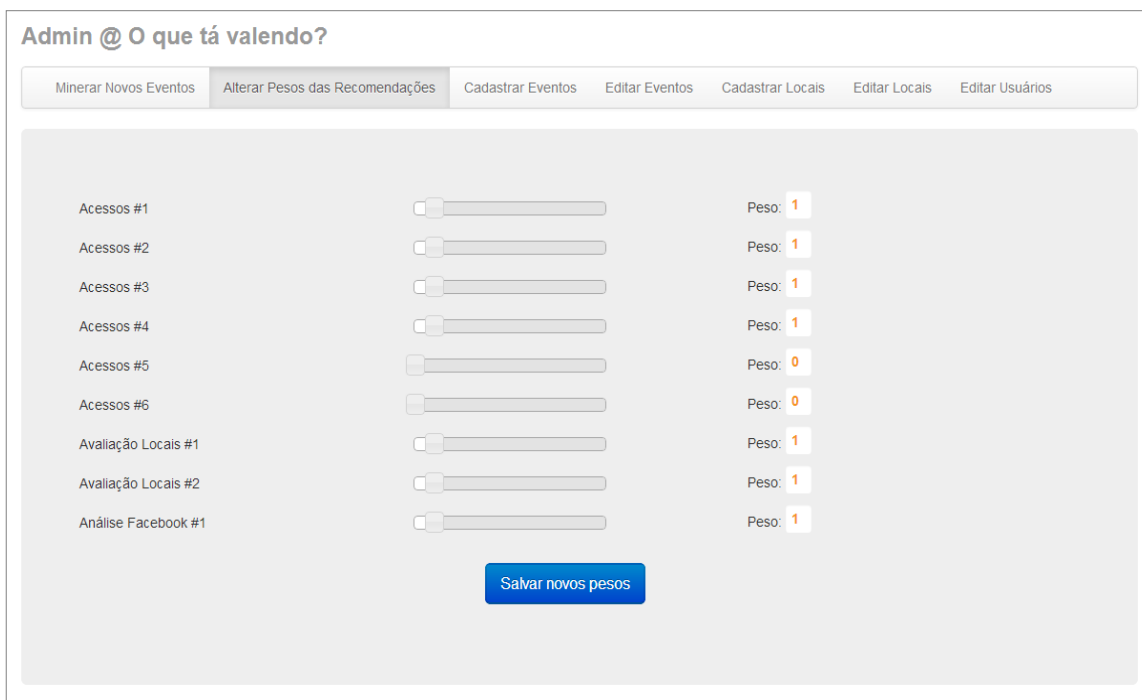


Figura 5.14: Aba “Alterar Pesos das Recomendações”.

5.2.3 Abas “Cadastrar Eventos” e “Cadastrar Locais”

As abas “Cadastrar Eventos” e “Cadastrar Locais” possuem um comportamento muito semelhante, sendo as duas, interfaces de cadastro de eventos e locais, respectivamente.

Ao clicar nelas, um formulário, similar ao exibido na figura 5.15, é exibido ao usuário, para o cadastro de novas informações. Ao preencher os campos e clicar no botão “salvar”, é feita a validação dos dados. Caso haja sucesso, um novo evento (ou local) é cadastrado no sistema, caso existam erros é informado ao administrador quais campos necessitam ser corrigidos.

The screenshot shows a web interface for an administrator. At the top, there is a navigation bar with the text 'Admin @ O que tá valendo?' and several menu items: 'Minerar Novos Eventos', 'Alterar Pesos das Recomendações', 'Cadastrar Eventos' (which is highlighted), 'Editar Eventos', 'Cadastrar Locais', 'Editar Locais', and 'Editar Usuários'. Below the navigation bar is a large form titled 'Cadastrar Evento'. Under the title, it says 'Campos com * são obrigatórios.' The form contains several input fields: 'Titulo *' (a text box), 'Tipo *' (a dropdown menu with 'Selecione um tipo'), 'Local *' (a dropdown menu with 'Selecione o Local'), 'Imagem (link)' (a text box), and 'Data e hora' (a text box). A green button labeled 'Salvar evento' is positioned below the form fields.

Figura 5.15: Aba de cadastro.

5.2.4 Abas “Editar Eventos”, “Editar Locais” e “Editar Usuários”

As abas “Editar Eventos”, “Editar Locais” e “Editar Usuários” possuem um comportamento similar, sendo todas, interfaces para visualização, edição e remoção de eventos, locais e usuários, respectivamente.

