

Porto Alegre Online com Sistema de Conteúdo Colaborativo

Gustavo Schonarth, Franz Figueroa

Curso de Análise e Desenvolvimento de Sistemas
Faculdade de Tecnologia Senac RS (FATEC/RS)
Porto Alegre – RS – Brasil

gustavo@schonarth.com.br, franzfigueroa@gmail.com

Resumo. *Este artigo descreve o trabalho envolvido na criação de um sistema de publicação de conteúdo com elementos de rede social para um website já existente, desde a reformulação do conceito do site original até a implementação da solução proposta. Benefícios e problemas surgidos como consequência de decisões de projeto são elencados, apontando elementos que podem contribuir para o sucesso ou fracasso de um projeto, dependendo da maneira e da experiência prévia com que são abordados.*

1. Introdução

Durante a última década, a web tem se tornado um espaço cada vez mais social. As redes sociais cresceram, tornando-se uma das maiores partes da internet (BELL, 2009). Começaram com os programas e sites que permitiam conversar em tempo real (IRC, chat) ou não (fóruns) e aos poucos foram ganhando notoriedade. Serviços inovadores que passaram a ser conhecidos como Web 2.0, como o Flickr e o Delicious, começaram a oferecer ferramentas para trocar, respectivamente, imagens e endereços eletrônicos, com o diferencial de não apenas listar o conteúdo disponível, mas também coletar informações sobre os mais acessados e colocá-los em destaque, baseado na sua popularidade – a *inteligência coletiva* que, segundo Deitel (2008), “é o conceito de que a colaboração pode resultar em ideias inteligentes. Trabalhando juntos, os usuários combinam seu conhecimento para o benefício de todos”.

1.1.O problema

Porto Alegre Online é um site de conteúdo local baseado em HTML estático. Criado em 2007 com o intuito de “celebrar a charmosa capital gaúcha para todo o mundo” (MADEIRA, 2007) e sem fins lucrativos, o site foi construído originalmente de forma artesanal, sem fazer uso de tecnologias como banco de dados para seu conteúdo, que por isso mesmo permitia-se ser não-estruturado.

Visto que a manutenção do site depende apenas de seu criador, optou-se por adotar elementos de redes sociais para a nova versão do site, permitindo uma participação maior dos leitores.

1.2.Solução proposta

O site, disponível no endereço www.portoalegreonline.net, apresenta um conteúdo coletado e organizado por seu idealizador, Fernando Madeira, focado em apresentar a capital gaúcha para a comunidade portoalegrense e também para interessados em qualquer parte do mundo.

A intenção da nova versão do Porto Alegre Online é de trazer novas tecnologias que não apenas facilitem a manutenção do conteúdo pelo administrador do site, mas também incentivem leitores e visitantes a participarem, elegendo os artigos mais populares e também contribuindo com conteúdo próprio. Por fim, o modelo de rede social deverá permitir que o site venha a ser gerido pela própria comunidade, uma vez que os colaboradores mais ativos podem vir a se tornar moderadores após conquistarem determinado número de pontos por quantidade e relevância do conteúdo contribuído (ver 2.5.1 Promoção de moderadores).

Para que o conteúdo do site seja facilmente encontrado, cada artigo poderá receber marcadores, também conhecidos como *tags*, que são “apenas uma palavra em um item de conteúdo. A palavra significa algo para quem a escreve” (BELL, 2009), atribuindo significados ao conteúdo que, combinados, podem descrevê-lo de maneira estruturada, fácil de ser recuperada.

Além disso, de acordo com Bell (2009), o software social cria muitas listas, mas “algumas das maneiras mais óbvias de mostrá-las não são as ideais por uma série de razões. Idealmente, as listas devem mudar com o passar do tempo.” O desafio principal na produção de um site de conteúdo baseado no modelo de rede social está no ordenamento das listas do conteúdo produzido, colocando os artigos mais populares em destaque, sem entretanto esconder os demais, impedindo-os de receber quaisquer votos. “Simples listas dos tags mais populares de um site tendem a tornarem-se bastante estáticas depois que se tem um nível decente de tráfego” (BELL, 2009). Com isso, as próprias listagens dos artigos devem ser dotadas de alguma inteligência para determinar a ordenação dos mesmos.

Visando resolver esse problema, o site agregador de links Linkibol.com publicou em seu blog um artigo intitulado “Como construir um algoritmo de popularidade do qual você possa se orgulhar” (ÖZÇELİK, 2009), no qual utiliza a idade de cada artigo para diluir a popularidade do mesmo, criando um equilíbrio entre novos desconhecidos e antigos populares.

Embora essa equação, como foi proposta, não se encaixe perfeitamente nas necessidades de um site de conteúdo como o Porto Alegre Online, a mesma foi adotada como ponto de partida para a equação deste.

2. Desenvolvimento

2.1.Modelagem

O sistema foi projetado segundo os casos de uso apresentados a seguir:

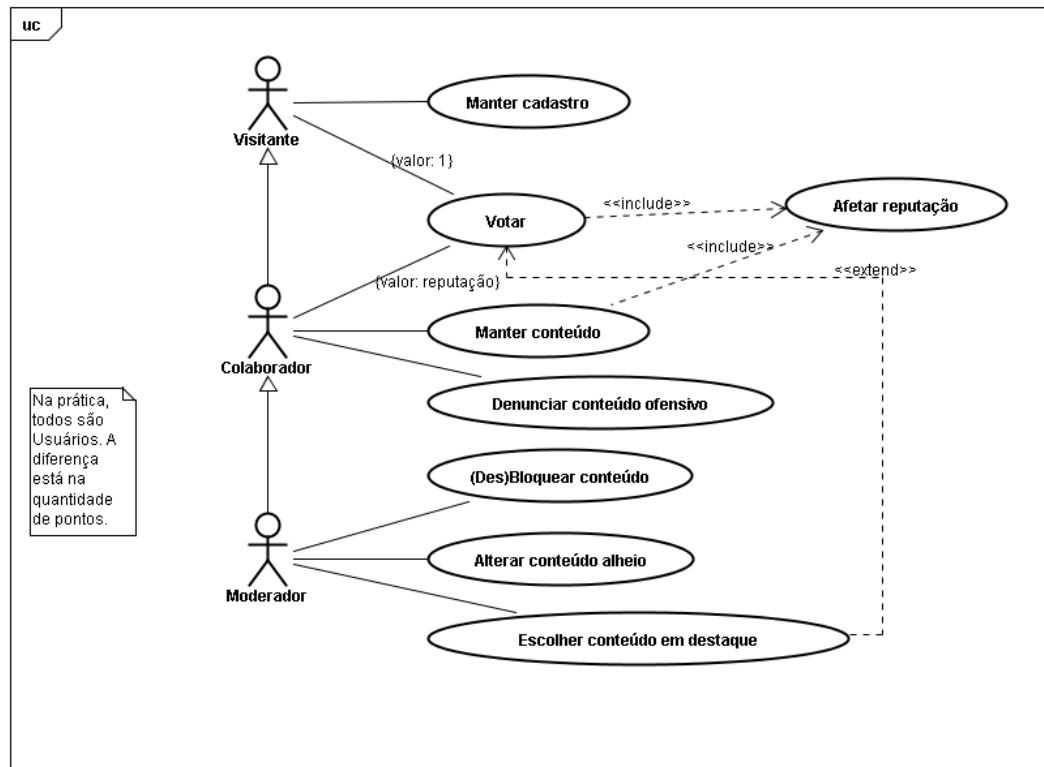


Figura 1. Diagrama de Casos de Uso

Resumidamente, os casos de uso atendem os seguintes requisitos:

- Manter cadastro: Usuário cria ou altera seu cadastro no site.
- Manter conteúdo: Usuário cria ou edita conteúdo de sua autoria.
- Votar: Usuário vota em artigo e/ou comentário para manifestar aprovação.
- Denunciar conteúdo ofensivo: Conteúdo inadequado pode ser denunciado por um colaborador, para que um moderador avalie o caso e decida pela manutenção ou remoção do artigo em questão.
- (Des)Bloquear conteúdo: Permite a moderadores remover conteúdo inadequado, diretamente ou através de denúncias, bem como rejeitar denúncias infundadas.
- Alterar conteúdo alheio: Permite que um moderador altere o documento criado por outro colaborador.
- Escolher conteúdo em destaque: Permite influenciar os artigos que devem aparecer em destaque no menu principal do site
- Afetar reputação: Altera a reputação de um artigo e de seu autor, baseado nos votos recebidos pelo artigo.

2.2.Tecnologia adotada

Visto que o site original já se encontrava hospedado em um servidor que dispõe desta tecnologia, a dupla PHP e SQL para páginas dinâmicas e banco de dados, respectivamente, foi uma escolha já esperada, por serem sistemas de software livre, amplamente disponíveis e suportados, além de serem tecnologias popularmente reconhecidas como práticas e versáteis.

Optou-se também por utilizar um *framework* para atuar como camada de abstração para o banco de dados e fornecer a mecânica básica de segurança e navegação. Por oferecer uma arquitetura baseada em MVC semelhante à do *Ruby on Rails*¹, prometendo maior produtividade por favorecer a convenção à configuração, o *framework* escolhido foi o CakePHP.

