

5

Utilização do sistema de gerenciamento e publicação de conteúdo Gutenberg no *site* da Faperj

Neste capítulo será apresentado o sistema de gerenciamento e publicação de conteúdo Gutenberg e o como foi o processo de decisão de utilização, compra e implementação do Gutenberg no projeto do novo *site* da Faperj - Fundação Carlos Chagas Filho de Amparo a Pesquisa do Estado do Rio de Janeiro.

5.1

História do Gutenberg

Segundo Ariel Nigri (2005), diretor da Winco Tecnologia da Informação, o sistema foi desenvolvido quando trabalhava na Celulloid em 2001 para atender às necessidades de publicação do *site* da Glaxo, indústria do segmento farmacêutico. A equipe era composta pelo Ariel Nigri, engenheiro elétrico, Inácio Correa, programador, e Leonardo Lana, web designer. Como fazia parte do contrato desenvolver um *software* mais barato com o direito de comercialização posterior, Nigri criou a Winco para comercializá-lo e batizou o sistema como Gutenberg.

No início, o Gutenberg vinha acompanhado de um *template* padrão, porém com um design bastante simples, que não se adaptava a qualquer arquitetura da informação. Depois, a Winco ofereceu o serviço de criação e implementação do projeto, caso o cliente não possuísse a sua própria equipe de designers e programadores em linguagem PHP (Hypertext Preprocessor).

Somente foram lançadas duas versões do Gutenberg, sendo que a primeira, adquirida pela Faperj, tinha um código complexo dificultando alguns clientes, como por exemplo, a própria Faperj, de realizar a sua implementação sem a ajuda da Winco. O manual fornecido também era muito inconsistente.

“Na segunda versão, também adquirida pela Faperj posteriormente, o sistema foi reprogramado com um código mais fácil e o manual fornecido foi mais completo, possibilitando qualquer programador de PHP fazer a implementação do sistema”, diz Nigri.

Correções de *bugs* e pequenas melhorias no *software* não são consideradas motivos para gerar uma nova versão, somente as alterações relevantes geram uma nova versão. Nigri explica que as alterações foram documentadas através de um programa de controle de versão, podendo gerar a versão de qualquer dia do passado, tanto por número, quanto por área. É um programa capaz de armazenar códigos fonte e avaliar o que foi alterado de uma versão para outra, muito utilizado por desenvolvedores de *software*.

As empresas que adquiriam o Gutenberg para o gerenciamento de conteúdo dos seus *sites* são:

- Fundação Carlos Chagas Filho de Amparo à Pesquisa do Estado do Rio de Janeiro – Faperj
- Fundação de Amparo à Pesquisa e Extensão Sanitária – Fapeu
- Academia Nacional de Medicina – ANM
- A Linha Amarela S/A - Lamsa
- Colégio Brasileiro de Cirurgiões – CBC
- Winco

Nigri complementa que sabe que faltou treinamento com os usuários do Gutenberg, e reconhece que o sistema tem problemas de usabilidade, além de não ter tido “*um banho de loja antes de ser comercializado*”, se referindo à interface apresentada pelo sistema.

Em 2005, o Gutenberg parou de ser comercializado, pois a Winco preferiu investir em outras tecnologias que trouxessem maior retorno financeiro. A manutenção continua sendo oferecida para os clientes que o utilizam. Porém, não deverá haver lançamento de novas versões.

5.2

Visão geral do sistema

Conforme definição retirada do *site* da Winco (2004), o Gutenberg é um sistema de gerenciamento e publicação de informações para *sites* institucionais, jornais e revistas eletrônicas, intranets e extranets, portais, sistemas eletrônicos de pesquisa, repositório de informações, entre outros. Seu maior objetivo é tornar

simples a atualização de conteúdo na web não sendo necessário conhecimentos específicos em informática para poder usá-lo.

Uma vantagem em relação a alguns concorrentes é não possuir um *template* pré-definido, podendo ser integrado aos mais diversos tipos de projetos, o que permite ao web designer liberdade de criação. Porém, a integração do sistema aos *templates* restringe-se a profissionais que tenham conhecimento na linguagem PHP, impedindo sua implementação por web designers que não sabem programar em PHP.

Para Nigri, o Gutenberg é um sistema pronto, porém novas funcionalidades podem ser acrescentadas conforme as necessidades específicas de cada projeto. E confirma que quando um sistema é desenvolvido para um determinado cliente e é reutilizado para outro, as adaptações podem ocasionar problemas de usabilidade. Porém, mesmo que o sistema seja desenvolvido especificamente para um cliente, poderá apresentar problemas.