A interface, vista na figura 5.16, exibe a tabela de eventos cadastrados no sistema possibilitando a visualização completa dos dados (através do ícone com a lupa), a edição dos dados (através do ícone com o lápis) e a remoção do registro (através do ícone com a letra “x”).

A primeira linha da tabela possui campos de entrada de texto que permitem ao usuário filtrar os dados exibidos, utilizando operadores lógicos, a fim de facilitar a busca do registro desejado.

A visualização dos dados, como visto na figura 5.17, mostra o registro completo do evento selecionado.

A edição dos dados, visto na figura 5.18, possui o mesmo comportamento da aba de cadastro, se diferenciando apenas pelo fato de já possuir os campos devidamente preenchidos.

A remoção do registro apresenta um *pop-up* de confirmação, como visto na figura 5.19 que, caso seja confirmado, exclui o registro da base de dados.

Admin @ O que tá valendo?




Minerar Novos Eventos Alterar Pesos das Recomendações Cadastrar Eventos **Editar Eventos** Cadastrar Locais Editar Locais Editar Usuários

Editar Eventos

Você pode utilizar os operadores lógicos (<, <=, >, <> ou =) no começo de cada um dos seus valores de busca para especificar como a comparação deve ser realizada.

Pressione ENTER para efetuar a busca.

Exibindo 1-10 de 134 resultados.

Id Event	Título	Tipo	Local	Data e hora	
127	Rock Clube	1	2	2013-10-17 23:55:00	 
128	Quinta-Feira	1	6	2013-10-17 23:55:00	 
129	Open Bar da Patricia	1	8	2013-10-17 23:55:00	 
130	Quinta-Feira	1	13	2013-10-17 23:55:00	 
131	Zeca Pagodinho	1	12	2013-10-17 23:55:00	 
132	Groundation + Reemah	1	4	2013-10-17 23:55:00	 
133	Gang Bang	1	1	2013-10-17 23:55:00	 
134	Oktoberfest	1	10	2013-10-18 23:55:00	 
135	Vegas	1	2	2013-10-18 23:55:00	 
136	Rindo Afú	1	11	2013-10-18 23:55:00	 

Ir à página: [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [Próximo >](#)

Figura 5.16: Aba “Editar Eventos”.

Admin @ O que tá valendo?

Minerar Novos Eventos Alterar Pesos das Recomendações Cadastrar Eventos **Editar Eventos** Cadastrar Locais Editar Locais Editar Usuários

Safári no Beco

Id Event	260
Título	Safári no Beco
Tipo	1
Local	1
Imagem (link)	http://www.beco203.com.br/resources/conteudos/imagens/agenda/big/1385149241.jpg
Link	http://www.beco203.com.br/agenda/safri-no-beco-17-40-42
Data e hora	2013-11-29 23:55:00

Figura 5.17: Opção “Visualização”.

Admin @ O que tá valendo?

Minerar Novos Eventos Alterar Pesos das Recomendações Cadastrar Eventos Editar Eventos Cadastrar Locais Editar Locais Editar Usuários

Editar Evento

Campos com * são obrigatórios.

Titulo *	Tipo *	Local *
<input type="text" value="Safári no Beco"/>	<input type="text" value="Festa"/>	<input type="text" value="Beco 203"/>
Imagem (link)	Data e hora	
<input type="text" value="http://www.beco203.com.br/resou"/>	<input type="text" value="2013-11-29 23:55:00"/>	

Figura 5.18: Opção de “Edição”.

A página em localhost diz:

Deseja realmente excluir este item?

Figura 5.19: *Pop-up* de confirmação na opção “Exclusão”.

6 CONCLUSÕES

A tentativa de criar um sistema que auxilie a escolha de eventos e atividades de lazer motivou a criação deste trabalho, que teve como objetivo desenvolver um sistema *Web* que possibilitasse a divulgação e recomendação de eventos, bem como a criação de um ambiente de fácil gerência, onde os eventos pudessem ser cadastrados de forma dinâmica, sem a necessidade de preenchimento de formulários, por parte do administrador.