2.3.Prática

O desenvolvimento prático teve início com a exploração dos recursos do *framework* CakePHP, através de tutoriais que instruíram na criação de páginas simples de conteúdo, porém já vinculando-as ao banco de dados. Recursos de segurança como lista de controle de acesso também foram fornecidas pelo *framework*, permitindo que uma versão rudimentar do site estivesse em funcionamento em pouco tempo.

O suporte a tags, por ser um elemento fundamental da navegação do site, foi adicionado em seguida, mesmo sem o recurso de adicionar as tags diretamente em cada artigo – foram adicionadas inicialmente diretamente no banco de dados. Com algumas tags atribuídas a cada artigo, foi possível começar a trabalhar em elementos como a Nuvem de Tags e o menu principal, ambos alimentados dinamicamente pelas tags.

2.3.1. Componentes

O *framework* CakePHP, por ser modular, favorece o uso de componentes externos como plug-ins na personalização do seu funcionamento. Essa arquitetura mostrou-se fundamental no desenvolvimento do Porto Alegre Online, e um dos grandes benefícios do *framework* além dos já citados, como contraponto à sua maior desvantagem, que foi observada durante o trabalho de desenvolvimento do site (ver 2.6 Problemas encontrados).

Pela sua própria natureza, o CakePHP assume a camada mais baixa da interação entre o usuário e o PHP, construindo uma plataforma sobre a qual o desenvolvedor escreve sua solução. Isso, contudo, apesar das vantagens elencadas acima, limita o desenvolvedor a apenas programar de acordo com o *framework*, o que o impede de utilizar técnicas dominadas anteriormente. Devido à curva elevada de aprendizagem para a realização de tarefas mais sofisticadas, o desenvolvedor do site recorreu a utilizar *plugins* disponíveis na internet, adaptando-os para suprir a necessidade do projeto.

O primeiro componente externo utilizado em conjunto com o *framework* CakePHP foi o *behavior* TagCloud, que possibilitou a construção da nuvem de tags presente na maioria das páginas da aplicação.

A nuvem de tags consiste em uma lista de todas as tags associadas a pelo menos um artigo publicado. Esses termos são dispostos em ordem alfabética e compacta, com apenas um espaço separando-os. Visualmente, as tags em maior uso são exibidas em tamanho de texto maior, tornando fácil a identificação dos assuntos mais recorrentes.

¹ “O Ruby on Rails é um *framework* voltado para desenvolvimento de aplicações Web (...) diferente da maioria das linguagens de programação porque se beneficia de muitas convenções para reduzir o tempo de desenvolvimento; convenções que, se seguidas, fazem o *framework* Rails gerar funcionalidade substancial e executar várias tarefas para você”. (DEITEL, 2008)

O componente TagCloud auxiliou na obtenção das quantidades de ocorrências de cada tag e no cálculo da sua relevância relativa, baseado na contagem de cada tag em comparação com as demais. O tamanho do texto de cada tag na tela é um número calculado a partir das contagens mínima e máxima de ocorrências, retornado pelo componente juntamente com o próprio termo.

De forma semelhante, já tendo como obter as tags gravadas no sistema, foi utilizado um componente de menus baseado na biblioteca de JavaScript MooTools para construir um menu dinâmico. Construído a partir de duas listas não-numeradas simples em HTML, o componente de JavaScript, utilizando folhas de estilo, oculta a lista secundária e só exibe os itens na mesma que estão relacionados ao item selecionado da lista principal.

Além do componente responsável pelo dinamismo imediato do menu, o mesmo também é dinâmico do ponto de vista da geração de conteúdo – constam ali links para artigos publicados no site, associados a tags predeterminadas e devidamente aprovados por um moderador. Os passos são como se segue, sem uma ordem definida:

- Um colaborador publica um artigo, atribuindo-lhe as tags que considerar apropriadas
- Um moderador cria um item de menu, que deverá aparecer na lista principal do menu, escolhendo a tag pela qual este item deverá filtrar o conteúdo
- Um moderador considera que o artigo é relevante e decide que o mesmo pode ser colocado em destaque no menu principal do site, marcando-o como aprovado

O artigo aprovado, se estiver marcado com a tag correspondente a um item do menu, será automaticamente exibido abaixo do mesmo, em todas as páginas do site, efetivamente promovendo o artigo de forma relevante dentro da estrutura do site.

Para criação e edição dos artigos e comentários no site, foi utilizado o componente CKEditor, reconhecido editor de *rich text*. O editor permite formatar texto, inserir imagens e outros elementos, mesmo que o usuário não domine o HTML, linguagem de marcação padrão na internet. Com isso, os colaboradores não precisam ter conhecimento técnico para escrever no site, sendo-lhes permitido concentrar-se apenas no conteúdo.

Para melhorar o desempenho da aplicação, foram tomadas medidas para reduzir a frequência de certos cálculos complexos e consultas repetidas ao banco, que na maioria das vezes retornam os mesmos valores. Para isso, foi utilizado o componente de *cache* disponível no CakePHP, que permite armazenar temporariamente valores pré-calculados de modo a não precisar recalculá-los a cada vez.

Os componentes que se beneficiam do *cache* são o menu e a nuvem de tags, presentes em todas as páginas, dependentes de muitos cálculos e consultas ao banco, e que tendem a se modificar lentamente. Cada vez que o método responsável pela geração destes for chamado, primeiramente ele procurará pelo valor armazenado no *cache*, retornando esse se for encontrado, e apenas em caso contrário procedendo com as consultas necessárias para criar o conteúdo novamente.

Esses *caches* são limpos a cada vez que haja uma alteração de conteúdo no site, potencialmente alterando os valores calculados.

2.4.O cálculo da popularidade

Talvez o fator mais importante de um site de conteúdo, a escolha dos artigos da primeira página pode ser determinante no sucesso ou fracasso do site, ao promover artigos muito ou pouco relevantes aos seus visitantes.

De acordo com Deitel (2008),

*Embora a colaboração possa resultar em uma abundância de conhecimento, alguns usuários podem submeter informações falsas ou errôneas. (...) Apesar de a **moderação** (monitoramento do conteúdo por funcionários) ser algumas vezes necessária, ela consome tempo e é dispendiosa; muitas empresas Web 2.0 baseiam-se na comunidade para ajudar no policiamento de seus sites. Essa **filtragem colaborativa** permite que os usuários promovam material valioso e sinalizem materiais ofensivos ou inadequados.*

Uma maneira simples de implementar filtragem colaborativa é através da votação. Cada artigo no site pode ser votado positiva ou negativamente pelas pessoas que o acessam, influenciando sua popularidade e criando um ranking dos artigos baseado nesse atributo. Entretanto, de acordo com Özçelik (2009), vários problemas podem surgir na tentativa de classificar artigos dessa forma:

- O sistema pode ter deficiências inerentes na ordenação de itens;
- Isto se deve principalmente ao fato de muitos desenvolvedores criarem seus próprios algoritmos ao invés de basear seus cálculos em fórmulas estatísticas reconhecidamente eficazes;
- Haverá pessoas que tentarão enganar o sistema para levar seus próprios artigos para o topo da lista;
- Haverá ineficiências do sistema devido à complexidade computacional.

Visando fazer frente a esses problemas, o algoritmo de cálculo da popularidade será elaborado a seguir, desde sua variação mais simples e suscetível a erros, evoluindo até sua forma final adotada no Porto Alegre Online, a exemplo do artigo de Özçelik.

Em sua forma mais simples, a quantidade de votos equivale à popularidade do artigo.

Popularidade = votos

Esse algoritmo está longe de ser perfeito por ser simplório demais. Supondo que em determinado período de tempo, e.g. um mês, o site esteja muito ativo e cada artigo receba em média 100 votos, e no mês seguinte o movimento decaia, e novos artigos recebam em média apenas 30 votos cada, os artigos mais recentes nunca conseguirão aparecer na primeira página por não serem tão populares.

O próximo passo é considerar a idade de cada artigo. Assim, novos artigos terão preferência sobre artigos mais antigos, mesmo que ambos tenham a mesma quantidade de votos.

$$\text{Popularidade} = \frac{\text{votos}}{\text{idade do artigo}}$$

Esse cálculo não é exatamente muito justo, visto que artigos muito bons podem ser publicados em momentos de baixo movimento do site e ficarem atrás de outros escritos em tempos de maior movimento, embora menos relevantes. Se considerarmos a idade dos votos também, teremos o seguinte, onde v_n é um voto e i_n é a idade desse voto (em minutos, por exemplo):

$$\text{Popularidade} = \left(\frac{v1}{i1}\right) + \left(\frac{v2}{i2}\right) + \dots + \left(\frac{vn}{in}\right)$$

Somando todos os valores de todos os votos parece resolver o problema anterior, porém introduz um novo – um único voto feito em um artigo com mais de um ano de idade valeria o equivalente a mais de 200 votos dados ao artigo na época da sua publicação, trazendo-o à página principal.