Os dados a seguir, foram retirados do manual do Gutenberg, fornecido junto com a 2ª versão adquirida pela Faperj, em 2003. Nesta ocasião, o *site* foi todo reformulado e houve o lançamento de um novo produto, o Boletim *online* semanal da Faperj, necessitando ainda mais da utilização da tecnologia de publicação *online*.

Os componentes do Gutenberg são:

- 1) **Administrador:** é o usuário responsável pela administração do *site*. Nele se pode cadastrar usuários, publicar textos, ver relatórios e efetuar todas as tarefas diárias associada ao *sites* publicado com o Gutenberg.
- 2) **API de acesso (*middleware*):** Todo acesso ao conteúdo armazenado no Gutenberg é feito através de chamadas as suas APIs (*Application Program Interface*), comumente chamado de *middleware*, por e colocar entre o *site* e o banco de dados.

- 3) **Site de demonstração:** O principal *site* de demonstração do Gutenberg é o EXEMPLO, que vem junto com a instalação (inclusive já vem com dados de demonstração).
- 4) **Banco de dados:** Todas as informações guardadas no Gutenberg, com exceção dos arquivos de imagens e mídia ficam armazenadas em um banco de dados relacional. Este banco tem um papel fundamental no funcionamento do sistema.

O sistema Gutenberg baseia-se nos seguintes conceitos para definir e classificar todas as informações inseridas. Estes conceitos serão citados no decorrer da pesquisa:

- 1) **Contextos:** representa a localização do usuário no *site* em relação a todos os assuntos disponíveis. Os contextos são hierarquizados e sintetizam toda a gama de assuntos/conhecimentos que são disponibilizados.
- 2) **Objetos de informação:** representa qualquer informação que é inserida no *site* de forma dinâmica: artigos, resumos, eventos (da agenda de eventos) e imagens são exemplos de objetos de informação, e também as ferramentas interativas: fóruns e enquetes.
- 3) **Termos de associação:** são definições que agrupam e classificam os objetos de informação, que tratam do mesmo tema. Cada objeto pode ser classificado em quantos termos for desejado, independente do seu contexto. Com isto, cria-se uma relação ampla entre todos os objetos de informação do *site*.

Tipos de objeto de informação:

Alguns atributos são comuns a todos os objetos de informação: nome, data da publicação, contexto base, lista de termo de associação, tipo de destaque. Além destes, cada objeto tem atributos relacionados pertinentes apenas aos objetos de mesmo tipo: texto, artigo, notícia etc. Segue todos os tipos de objetos suportados no sistema de redação:

- 1) **Conteúdo editorial:** pode ser desde uma notícia até uma foto comentada e compõe a maior parte das informações que serão disponibilizadas no *site*. Seus atributos comuns são: referência, título, título interno, chamada e formato editorial (se é uma notícia, um artigo, um resumo comentado ou uma foto etc). Pode conter listas de *links* interessantes sobre o assunto ou até ser aberto a comentários dos leitores. Todas as informações sobre o conteúdo colocadas no banco de dados através dos *templates* de HTML.
- 2) **Eventos (da agenda de eventos):** contém os atributos pertinentes a um evento: local (nome, endereço e telefone), data de início, data de término, organizador (nome, endereço, telefone, e-mail e URL), preço, e-mail e URL.
- 3) **Fórum:** é um objeto linear (em linhas de discussão) e exige que o usuário se identifique antes de poder ler ou publicar uma mensagem. Contém os seguintes atributos: remetente, título, data de publicação e corpo da mensagem.
- 4) **Enquetes:** são pequenas pesquisas compostas de uma única pergunta com um número pequeno de respostas objetivas. Possuem os seguintes atributos: data de entrada no *site*, data de saída do *site*, respostas e resultado.
- 5) **Testes (Quizzes):** utilizados para e publicar prova e teste no *site*, através da implementação de um sistema de *e-learning* ou de um complemento interativo ao *site*.

O conteúdo editorial será o objeto testado por ser o único utilizado no *site* da FAPERJ. Portanto, os demais objetos não farão parte da pesquisa.

5.3

Motivação da utilização do Sistema de Gerenciamento de Conteúdo (CMS – Content Management System) no *site* da Faperj.