Após a análise das necessidades, transformadas em histórias de usuário, foi iniciada a modelagem do sistema, bem como a escolha das tecnologias e metodologias que seriam utilizadas. A escolha da utilização de um *framework* de desenvolvimento, na tentativa de se usar o que há de mais recente no desenvolvimento *Web*, acabou dificultando o desenvolvimento no início, devido à curva de aprendizagem na utilização do padrão MVC e do funcionamento, em si, do *framework*. Porém, com a experiência adquirida pode-se experienciar, de fato, as vantagens de sua utilização e a agilidade que ele agrega ao desenvolvedor.

Ao longo do desenvolvimento diversas ideias surgiram e foram adicionadas ao projeto inicial, a fim de tornar o sistema mais completo, com melhor usabilidade e menor tempo de resposta ao usuário.

6.1 Resultados

Através da definição das histórias, do estudo de sistemas similares, das tecnologias e conceitos utilizados foi possível atingir o objetivo proposto, sendo um dos principais a criação de uma interface de gerência que obtém as informações sobre eventos através de sites externos, possibilitando a alimentação da base de dados de maneira dinâmica.

O sistema proposto implementa um método de recomendação simples, porém que realiza aquilo ao que se propõe. A utilização de diferentes formas visuais e de tamanhos proporcionais à importância do item para o usuário, através do conceito de *treemapping*, introduz uma nova ideia a ser explorada nos estudos de sistemas de recomendação.

6.2 Limitações

Durante os testes e uso do sistema foi constatado que a versão atual possui uma limitação no módulo de *Web crawling* que, ao ser executado, consome todos os recursos do servidor no processamento, o que acaba impedindo o uso do sistema por outros usuários até o término do processamento. A execução do módulo de *Web crawling*, para cada um dos sites definidos, demora em média vinte segundos, tempo esse em que o servidor e o sistema, conseqüentemente, ficam indisponíveis.

Por não estar, ainda, disponível na *Web* para o público em geral, não foi possível realizarmos a avaliação de uso por usuários reais.

6.3 Trabalhos futuros

Com o objetivo de corrigir a limitação hoje apresentada no módulo de *Web crawling*, seria interessante a trabalhos futuros a implementação do uso de *threads* na execução do processamento. A paralelização das requisições HTTP a sites externos, feitas para a obtenção do código HTML, deverá diminuir consideravelmente o tempo de inacessibilidade do sistema.

Normalmente, os sites de eventos realizam o cadastro de novos eventos em diferentes dias da semana e em diferentes horários. Uma melhoria para o futuro seria a criação de um *script* que ficasse monitorando a adição de novos eventos nos sites definidos, informando ao administrador quando um novo evento for adicionado e possibilitando a sua inserção no “O que tá valendo?”.

Durante o desenvolvimento surgiu a ideia da criação de um ambiente de rede social entre os usuários do sistema, possibilitando assim uma interação entre eles, a troca de experiências e de sugestões de locais. Além disso, as novas informações que seriam criadas também poderiam ser adicionadas no processo de recomendação.

Faz-se necessário, também, a realização da avaliação de uso por usuários reais para que seja medida a satisfação quanto à facilidade de uso do sistema, à taxa de acerto nas recomendações efetuadas e à praticidade introduzida pela solução, em comparação aos sistemas existentes.

REFERÊNCIAS

ADOMAVICIUS, Gediminas; TUZHILIN, Alexander. **Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions.** IEE Transactions on Knowledge and Data Engineering, v. 17, n. 6, p. 734-749, 2005.

ALLEN, Rob; LO, Nick; BROWN, Steven. **ZEND framework em ação.** Rio de Janeiro: Alta Books, 2009.

BEDERSON, B.B.; SHNEIDERMAN, B.; WATTENBERG, M. **Ordered and Quantum Treemaps: Making Effective Use of 2D Space to Display Hierarchies.** ACM Transactions on Graphics (TOG), v. 21, n. 4, p. 833-854, 2002.

BURKE, Robin. **Hybrid recommender systems: Survey and experiments.** User Modeling and User Adapted Interaction, v.12, n. 6, p. 331–370, nov. 2002. Disponível em: < <http://josquin.cti.depaul.edu/~rburke/pubs/burke-umuai02.pdf> >. Acesso entre agosto e novembro de 2013.

DA SILVA, Alexandre H. **SensoRem: um sistema Web para Armazenamento e Análise de Dados de Sensoriamento Remoto Geológico.** Universidade Federal do Rio Grande do Sul, 2012.