Introduzindo a idade do artigo à equação para suavizar o peso de cada voto, temos o seguinte:

$$\text{Popularidade} = \frac{\left(\frac{v1}{i1}\right) + \left(\frac{v2}{i2}\right) + \dots + \left(\frac{vn}{in}\right)}{\text{idade do artigo}}$$

Essa fórmula suaviza os valores calculados acima usando a idade do artigo, o que impede que histórias antigas saltem à página principal, entretanto é computacionalmente ineficiente. Por depender demais do fator tempo, a cada minuto os valores de cada voto, bem como do próprio artigo, precisam ser recalculados, o que torna a fórmula dispendiosa demais para usar em um sistema com recursos limitados.

Descartando esse algoritmo como impraticável, voltamos a algumas considerações mais básicas. Ao utilizarmos apenas votos positivos e negativos, temos o seguinte: se um leitor gostar de um artigo, avalia-o positivamente com um voto “mais”; caso contrário, com um voto “menos”. A classificação de cada artigo passa a ser a quantidade de votos positivos dividida pela quantidade total de votos:

$$\text{Popularidade} = \frac{\text{votos positivos}}{\text{total de votos}}$$

Outro problema se configura com essa abordagem: supondo que um determinado artigo (A) tenha uma popularidade valendo 93%, após receber centenas de votos, e um outro artigo (B) receba apenas um único voto, positivo, B imediatamente saltará para a primeira colocação, pois seu escore é de 100%. Para evitar esse comportamento, é preciso considerar a quantidade de votos como representativo da

inteligência coletiva, conferindo maior relevância aos artigos que receberem mais votos, o que nos leva à próxima equação.

O tipo de cálculo capaz de considerar as quantidades de votos relativas entre os artigos é conhecido como “média bayesiana” e advém do Teorema de Bayes, que

se propõe a caracterizar a aprendizagem com a experiência, isto é, a modificação da atitude inicial em relação aos ‘antecedentes’, ‘causas’, ‘hipóteses’ ou ‘estados’ depois de ter a informação adicional de que certo acontecimento ou acontecimentos se realizaram (depois de conhecer os dados da experiência ou da observação). (PAULINO et al, 2003)

Este cálculo vale-se de uma informação adicional, quantitativa, para determinar o quão confiável é o valor qualitativo em mãos; esse nível de confiança afeta o peso dado aos votos de acordo com a quantidade de votos que o artigo recebeu.

Assim, a classificação bayesiana dos artigos é calculada da seguinte forma:

$$\text{class_bayes} = \frac{(C * \text{class_media}) + (\text{num_votos_item} * \text{votos_pts_item})}{(C + \text{num_votos_item})}$$

Onde:

- C: é uma constante *ad-hoc*. Se for alta, requererá mais votos para que a classificação ajustada (suavizada) de um item aproxime-se do seu valor original e não-ajustado.
- class_media: classificação média de cada item (apenas os que tiverem pelo menos um voto)
- num_votos_item: número de votos conferidos ao item em questão
- votos_pts_item: número de votos positivos conferidos ao item em questão

A constante C deve ser ajustada à realidade do site, levando em consideração a quantidade média de votos que cada artigo recebe. Não faz sentido exigir mil votos para alterar significativamente a classificação de um item se cada um costuma receber apenas alguns votos, portanto nesse caso o valor de C deveria ser relativamente baixo.

2.4.1. O algoritmo usado no Porto Alegre Online

Para calcular a popularidade de cada artigo no site, foi empregada a média bayesiana explanada acima, com as considerações adicionais a seguir, visando adequá-lo à realidade do site.

A fórmula apresentada funciona estatisticamente melhor que a anterior (i.e. suavizando os votos de acordo com suas idades). Enquanto aquele algoritmo depende muito do fator tempo, esse não depende dele em nada. Pode-se melhorá-lo ainda mais, adicionando por exemplo elementos como a quantidade de visitas do artigo, o próprio efeito do tempo e a reputação dos leitores votantes.

Para considerar as reputações dos usuários, ao invés de simplesmente contarmos a quantidade dos votos positivos, cada voto positivo carregará o valor da própria reputação do usuário, de modo que a contagem considerada será o somatório desses valores. Votos negativos simplesmente valem zero.

Dessa forma, o algoritmo desenvolvido nesse trabalho pode ser expresso da seguinte maneira:

$$\text{Popularidade} = \left(\frac{\left(\frac{(C * \text{class_media}) + (\text{num_votos_item} * \text{votos_pts_item}) + (\text{visitas} * \text{media_visitas})}{C + \text{num_votos_item}} \right)}{\text{idade}^{0.1}} \right)$$

Onde:

- C: constante *ad-hoc*. Atualmente, seu valor está fixo em 10.
- class_media: classificação média de cada item (apenas os que tiverem pelo menos um voto)
- num_votos_item: número de votos conferidos ao item em questão
- votos_pts_item: somatório de todos os valores (reputações) atribuídos ao item em questão
- visitas: contagem de visitas ao artigo
- media_visitas: quantidade média de visitas entre todos os artigos do site
- idade: idade do item em questão, contada em dias

É notável que esse cálculo também exige um considerável poder de processamento, e realizá-lo a cada vez que o artigo fosse listado ou exibido, assim como na tentativa de suavizar os pesos dos votos pela idade do artigo, acima, seria inviável. Nesse caso, contudo, apesar de o algoritmo considerar a ação do tempo como suavizador da classificação de um item, esta tem um efeito muito menor que no caso em que apenas a idade era considerada. Ademais, as idades agora são contadas em dias, permitindo que, no pior dos casos, seja necessário calcular cada idade no máximo uma vez por dia.

Ainda assim, como forma de minimizar essa necessidade de processamento, os valores relativos às popularidades são armazenados junto dos artigos, permitindo recalculá-los a cada vez que o artigo fosse listado ou exibido, assim como na tentativa de suavizar os pesos dos votos pela idade do artigo, acima, seria inviável. Nesse caso, contudo, apesar de o algoritmo considerar a ação do tempo como suavizador da classificação de um item, esta tem um efeito muito menor que no caso em que apenas a idade era considerada. Ademais, as idades agora são contadas em dias, permitindo que, no pior dos casos, seja necessário calcular cada idade no máximo uma vez por dia.

2.4.2. Outras considerações

Não basta apenas criar um algoritmo de popularidade sem protegê-lo de efeitos externos. É preciso que ele seja resistente à influência de usuários determinados a manipular sua posição na lista. Um bom ponto de partida é determinar que os criadores de determinado conteúdo não consigam afetar a popularidade dos seus próprios artigos (BELL, 2009).

No Porto Alegre Online, como forma de prevenir manipulações mal-intencionadas, as seguintes restrições são aplicadas:

- Visitas a cada artigo são contabilizadas apenas uma vez por número IP. Mesmo que isso venha a mascarar vários visitantes acessando através do proxy de uma

instituição, a alternativa de contar simplesmente cada acesso dá maior margem à manipulação.

- Apenas um voto por leitor: cada leitor pode votar apenas uma vez em determinado artigo; votos de leitores anônimos serão limitados com base no endereço IP de forma semelhante à citada acima.
- Votos de leitores anônimos valem apenas 1 ponto.
- Autores não votam em seus próprios artigos.

2.5.O cálculo da reputação

Assim como a popularidade de cada artigo, a reputação de cada usuário é refletida a cada votação. Esta ocorre indiretamente, pois é calculada a partir do voto no artigo, e não no próprio autor. Valendo-se novamente do cálculo da média bayesiana, a reputação dos colaboradores é obtida a partir da seguinte fórmula, derivada daquela:

$$\text{Reputação} = \frac{(C * \text{rep_media}) + (\text{num_votos_user} * \text{votos_pts_user} * \text{num_artigos_user})}{(C * \text{num_votos_user})}$$

Onde:

- C: constante *ad-hoc*. Atualmente, seu valor está fixo em 100.
- rep_media: reputação média de cada usuário
- num_votos_user: número de votos conferidos aos artigos do usuário em questão
- votos_pts_user: somatório de todos os valores (reputações) atribuídos ao usuário em questão
- num_artigos_user: quantidade de artigos que o usuário já publicou

Nesse caso, diferentemente do cálculo da popularidade de artigos, ao invés de suavizarmos o valor pela idade do artigo em questão, a quantidade de artigos publicados pelo usuário é considerada como representativa da sua participação no site, efetivamente valorizando os autores que publicarem mais artigos relevantes. A tabela a seguir simula vários cálculos de reputações, baseadas em votações e quantidades de artigos diferentes, em ordem crescente de reputação alcançada:

Tabela 1. Simulações de cálculos de reputação

num_artigos_user	num_votos_user	votos_pts_user	rep_media	C	Reputação
1	3	9	3	100	1,09
10	10	10	3	100	1,30
1	2	6	3	100	1,56
1	1	3	3	100	3,03
10	10	30	3	100	3,30
10	10	50	3	100	5,30
20	200	100	3	100	20,02
20	100	200	3	100	40,03
20	250	750	3	100	150,01
10	1.000	3.000	3	100	300,00
50	500	1.000	3	100	500,01
100	200	600	3	100	600,02
50	500	2.000	3	100	1000,01
50	1.000	3.000	3	100	1500,00

2.5.1. Promoção de moderadores

Colaboradores que obtiverem reputação acima de dez vezes o valor de C (neste caso, 1.000 pontos) passam a ser moderadores no ato do voto que os fizer alcançar esse valor, efetivamente sendo adicionados ao grupo *Moderadores* da lista de controle de acesso do site.