Em 2002, a Faperj decidiu reestruturar o seu *site* para que ele fosse, além de um *site* institucional, uma mídia de referência em notícias de Ciência e Tecnologia do Estado do Rio de Janeiro. Para atender a esse objetivo era necessário criar forma de comunicação digital dinâmica que facilitasse e agilizasse a publicação de notícias, boletins, *releases*, *clippings*, que pudessem ser gerenciadas de maneira fácil pelo setor de Comunicação. Um sistema que

funcionasse como uma redação virtual, onde todo o trabalho pudesse ser realizado através da Internet.

A nova coordenação do setor Núcleo de Difusão Científica, responsável pela Comunicação da Faperj, resolveu por utilizar um Sistema de Gerenciamento de Conteúdo no projeto do novo *site*. Pois, conforme visto no capítulo anterior o CMS é indicado para *sites* que precisam administrar e publicar grande quantidade de conteúdos, mesmo que os colaboradores estejam dispersos geograficamente. Além de oferecer aos jornalistas envolvidos no projeto a facilidade de realizar as suas tarefas via Internet sem precisar ter conhecimentos em HTML, ou seja, independente da interferência do web designer toda vez que precisasse publicar ou alterar um conteúdo do *site*.

5.4

Aquisição do CMS

O próximo passo era saber se deveria montar ou comprar um CMS. FRIEDLÉIN (2003) apresenta uma lista de motivos para comprar ou montar o seu próprio CMS:

Motivos para não montar o seu próprio CMS:

- Os CMSs não são sistemas simples, de modo que seria necessária uma equipe de desenvolvimento muito forte que precisaria não só ter conhecimento profundo de banco de dados e linguagem de programação, mas que também precisaria entender a importância das interfaces com o usuário e processos editoriais, para poder criar um sistema de publicação eficaz;
- Será necessário continuar a desenvolver o *software*, atualizar versões, à medida que novas necessidades vão surgindo;
- Terá que dar suporte à solução quando surgirem problemas;
- Mesmo que possa priorizar os elementos que mais precisa, gastaria muito tempo para desenvolver seu próprio sistema. Quando se compra um CMS pronto, já vem com mais das funções que precisa prontas, reduzindo o requisito de tempo necessário para que o sistema esteja apto a funcionar;

- O mercado está evoluindo rapidamente e depois de desenvolver sua própria solução, poderia descobrir que já existe outro CMS que tem todas as ferramentas que precisa por um custo menor.

Motivos para não comprar um CMS:

- Existem soluções intermediárias no mercado a orçamentos menores, mas os CMSs costumam ser muito dispendiosos de comprar, implementar e manter. Dependendo do custo, pode-se desenvolver o seu próprio CMS por um custo menor, fazendo exatamente o que precisa;
- Virão com recursos que não precisa. Pagará não apenas pelos recursos que não usa, mas também para personalizar os que comprou. Precisaré pagar pelo suporte e nas emergências precisará esperar que o fornecedor resolva o problema. Pelo menos, se montar o seu próprio CMS saberá como consertá-lo se houver algum problema;
- Como o mercado está evoluindo rapidamente, pode ocorrer da empresa que desenvolveu o seu sistema saia do mercado, ou também, com o desenvolvimento de novas tecnologias, sua compra pode se tornar obsoleta.

Para Nigri,

“é mais barato comprar um sistema de publicação pronto do que desenvolvê-lo, apesar de quando você compra um sistema pronto, nunca é do jeito que você queria. Porém, mesmo que você o desenvolva, pode ficar mais próximo, mas nunca ficará exatamente o que você queria”.

Como a equipe de desenvolvimento estava ocupada com outros sistemas e o tempo previsto para a implementação do projeto do novo *site* era curto, optou-se pela compra de um sistema pronto. A próxima etapa era escolher o CMS que se adequasse melhor as nossas condições e projeto.

5.5

Seleção do CMS

Segundo FRIEDLÉIN (2003), o processo de seleção de um CMS possui 8 etapas:

1. Defina requisitos: é a definição do que precisa, listando critérios de seleção para atender aos seus requisitos. Além de definir quem fará parte do processo de seleção, da avaliação e quem tomará a decisão de compra;
2. Examine o mercado: procurar no mercado quais os fornecedores que provavelmente se ajustarão às suas necessidades, que poderá ser através de pesquisas em *sites* dedicados ao gerenciamento de conteúdo, relatórios de analistas, perguntar a alguns amigos ou a especialistas da área;
3. Lista curta de fornecedores: reduzindo a lista de fornecedores descartando os fornecedores inadequados, procurando opiniões sobre seus produtos, fazendo reuniões para conhecer melhor a empresa e o produto;
4. Envie a solicitação de proposta: é um documento mais formal e rigoroso, que deve ser preenchido pelo fornecedor. Na solicitação de proposta deverá ser enviado um resumo dos seus objetivos e uma lista detalhada de perguntas em relação ao desempenho, autoria de conteúdo, controle de versão, implementação, técnica e integração com outros sistemas, escalabilidade e recuperação de desastre, armazenamento, gerenciamento e publicação do conteúdo, *templates*, personalização, fluxo de trabalho, gerenciamento de segurança e direitos, relatório de informações de gerenciamento, treinamento e suporte;
5. Reuniões e apresentações de fornecedores: nesta etapa deverá tirar as dúvidas pessoalmente com os fornecedores sobre as respostas obtidas através da solicitação de proposta, detalhando os pontos fortes e fracos de cada sistema;
6. Lista curta final: estreitar a lista para dois ou três finalistas com a participação da equipe de projeto e dos membros operacionais, equipe técnica e editorial, que deverão avaliar o sistema através de uma demonstração de funcionalidade;

7. Negociações comerciais e devidos cuidados: no lado comercial, combinar preço, prazos de pagamento e acordos de serviço e suporte. Os devidos cuidados envolvem o acompanhamento de referências, conversa com cliente atuais e verificar as circunstâncias comerciais do fornecedor (status financeiro, propriedade, compromissos, histórico etc);
8. Decisão: se a decisão da escolha do fornecedor não for clara, pode ser devido a problemas internos da sua empresa ou porque não encontrou o produto adequado. Neste caso, reconsidere a montagem do próprio sistema.

A escolha do Gutenberg, entretanto, foi através do processo de indicação: um consultor contratado indicou o sistema por conhecê-lo anteriormente e por ter um custo razoável no mercado.

Não houve concorrência com outros produtos. As negociações foram feitas em reuniões entre a coordenação do Núcleo de Difusão Científica da Faperj, a consultoria e a Winco, empresa desenvolvedora do sistema.

Em maio de 2002, a Winco apresentou o Gutenberg para a coordenação do setor e a equipe operacional, quando foram demonstradas suas funcionalidades através do Demonews, *site* de demonstração do *software*. Foram respondidas questões sobre as licenças e implementação ao *front-end* (parte do programa que é responsável pela interface do usuário). Nesta apresentação, também foi colocado pela empresa que desenvolveu o sistema, que pelo fato do Gutenberg ser um produto novo, requeria um aperfeiçoamento constantemente.

Foi disponibilizado à equipe operacional um *login* para a utilização do Gutenberg através do *site* de demonstração, para que fosse testado.

5.6

Implementação do sistema

Os representantes da Winco explicaram que o Gutenberg não oferecia um *template* padrão (modelos de página com marcações dos lugares onde a informação dinâmica deve ser apresentada), diferente de outros sistemas de publicação de conteúdo, utiliza a linguagem PHP como base para todas as marcações feitas nos modelos. Mas estas características têm vantagens e

desvantagens, principalmente para programadores sem experiência em programação de *sites* dinâmicos (Manual do Gutenberg, 2004):

Vantagens:

- Não é necessário utilizar uma outra linguagem para escrever os *templates*;
- Todas as funções do PHP ficam disponíveis para utilização, viabilizando assim qualquer layout/navegação que seja imaginado pelo web designer;
- Outras informações, vindas de bases de dados que são externas ao Gutenberg podem ser exibidas na mesma página HTML, possibilitando uma integração total com qualquer sistema.

Desvantagens:

- Flexibilidade implica em planejamento. Cada *site* deve ser planejado antes de se iniciar a execução para que os resultados sejam o desejados. Afinal, os exemplos fornecidos com o sistema funcionam, mas são apenas sugestões de implementação;

Portanto, a equipe de produção (designers e jornalistas) estava livre para criar a interface e a arquitetura da navegação, mas precisava haver um planejamento prévio do *site*, ou seja, planejar a sua estrutura e separar os tipos de informações que seriam apresentadas para que se pudesse criar a programação que fizesse a integração do sistema aos *templates*.

De acordo com e-mail enviado pela empresa Winco aos integrantes do processo de compras do Gutenberg, a implementação podia ser realizada pela empresa que desenvolveu o sistema ou pelo próprio cliente e estava composta em 3 etapas:

- A. Projeto do *site* (arquitetura da informação): quando era definida a quantidade e a localização dos elementos do Gutenberg que seriam definidos no *front-end*.
- B. Criação de arquivos PHP (páginas.php): quando eram criadas as páginas .php que seriam associadas aos *templates*. Nestas páginas eram buscadas

as funções necessárias para se aplicar ao *front-end* os elementos do Gutenberg que seriam utilizados.