JOHNSON, Brian; SHNEIDERMAN, Ben; **Tree-Maps: a space-filling approach to the visualization of hierarchical information structures.** VIS '91 Proceedings of the 2nd conference on Visualization '91. p. 284-291, 1991.

KRASNER, Glenn E.; POPE, Stephen T. **A Cookbook for Using the Model-View-Controller User Interface Paradigm in Smalltalk-80.** Journal of Object-Oriented Programming, v. 1, n. 3, p. 26-49, 1988.

LIU, Bing. **Web Data Mining: Exploring Hyperlinks, Contents and Usage Data.** 2. ed. Springer, 2011.

NASCENTES, Antenor; **Dicionário da Língua Portuguesa da Academia Brasileira de Letras.** Rio de Janeiro: Bloch Ed., 1988.

NICHOLS, David. M. **Implicit Rating and Filtering.** Proceedings of the Fifth DELOS Workshop on Filtering and Collaborative Filtering, Budapeste, Hungria: ERCIM, 1998, p. 31-36. Disponível em: < <http://www.ercim.eu/publication/ws-proceedings/DELOS5/nichols.pdf> >. Acesso em: novembro de 2013.

O'DONOVAN, John; SMYTH, Barry. **Trust in recommender systems.** In: INTERNATIONAL CONFERENCE ON INTELLIGENT USER INTERFACES, 2005,

San Diego. Proceedings of the 10th international conference on Intelligent user interfaces. Nova York: ACM Press, 2005, p. 167-174.

REATEGUI, Eliseo Berni; CAZELLA, Sílvia César. **Sistemas de Recomendação**. XXV CONGRESSO DA SOCIEDADE BRASILEIRA DE COMPUTAÇÃO, 2005, São Leopoldo. Anais do Encontro Nacional de Inteligência Artificial. São Leopoldo, 2005, p. 306 - 348.

RESNICK, Paul; VARIAN, Hal R.; **Recommender Systems**. Magazine Communications of the ACM, v.40, n. 3, p. 56-58, 1997.

SOMMERVILLE, I. **Software Engineering**. 9. ed. Boston: Addison-Wesley, 2011.

THOMAS, James J.; COOK, Kristin A.; **Illuminating the Path: The research and Development Agenda for Visual Analytics**. IEEE CS Press, 2005.

TITTEL, Ed. **Teorias e problemas de XML**. Porto Alegre: Bookman, 2003.

VLIEGEN, R.; VAN WIJK, J.J.; VAN DER LINDEN, E.-J. **Visualizing Business Data with Generalized Treemaps**. Visualization and Computer Graphics, IEEE Transactions on, v. 12 , n. 5, p. 789-796, 2006.

VOZALIS, Emmanouil; MARGARITIS, Konstantinus G. **Analysis of recommender systems algorithms**. Proceedings of the Sixth Hellenic-European Conference on Computer Mathematics and its Applications - HERCMA 2003, 2003.

W3C. **Document Object Model (DOM) Level 3 Core Specification**. Disponível em < <http://www.w3.org/TR/DOM-Level-3-Core/> >. Acesso em dezembro de 2013.

WINESETT, Jeffrey. **Web Application Development with Yii and PHP**. 2. ed. Birmingham: Packt Publishing, 2012.

YII. **The definitive guide**. Disponível em < <http://www.yiiframework.com/doc/guide/> >. Acesso entre junho e novembro de 2013.

Referências dos produtos utilizados

BOOTSTRAP. **Bootstrap**. Disponível em < <http://getbootstrap.com/2.3.2/> >

DEVELOPERS FACEBOOK. **Developers Facebook**. Disponível em: < <https://developers.facebook.com> >

ISOTOPE JS. **Isotope**. Disponível em: < <http://isotope.metafizzy.co/> >

ISOTOPE-PERFECTMASONRY. **Isotope-perfectmasonry**. Disponível em: < <https://github.com/zonear/isotope-perfectmasonry> >

MYSQL. **MySQL Database**. Disponível em < <http://www.mysql.com/> >

MYSQL. **MySQL Workbench**. Disponível em < <http://www.mysql.com/products/workbench/> >

SIMPLEHTMLDOM. **PHP Simple HTML DOM Parser**. Disponível em < <http://www.yiiframework.com/extension/simplehtmlDOM> >

YII. Yii Framework. Disponível em < <http://www.yiiframework.com/> >