2.6.Problemas encontrados

O desenvolvimento do projeto Porto Alegre Online não está livre de problemas. O projeto foi iniciado sem um conhecimento prévio do *framework* CakePHP, o que se mostrou um problema devido à curva de aprendizagem do mesmo ser elevada. A maior vantagem anunciada no site oficial do *framework* é “desenvolver aplicações web robustas rapidamente” (CAKEPHP, 2010), entretanto seu uso sem o acompanhamento de um profissional experiente no mesmo resultou em frustrações em vários momentos, decorrentes do seu *modus operandi* particular e, por conta disso, da impossibilidade muitas vezes de utilizar técnicas já conhecidas pelo desenvolvedor, porém incompatíveis com o *framework*.

3. Avaliação

Para validar o trabalho realizado, foi elaborado um questionário e disponibilizado online, tendo sido respondido por 32 pessoas no total, durante o período de três dias (ver dados tabulados em anexo).

Para manter a credibilidade da validação, foi feita inicialmente uma série de questões pessoais, visando obter o perfil dos participantes. Estes se mostraram majoritariamente estar na faixa etária entre 22 e 39 anos, ter educação superior e trabalhar na área de desenvolvimento de software.

Em seguida, os participantes foram solicitados a realizar as tarefas mais comuns no site – navegar, encontrar conteúdo, realizar cadastro, votar em artigos, comentar artigos – e indicar o nível de dificuldade encontrado em cada um. Na sua grande maioria, os pareceres foram favoráveis, com apenas uma pequena parcela apontando dificuldades para escrever novos artigos ou encontrar artigos específicos.

A seguir, foram feitas perguntas acerca da percepção do participante quanto ao site. A maioria dos votos afirma que a proposta do site é clara, que ele aparenta ser capaz de atingir seus objetivos, e que voltariam a utilizar o site no futuro, quando houvesse interesse no assunto proposto pelo mesmo.

Por fim, os participantes foram solicitados a responder de forma discursiva três perguntas de caráter livre. Os itens mais recorrentes para cada pergunta são listados a seguir:

- **Qual sua maior dificuldade?**
 - Navegação (menu, tags)
 - Alguns termos em inglês
 - Falta um campo de busca
 - Página inicial com temas misturados
- **Pontos de melhoria**
 - Design gráfico
 - Menu
 - Busca
 - Localização (L10N)
- **Novas utilidades**
 - Produção de conteúdos específicos/notícias:
 - Política
 - Agenda cultural
 - Colunas
 - Educação
 - Turismo
 - Vincular a mapas/sites de relacionamento

4. Conclusões

Ao realizar este trabalho, concluiu-se que os elementos sociais e colaborativos enriquecem o conteúdo de um site, onde o modelo de comunicação deixa de ser o tradicional, com um transmissor e muitos receptores, e passa a ser o de muitos transmissores/receptores. Nesse modelo, a inteligência coletiva influencia o conteúdo do site, para benefício de todos, deixando de haver o controle de um pequeno grupo em benefício próprio.

Equacionar relacionamentos entre pessoas e seu trabalho, aqui expressos na forma dos cálculos de reputação e popularidade, é uma tarefa complexa. Apesar de considerar uma quantidade relativamente pequena de variáveis, a equação estabelece relações dinâmicas de forma a levar em conta não apenas a opinião das pessoas acerca do que estão votando, mas também julgar o quanto se pode confiar nesses votos, baseado no tamanho da amostra.

Também se pode concluir, numa observação sobre a mecânica do trabalho, que confiar cegamente na promessa de um *framework* é uma premissa arriscada. O *framework* oferece produtividade, mas em troca exige disciplina e o comprometimento do desenvolvedor em seguir seu modo de trabalho, ameaçando com a própria produtividade o desenvolvedor que não esteja bem assistido, seja de mentores já experientes ou documentação farta.

O projeto Porto Alegre Online ainda pode, também, receber melhorias futuras, as quais podem fazer parte de um novo trabalho nos moldes desse ou simplesmente visando seu continuado desenvolvimento. Durante o desenvolvimento, surgiram ideias de itens que não foram implementadas dentro do escopo inicialmente proposto e poderiam ou deveriam ser desenvolvidos no futuro. De forma semelhante, a pesquisa de validação, como se vê no item 3 Avaliação, apontou elementos que deveriam ser melhorados. Os principais itens são enumerados a seguir:

- **Caixa de busca** – como apontado pela pesquisa, faltou uma busca como forma mais rápida e prática de localizar artigos específicos no site.
- **Denúncia de conteúdo impróprio** – o caso de não foi desenvolvido, e deveria voltar ao escopo na próxima fase do site.
- **Combinações de tags** – ao permitir que o visitante navegue pelo site combinando diferentes tags, listas personalizadas de artigos surgirão, permitindo um refinamento do conteúdo ao gosto de cada pessoa.
- **Menu** – poderia ser melhorado, funcionando de forma mais intuitiva; além disso, assim como no item acima, os itens do menu poderiam ser filtrados utilizando uma combinação de tags, resultando em conteúdo possivelmente mais refinado.
- **Perfil dos colaboradores** – um artigo específico, vinculado diretamente ao perfil de um colaborador, permitindo a este falar um pouco sobre si próprio e sobre os assuntos nos quais pretende escrever no site.
- **Envio de imagens** – apesar de o componente CKEditor permitir incluir imagens no texto, ele não fornece o suporte para enviá-las e salvá-las no servidor. Essa funcionalidade deveria ser desenvolvida, salvando os arquivos enviados em um diretório apropriado, bem como registrando-os em banco de dados e vinculando cada arquivo ao colaborador que o enviou.
- **Dinamização da constante C** – nos cálculos de popularidade e reputação de artigos e usuários, a constante C poderia ser substituída por uma variável que refletisse valores médios entre todos os itens considerados, permitindo que a quantidade de votos necessários para afetar o ranking de determinado item seja ajustada de acordo com os votos já computados no site.
- **Melhoria do cálculo de reputação** – como observado na Tabela 1. Simulações de cálculos de reputação, ao receber o primeiro voto a reputação de um usuário é ligeiramente maior que a média, e nos próximos votos sua reputação invariavelmente *cai*, até acumular alguns votos a mais e começar a subir novamente. Idealmente, essa flutuação não deveria ocorrer, e a fórmula deveria ser corrigida para não permitir isso.
- **Melhorias nos comentários** – estes poderiam receber votos, assim como os artigos, e ser agrupados em forma de discussão.

Por fim, para desenvolver um sistema colaborativo de conteúdo é preciso não apenas um algoritmo eficiente, mas também pessoas colaborando com seu conhecimento e julgamento. Cabe a elas, como influenciadoras diretas do conteúdo do site, fazer com que este seja relevante, tanto buscando a qualidade do conteúdo publicado, quanto estimulando este através dos votos.

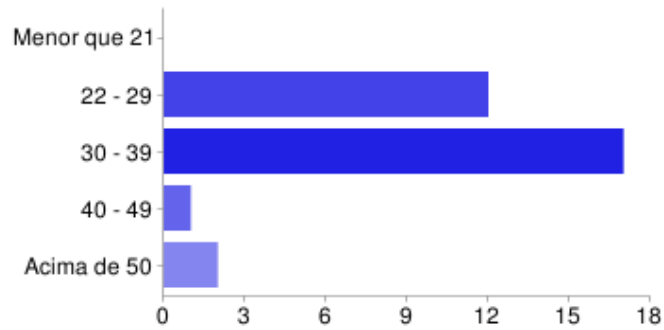
5. Referências

- BELL, Gavin. *Building Social Web Applications*. O'Reilly, Sebastopol, California. 1ª ed, 2009.
- CAKEPHP, Framework de Desenvolvimento Rápido. Disponível em: <<http://www.cakephp.com.br>>. Acesso em: 13 jun. 2010.
- DEITEL, Paul J. e Harvey M. *Ajax, Rich Internet Applications e desenvolvimento Web para programadores*. Pearson Prentice Hall, São Paulo, 2008. Série do Desenvolvedor.
- MADEIRA, Fernando Faganello. *Sobre o Site*. Porto Alegre Online, 2007. Disponível em: <<http://www.portoalegreonline.net/sobreosite.php>>. Acesso em: 12 abr. 2010.
- ÖZÇELİK, Volkan. *How to Build a Popularity Algorithm You can be Proud of*. Linkiblog, 2009. Disponível em: <<http://blog.linkibol.com/post/How-to-Build-a-Popularity-Algorithm-You-can-be-Proud-of.aspx>>. Acesso em: 11 abr. 2010.
- PAULINO, Carlos Daniel, TURKMAN, M. Antónia Amaral, MURTEIRA, Bento. *Estatística Bayesiana*. Fundação Calouste Gulbenkian, Lisboa, 2003.

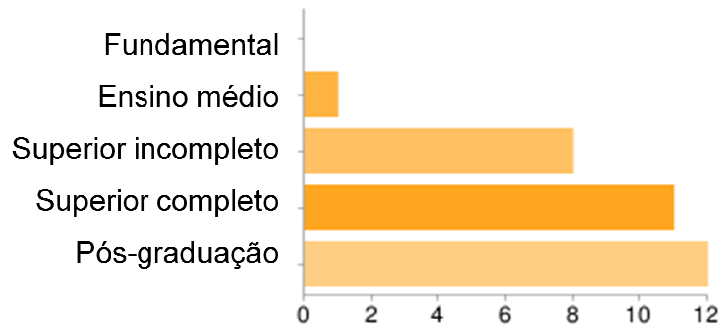
Anexos

Pesquisa de Validação – dados consolidados

Idade



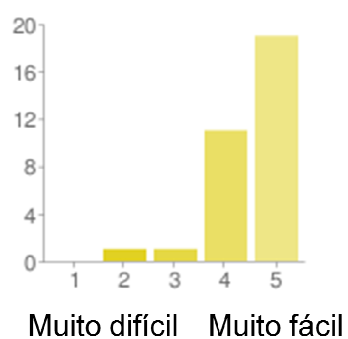
Escolaridade



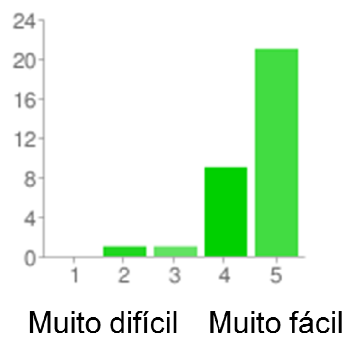
Ocupação



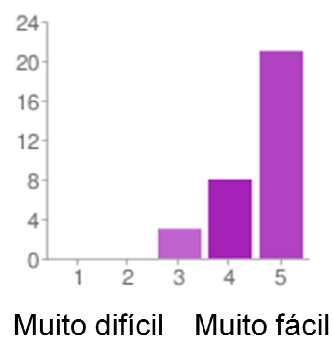
Navegar pelo site



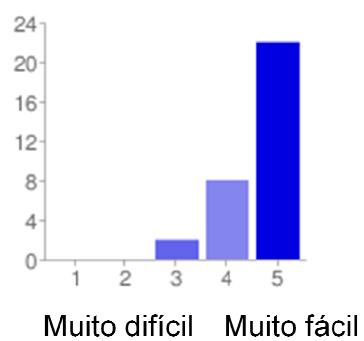
Criar seu cadastro



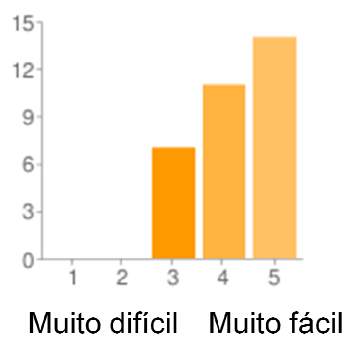
Votar em artigos



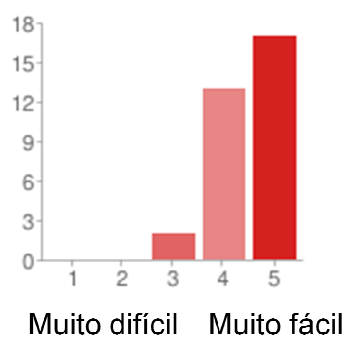
Comentar artigos



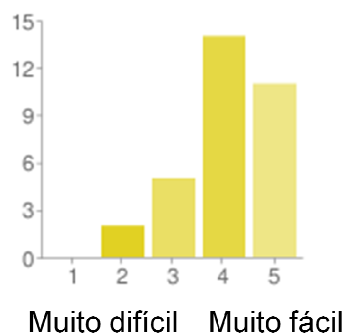
Escrever novos artigos



Formatar texto

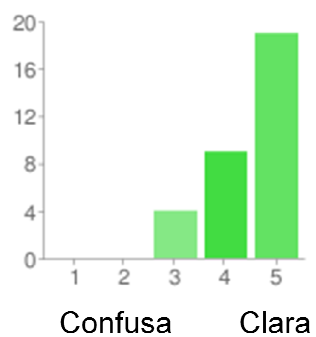


Encontrar artigos específicos



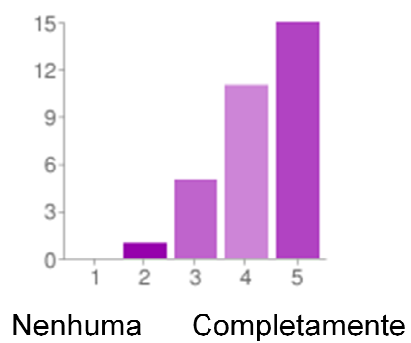
A proposta do site é:

O site apresenta de maneira clara seus objetivos?



Eficácia do site

Quanto o site aparenta ser capaz de atingir seus objetivos?



Utilidade do site

Tendo interesse no assunto, você voltaria a utilizá-lo no futuro?

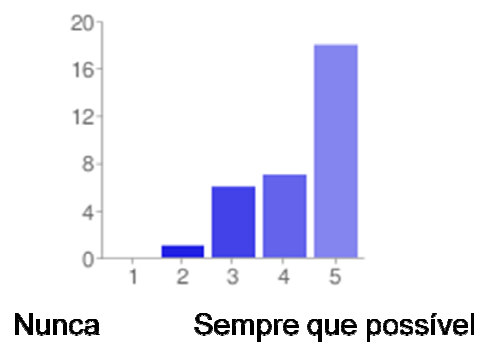


Diagrama ER

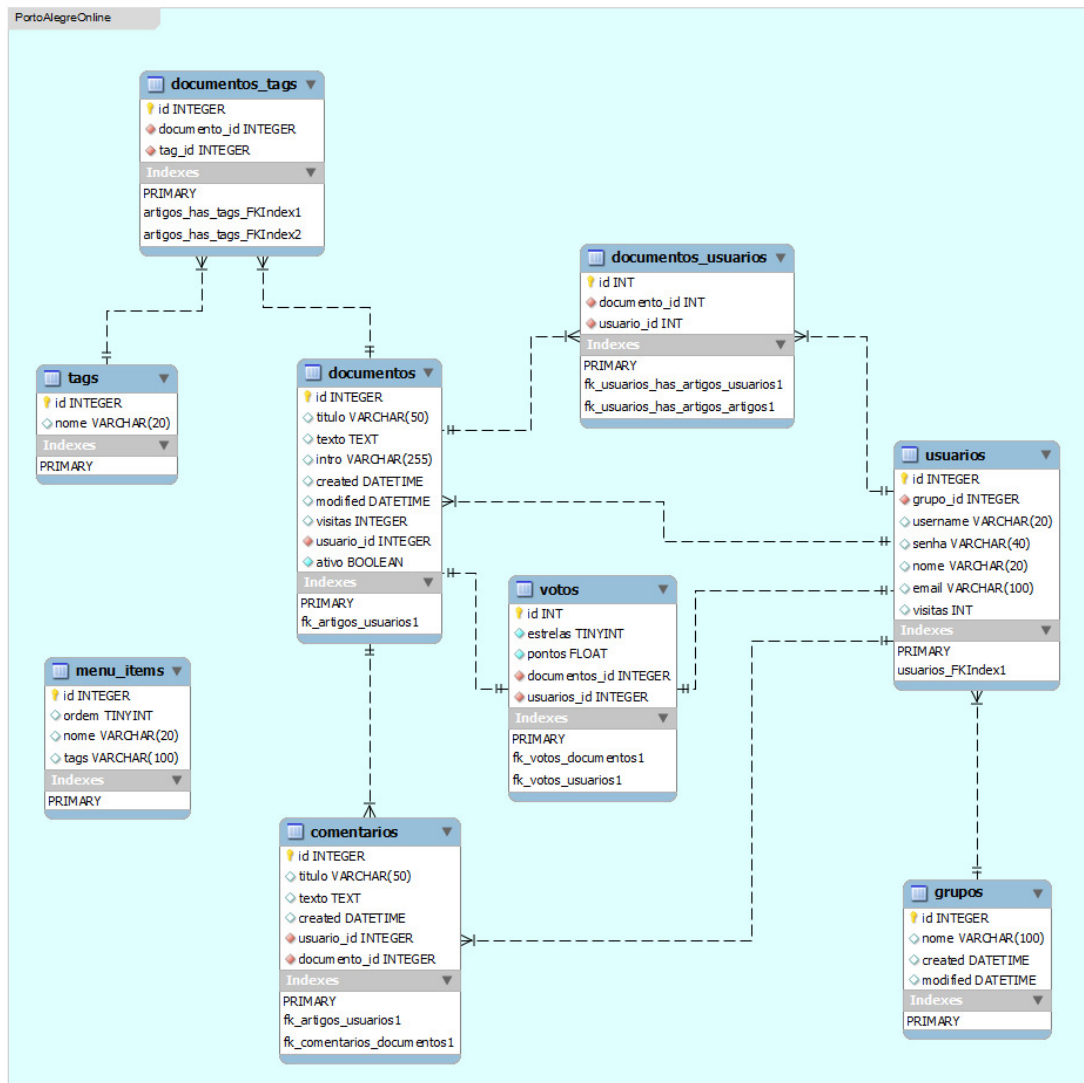


Diagrama de Atividade:

Influência dos votos na popularidade de um artigo e na reputação de um usuário

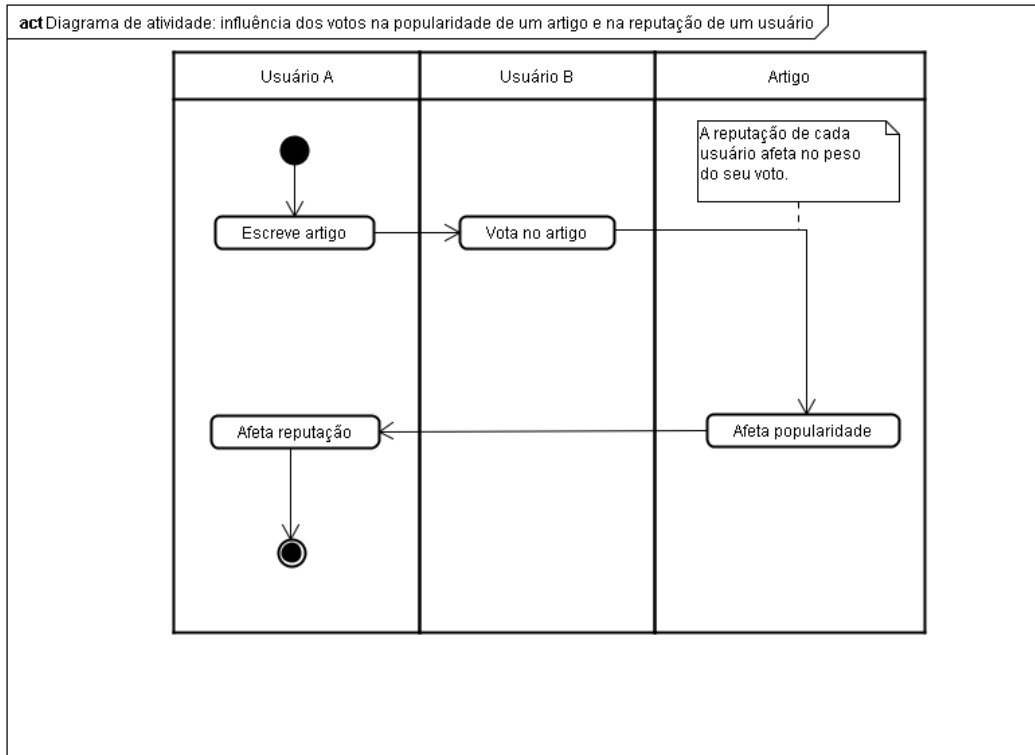
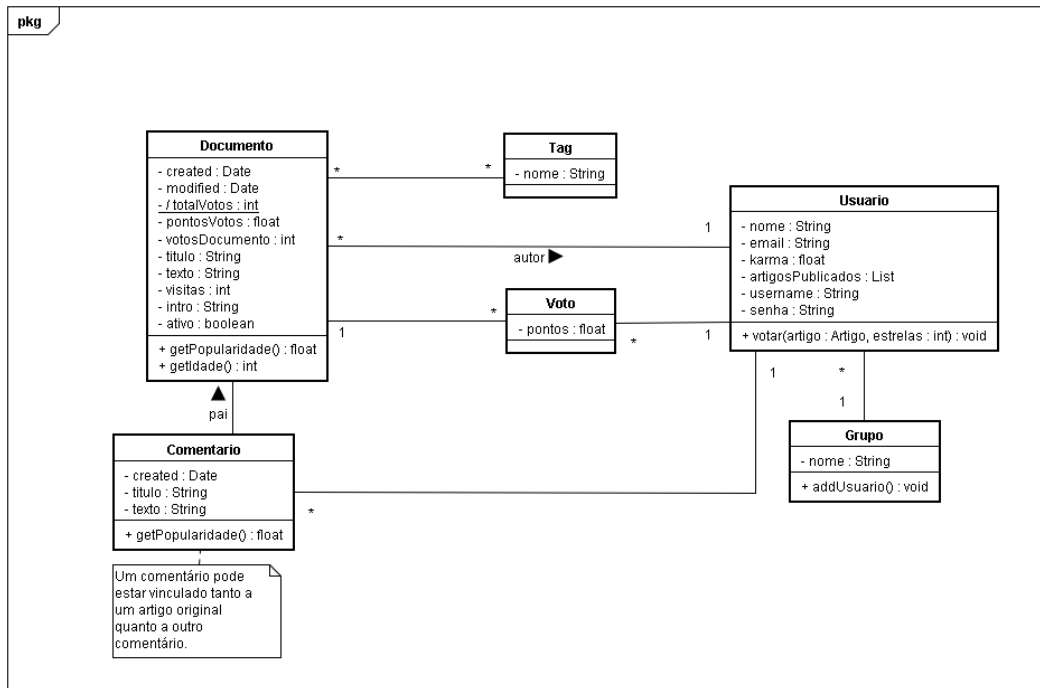


Diagrama de Classes



Especificação de Casos de Uso

1. Manter Cadastro

1.1. Sumário

- Usuário cria ou altera seu cadastro no site.

1.2. Atores

- Visitante
- Colaborador

1.3. Precondições

- Para criar um cadastro: usuário deve exercer papel de visitante.
- Para editar um cadastro: usuário deve exercer papel de colaborador, no mínimo.

1.4. Fluxo ou cenário principal

1.4.1. Criação de cadastro

1.4.1.1. Usuário (visitante) clica no link “Cadastro” na tela principal.

1.4.1.2. O formulário de cadastro apresenta os campos:

- Nome
- E-mail
- Username
- Senha
- Repetir senha

1.4.1.3. O usuário preenche os campos e clica em “Pronto”.

1.4.1.4. O sistema valida e persiste os dados do usuário e retorna à tela principal.

1.5. Fluxo ou cenário alternativo

1.5.1. Edição de cadastro

1.5.1.1. Usuário (colaborador) clica no link “Meus dados” na tela principal.

1.5.1.2. O formulário de atualização de cadastro apresenta os campos, já preenchidos com os valores atuais:

- Nome
- E-mail

1.5.1.3. O usuário altera o que achar necessário e clica em “Pronto”.

1.5.1.4. O sistema persiste os dados do usuário e retorna à tela principal.

1.5.2. Alteração de senha

1.5.2.1. Durante 1.5.1 Edição de cadastro, o usuário clica no link “Trocar minha senha” presente no formulário.

1.5.2.2. Um diálogo modal é apresentado, com os seguintes campos:

- Senha atual
- Nova senha
- Repetir nova senha

1.5.2.3. O sistema valida e persiste os novos dados e fecha o diálogo modal.

1.5.3. Exclusão de cadastro

1.5.3.1. Durante 1.5.1 Edição de cadastro, o usuário clica no link “Excluir meu cadastro” presente no formulário.

1.5.3.2. O sistema apresentará um diálogo modal e solicitará confirmação, avisando que essa ação não pode ser desfeita.

1.5.3.3. O usuário confirma clicando em “Sim, eu quero excluir meu cadastro”.

1.5.3.4. O sistema realiza a exclusão lógica² do usuário.

1.6. Fluxo ou cenário de exceção

1.6.1. Nome de usuário ou e-mail já existe

1.6.1.1. Caso a regra 1.8.3 Usuários únicos seja violada durante 1.4.1 Criação de cadastro, o sistema apresentará um aviso no formulário, solicitando ao usuário que escolha outro nome de usuário e/ou endereço de e-mail, conforme for apropriado. Os dados já digitados não serão perdidos.

1.7. Pós-condições

- Para criação ou edição de cadastro: usuário exerce papel de Colaborador
- Para exclusão de cadastro: usuário exerce papel de Visitante

² Exclusão lógica: os dados não são removidos do sistema, mas passam a ser ignorados deste ponto em diante. Permanecem apenas para permitir consultas relacionadas a artigos criados por este usuário que continuem à disposição no site.

1.8. Regras de negócio

1.8.1. Validação de senha

1.8.1.1. Os campos “Senha” e “Repetir Senha” não podem estar em branco e devem ser preenchidos com exatamente o mesmo valor, de no mínimo 4 caracteres.

1.8.1.2. O mesmo se aplica aos campos “Nova Senha” e “Repetir Nova Senha”; no caso 1.5.2 Alteração de senha, a senha atual deve ser validada com o que está armazenado no cadastro do usuário.

1.8.2. Validação de e-mail

1.8.2.1. O endereço de e-mail deve seguir as seguintes regras:

- Deve conter o caractere “@” (arroba) exatamente uma vez
- Deve conter pelo menos um ponto “.” depois da arroba
- O ponto não pode estar adjacente à arroba (@. ou .@)
- Não pode haver dois pontos consecutivos (..)
- O endereço não pode começar nem terminar com arroba ou ponto

1.8.3. Usuários únicos

1.8.3.1. Não pode haver dois usuários com o mesmo username ou endereço de e-mail

2. Manter Conteúdo

2.1. Sumário

- Usuário cria ou edita conteúdo de sua autoria

2.2. Atores

- Visitante
- Colaborador

2.3. Precondições

- Para criar um artigo, usuário está cadastrado e identificado como um Colaborador

2.4. Fluxo ou cenário principal

2.4.1. Escrever artigo

2.4.1.1. Colaborador clica em “Novo artigo” na página inicial

2.4.1.2. O formulário de novo artigo é apresentado com os seguintes campos:

- Título
- Introdução
- Texto

2.4.1.3. O usuário preenche os campos e clica em “Publicar”

2.4.1.4. O sistema valida os dados e publica o artigo

2.5. Fluxo ou cenário alternativo

2.5.1. Classificar artigos

2.5.1.1. Colaborador navega até a página de um artigo

2.5.1.2. Ao lado do artigo, o sistema exibe os tags atualmente associados ao artigo, bem como um campo para adição de novos tags

2.5.1.3. Colaborador escreve no campo os termos que quiser, separados por vírgula

2.5.1.4. Colaborador tecla em <ENTER> ou clica no botão “Ok” próximo ao campo

2.5.1.5. O sistema adiciona as tags ao artigo

2.5.2. Comentar em artigo

2.5.2.1. Usuário navega até a página de um artigo

2.5.2.2. Abaixo do artigo há um formulário de comentário e uma lista dos comentários já publicados

2.5.2.3. O formulário contém os seguintes campos:

- Título
- Texto

2.5.2.4. Se o usuário não estiver identificado (visitante), dois campos adicionais do tipo *radio button* serão apresentados com a legenda “Identificação”:

- Anônimo
- Fazer login

2.5.2.5. Usuário preenche os campos e clica em “Publicar”

2.5.2.6. Caso o visitante tenha selecionado a opção “Fazer login”, o formulário de login será apresentado antes de prosseguir

2.5.2.7. O comentário é adicionado à lista de comentários desse artigo.

2.5.3. Editar artigo

2.5.3.1. Colaborador navega até a página de um artigo de sua autoria

2.5.3.2. Colaborador clica no link “Editar” presente nessa página

2.5.3.3. O formulário de edição de artigo (idêntico ao de novo artigo (2.4.1.2), porém com os campos preenchidos) é apresentado

2.5.3.4. O usuário preenche os campos e clica em “Publicar”

2.5.3.5. O sistema valida os dados e publica o artigo

2.5.4. Excluir artigo

2.5.4.1. Colaborador navega até a página de um artigo de sua autoria

2.5.4.2. Colaborador clica no link “Excluir este artigo”

2.5.4.3. O sistema apresenta um diálogo modal solicitando confirmação e alertando que esta ação não pode ser desfeita

2.5.4.4. O colaborador confirma clicando em “Quero excluir esse artigo”

2.5.4.5. O sistema exclui o artigo fisicamente. Os dados referentes à popularidade do artigo, bem como seu efeito na reputação do usuário, são eliminados.

2.5.5. Lista “Meus Artigos”

2.5.5.1. A partir da página inicial, colaborador clica em “Meus artigos”

2.5.5.2. O sistema exibe uma lista com todos os artigos criados pelo colaborador identificado. Artigos bloqueados (ver 5 (Des)Bloquear conteúdo) são incluídos nessa lista.

2.5.6. Lista “Mais artigos de...”

2.5.6.1. Em cada página de artigo, entre o artigo e o formulário de comentário o visitante clica em “Mais artigos de [nome do autor]”

2.5.6.2. O sistema exibe uma lista com todos os artigos criados pelo autor do artigo em questão. Artigos bloqueados (ver 5 (Des)Bloquear conteúdo) são excluídos dessa lista.

2.6. Fluxo ou cenário de exceção

2.6.1. Criar artigo com visitante

2.6.1.1. Se um usuário visitante (i.e. sem cadastro ou não identificado) clicar no link “novo artigo”, dois campos do tipo *radio button* serão apresentados com a legenda “Identificação”:

- Cadastre-se
- Fazer login

2.6.1.2. O usuário seleciona um deles e imediatamente o sistema apresenta o formulário apropriado.

2.6.1.3. Após o cadastro ou login, o usuário é levado à tela de novo artigo (2.4.1).

2.7. Pós-condições

- Um novo artigo é publicado, vinculado ao colaborador identificado.

2.8. Regras de negócio

2.8.1. Autor identificado

2.8.1.1. Somente colaboradores identificados podem criar ou editar artigos.

2.8.1.2. Cada colaborador só pode editar artigos de sua própria autoria.

2.8.1.3. Visitantes que não quiserem se identificar podem publicar comentários como “Anônimo”.

2.8.2. Campos obrigatórios (artigo)

2.8.2.1. Na criação de um artigo, os campos “Título” e “Introdução” são de preenchimento obrigatório

2.8.3. Campos obrigatórios (comentário)

2.8.3.1. Na criação de um comentário, ao menos um dos campos (“Título” ou “Texto”) deve estar preenchido

2.8.4. Não-repetição de tags

2.8.4.1. Ao adicionar tags a um artigo, caso essa já exista, não será criada uma nova tag; a tag já existente será usada

2.8.4.2. Ao adicionar tags a um artigo, caso essa já esteja associada ao artigo, não haverá alteração

2.8.5. Participação afeta reputação

2.8.5.1. Cada artigo criado afeta o cálculo da reputação de um colaborador, no quesito quantidade. Ver 8.8.3 Equação da reputação de um colaborador.

2.8.6. Exclusão física

2.8.6.1. Excluir um documento é uma operação de exclusão física, removendo definitivamente o registro do banco de dados.

3. Votar

3.1. Sumário

- Colaborador vota em artigo e/ou comentário para manifestar aprovação

3.2. Atores

- Colaborador

3.3. Precondições

- Usuário deve estar identificado como um colaborador

3.4. Fluxo ou cenário principal

3.4.1. Votar em artigo

3.4.1.1. Colaborador navega até a página de um artigo

3.4.1.2. Próximo ao título do artigo há o espaço para a votação, com 5 estrelas a serem escolhidas

3.4.1.3. O usuário faz sua escolha, clicando da primeira estrela (desaprovo completamente) à última (aprovo completamente)

3.4.1.4. Imediatamente o sistema registra o voto do usuário

3.5. Fluxo ou cenário alternativo

3.5.1. Alterar o voto

3.5.1.1. Um colaborador pode alterar seu voto, navegando até a página do artigo.

3.5.1.2. O processo é idêntico a 3.4.1 Votar em artigo, salvo que as estrelas já estão pintadas refletindo a escolha anterior do colaborador.

3.6. Fluxo ou cenário de exceção

3.6.1. Visitantes não podem votar

- 3.6.1.1. Para evitar fraudes e manipulações, apenas usuários cadastrados e identificados (colaboradores ou moderadores) podem votar em artigos do site.

3.7. Pós-condições

- Um voto é registrado para o artigo em questão

3.8. Regras de negócio

3.8.1. Votar em si próprio

- 3.8.1.1. Um colaborador não pode votar em artigos de sua autoria.

3.8.2. Relevância dos votos

- 3.8.2.1. O voto de cada colaborador leva em consideração a reputação deste no cômputo da popularidade do artigo (ver 8.4.1 Cálculo de popularidade do artigo).

4. Denunciar conteúdo ofensivo

4.1. Sumário

- Conteúdo inapropriado pode ser denunciado por um colaborador, para que um moderador avalie o caso e decida pela manutenção ou remoção do artigo em questão

4.2. Atores

- Colaborador

4.3. Precondições

- Usuário está identificado como colaborador

4.4. Fluxo ou cenário principal

4.4.1. Colaborador denuncia artigo

- 4.4.1.1. Colaborador navega até a página do artigo que considera inapropriado

- 4.4.1.2. Colaborador clica no link “Denunciar este artigo”

4.4.1.3. Um diálogo modal é apresentado ao colaborador, solicitando confirmação da ação e uma justificativa.

4.4.1.4. O colaborador preenche o campo e clica em “Denunciar”

4.4.1.5. O sistema marca o artigo como denunciado e associa-o à identificação do colaborador que fez a denúncia. O artigo continua sendo acessado normalmente pelos usuários do site, mas exibe um aviso quando o usuário for o autor ou um moderador.

4.5. Fluxo ou cenário alternativo

- Não há

4.6. Fluxo ou cenário de exceção

- Não há

4.7. Pós-condições

- O artigo denunciado será marcado como tal e adicionado à lista de denúncias disponível para os moderadores (ver 5.4.1 Revisar denúncias)

4.8. Regras de negócio

4.8.1. Denúncias não afetam o conteúdo

4.8.1.1. Artigos denunciados constarão em uma lista especial para atenção dos moderadores mas, até que sejam avaliados, permanecem à disposição de todos.

4.8.2. Justificativa

4.8.2.1. O campo de justificativa não deve estar vazio.

5. (Des)Bloquear conteúdo

5.1. Sumário

- Permite a moderadores remover conteúdo inapropriado, diretamente ou através de denúncias, bem como rejeitar denúncias infundadas.

5.2. Atores

- Moderador

5.3. Precondições

- Usuário está identificado como moderador

5.4. Fluxo ou cenário principal

5.4.1. Revisar denúncias

5.4.1.1. Moderador clica em “Revisar denúncias” na página inicial

5.4.1.2. O sistema exibe uma lista dos títulos dos artigos denunciados, com o nome do autor e do denunciante, cada qual um link para seu endereço de e-mail. O título de cada artigo é um link para a página do mesmo.

5.4.1.3. Cada artigo tem um checkbox ao lado do seu nome na lista. O moderador pode selecionar os artigos que quiser.

5.4.1.4. Moderador clica em “Bloquear artigos selecionados” ou em “Liberar artigos selecionados”.

5.4.1.5. O sistema bloqueia ou libera (retirando a denúncia) os artigos selecionados conforme a opção escolhida pelo moderador.

5.5. Fluxo ou cenário alternativo

5.5.1. Listar e desbloquear artigos bloqueados

5.5.1.1. Moderador clica em “Artigos bloqueados” na página inicial

5.5.1.2. O sistema exibe uma lista semelhante à de 5.4.1 Revisar denúncias, contendo apenas artigos previamente bloqueados.

5.5.1.3. O moderador pode selecionar os artigos que quiser, nos seus respectivos checkboxes, e clicar em “Desbloquear”

5.5.1.4. O sistema libera os artigos para acesso normal pela comunidade.

5.5.2. Exibir artigo denunciado

5.5.2.1. Ao exibir um artigo denunciado, quando o usuário identificado for um moderador ou o autor do artigo, o sistema exibe um alerta acima do mesmo, avisando que este encontra-se sob denúncia, exibindo o nome, endereço de e-mail e justificativa do colaborador que realizou a denúncia (ver 4.4.1 Colaborador denuncia artigo).

5.5.3. Exibir artigo bloqueado

5.5.3.1. Ao exibir um artigo bloqueado, o sistema exibe um alerta acima do mesmo, avisando que este encontra-se bloqueado e indisponível para usuários além do seu autor ou dos moderadores. Se foi bloqueado mediante denúncia, a mesma identificação descrita em 5.5.2 Exibir artigo denunciado estará visível.

5.5.4. Bloquear artigo

- 5.5.4.1. Além da lista 5.4.1 Revisar denúncias, nas páginas de leitura de artigos estará disponível aos moderadores um botão de “Bloquear artigo”, excluindo-o logicamente.

5.6. Fluxo ou cenário de exceção

5.6.1. Exibir artigo bloqueado

- 5.6.1.1. Visitantes ou colaboradores que não sejam o autor ou um moderador, ao acessar o artigo pela URL, receberão apenas um aviso de que o mesmo está bloqueado.
- 5.6.1.2. Moderadores ou o autor do artigo, ao acessarem o mesmo, poderão ver o artigo, junto do aviso de bloqueado no topo e os dados da denúncia (se houver).
- 5.6.1.3. Moderadores podem clicar em “Desbloquear artigo”
- 5.6.1.4. O autor do artigo pode editá-lo (ver 2.5.3 Editar artigo), mas o artigo permanecerá bloqueado e continuará dependendo da liberação explícita de um moderador.

5.7. Pós-condições

- Os artigos em questão encontram-se bloqueados ou liberados para serem acessados normalmente

5.8. Regras de negócio

5.8.1. Permissões de visualização

- 5.8.1.1. Qualquer usuário pode acessar um artigo denunciado normalmente
- 5.8.1.2. Apenas o autor e os moderadores podem acessar um artigo bloqueado

5.8.2. Exclusão lógica

- 5.8.2.1. Bloquear um artigo equivale a excluí-lo logicamente, isto é, marcá-lo como excluído, entretanto sem removê-lo do banco de dados de fato.

6. Alterar conteúdo alheio

6.1. Sumário

- Permite que um moderador altere o documento criado por outro colaborador

6.2. Atores

- Moderador

6.3. Precondições

- Usuário deve estar identificado como moderador

6.4. Fluxo ou cenário principal

6.4.1. Editar artigo de outro colaborador

- 6.4.1.1. Moderador pode realizar 2.5.3 Editar artigo como se fosse o autor do mesmo; a autoria é mantida, entretanto.

6.5. Fluxo ou cenário alternativo

6.5.1. Excluir artigo de outro colaborador

- 6.5.1.1. Moderador pode realizar 2.5.4 Excluir artigo como se fosse o autor do mesmo

6.6. Fluxo ou cenário de exceção

- Não há

6.7. Pós-condições

- O artigo em questão é alterado ou removido

6.8. Regras de negócio

6.8.1. Apenas moderadores

- 6.8.1.1. Apenas usuários identificados como moderadores podem alterar ou excluir artigos criados por outros colaboradores

7. Escolher conteúdo em destaque

7.1. Sumário

- Permite influenciar os artigos que devem aparecer em destaque na página principal do site

7.2. Atores

- Moderador

7.3. Precondições

- O usuário deve estar identificado como Moderador.

7.4. Fluxo ou cenário principal

7.4.1. Promover artigo

7.4.1.1. Moderador navega até um artigo que deseja promover

7.4.1.2. Na página de leitura do artigo, moderador clica em “Promover”

7.4.1.3. O sistema aumenta automaticamente a popularidade do artigo, colocando-o na página principal do site.

7.5. Fluxo ou cenário alternativo

7.5.1. Promover tags

7.5.1.1. Moderador seleciona tags para serem exibidas na página inicial

7.5.1.2. O sistema registra esses tags e passa a exibir uma área de destaque para artigo(s) com esse tag na página principal, escolhidos automaticamente por popularidade

7.5.2. Montar menu

7.5.2.1. Moderador seleciona tags para serem exibidos no menu principal, bem como um nome para esse item no menu

7.5.2.2. O sistema registra esses tags e passa a exibir todos os documentos que estiverem marcados com todos os tags selecionados no menu, ordenados alfabeticamente

7.6. Fluxo ou cenário de exceção

- Não há

7.7. Pós-condições

- O artigo promovido encontra-se na capa do site e/ou no menu

7.8. Regras de negócio

7.8.1. Critério de seleção para capa e menu

7.8.1.1. Os artigos na capa do site são escolhidos automaticamente baseados na sua popularidade e configurações de tags para a capa. A quantidade exibida é definida em um parâmetro do sistema.

7.8.1.2. Os artigos no menu são todos os que satisfizerem a seleção de tags determinados para cada item do menu.

8. Afetar reputação

8.1. Sumário

- Altera a reputação de um artigo e de seu autor, baseado nos votos recebidos pelo artigo.

8.2. Atores

- Colaborador

8.3. Precondições

- Colaborador vota em um artigo

8.4. Fluxo ou cenário principal

8.4.1. Cálculo de popularidade do artigo

8.4.1.1. Um colaborador C vota em artigo A

8.4.1.2. A reputação de C é considerada como multiplicador na equação que determina o peso do voto, juntamente com a quantidade de estrelas votadas. Ver 8.8.2 Equação da popularidade de um artigo mediante voto.

8.4.1.3. Após contabilizados os votos, a popularidade do artigo deve ser decrementada em um valor calculado a partir da sua idade, de modo a evitar um círculo vicioso que mantenha sempre os documentos mais antigos como mais populares do site, pela mera contagem de votos acumulados.

8.4.1.4. O sistema armazena os valores calculados para uso nas próximas operações de seleção automática de conteúdo. Ver

8.5. Fluxo ou cenário alternativo

8.5.1. Cálculo de reputação do autor

8.5.1.1. O sistema realiza novo cálculo de popularidade de um artigo (ver 8.4.1 Cálculo de popularidade do artigo)

8.5.1.2. A popularidade de cada artigo é levada em consideração no cálculo da reputação do seu autor, juntamente com seu nível de participação no site. Ver 8.8.3 Equação da reputação de um .

8.5.1.3. O sistema armazena a nova reputação do usuário para os próximos cálculos.

8.6. Fluxo ou cenário de exceção

- Não há

8.7. Pós-condições

- O artigo votado tem sua popularidade atualizada
- O autor do artigo tem sua reputação atualizada

8.8. Regras de negócio

8.8.1. Uso indireto

8.8.1.1. Este caso de uso não é realizado diretamente por nenhum usuário do site.

8.8.2. Equação da popularidade de um artigo mediante voto

8.8.2.1. $V = R(C) * EV$, onde:

- V = Voto
- $R(C)$ = reputação do colaborador que está votando
- EV = estrelas votadas

Essa é uma equação rudimentar que apenas ilustra o cálculo a ser desenvolvido durante o desenvolvimento do caso de uso. A premissa é de que a reputação do votante seja considerada juntamente com os pontos atribuídos no voto, dando maior notoriedade aos artigos aprovados pelos colaboradores mais respeitados do site.

8.8.3. Equação da reputação de um colaborador

8.8.3.1. $R(C) = \text{média}(P(A(C))) * \text{desvio}(QP(C))$, onde

- $R(C)$ = reputação do colaborador
- $P(A(C))$ = popularidade dos artigos do colaborador
- $QP(C)$ = quantidade de participações do colaborador

Essa é uma equação rudimentar que apenas ilustra o cálculo a ser desenvolvido durante o desenvolvimento do caso de uso. A premissa é de que a reputação seja uma relação entre a quantidade e a qualidade da produção do colaborador no site, conferindo mais notoriedade aos que participam muito e com qualidade do que aos que participam pouco e com qualidade, entretanto sem dar visibilidade injustamente a usuários que participam muito, porém sem qualidade.

8.8.4. Colaborador torna-se moderador

- 8.8.4.1. Ao atingir determinada pontuação de reputação, a ser definida durante o desenvolvimento do trabalho, um colaborador atinge o status de moderador e passa a ter à sua disposição os recursos elencados anteriormente, limitados aos moderadores do site.