- C. Integração dos arquivos HTML com arquivos PHP: quando traziam para os *templates* as chamadas dos elementos que seriam exibidos nas páginas phtml (páginas programadas nas linguagens PHP + HTML).

A Faperj optou por realizar a implementação com recursos próprios, utilizando a documentação fornecida (API) e o suporte técnico da Winco. Mas mesmo assim, via-se a dificuldade encontrada pela fundação para concluir a implementação. Portanto, optou-se para que a execução da etapa de criação de arquivos PHP (etapa B) fosse realizada pela equipe da Winco, tornando possível à equipe da FAPERJ a execução da etapa de integração do HTML ao PHP (etapa C), ou seja, a confecção dos *templates*. Porém, esta integração restringe-se a profissionais que tenham conhecimento na linguagem PHP e conforme declaração do Nigri apresentada anteriormente, a primeira versão tinha um código difícil impossibilitando a implementação do sistema. Por este motivo, foi decidido que a Winco ficaria responsável pelo processo das 3 etapas de implementação.

Foram necessárias várias reuniões entre a equipe e os coordenadores do setor para analisar o conteúdo existente, as informações que seriam adicionadas e como ficariam estruturadas no novo *site* (etapa A). Depois de pronto, o projeto foi entregue a Winco, junto com as páginas em HTML que serviriam de base para os *templates* (etapa C).

Finalizados os *templates*, o sistema foi implantado em um servidor de web paralelo, pelas equipes da Winco e da Faperj, para que as informações do *site* em vigência fossem transferidas para o banco de dados conectado ao Gutenberg. A pedido da Faperj, a Winco realizou um treinamento dos usuários da instituição (designers, jornalistas e programador) e confeccionou um manual que pudesse orientá-los no uso e na possível atualização do *site*.

Os testes foram realizados e os usuários encontraram alguns problemas de usabilidade no sistema durante esta fase. Alguns ajustes foram feitos e o *site* foi lançado.

Em 2004, a Faperj decidiu lançar um novo veículo *online*: o Boletim Eletrônico Semanal da Faperj. Era um projeto considerado ousado por utilizar em uma mídia *online* um design similar às mídias impressas. Ou seja, a capa do

boletim é diagramada de forma que as fotos, títulos, chamadas e *leads* ficassem colonados conforme os jornais impressos, diferente dos demais boletins eletrônicos utilizados na época. Tornando assim, a leitura da capa do boletim mais atrativa e menos cansativa para os leitores.

Como o Gutenberg é uma ferramenta flexível, que pode ser adaptada a qualquer design, optou-se em adquirir a nova versão e utilizá-lo no novo projeto. Apesar dos desenvolvedores terem realizado uma série de modificações do sistema para tornar a sua implementação mais fácil, foi decidido que a Winco realizaria novamente as etapas B (criação de arquivos PHP) e C (integração dos arquivos HTML com arquivos PHP) do processo.

A implementação foi realizada com sucesso, foram entregues os *templates* da home do *site*, páginas internas e 8 modelos diferentes da home do boletim, juntamente com um novo manual do sistema para designers e programadores e o manual do usuário, onde continha informações para a publicação tanto no novo *site*, quanto no boletim. A facilidade oferecida pelos *templates* (.phtml) possibilitou ao web designer a criação de outros 30 novos *templates* da home do boletim, adaptados às necessidades surgidas no decorrer de sua aplicação.

Durante o período de testes, alguns problemas de usabilidade encontrados pelos usuários da Faperj, foram analisados e ajustados pela empresa que desenvolveu o sistema.

5.7

Conclusão parcial

A maioria dos *bugs* (erros de programação que causam defeito no programa) encontrados durante a utilização das duas versões do Gutenberg adquiridas pela Faperj foram resolvidos devido a parceria entre os profissionais da Winco e os usuários da Faperj.

Alguns problemas de usabilidade não foram resolvidos. Muitos deles foram causados pela reutilização do sistema em novo contexto, comprovando a hipótese levantada pelo pesquisador, pois estava sendo utilizado um sistema pronto, desenvolvido para outro cliente e adaptado para as novas necessidades, portanto, sem planejamento prévio do crescimento. Outros problemas ocorreram

devido às adaptações ao novo contexto, que foram solicitadas pelos próprios usuários da Faperj.

Apesar do Gutenberg ser um sistema que não tem um *template* pré-definido e oferecer o processo de desenvolver *templates* exclusivos para cada cliente que o adquira, muitas de suas características ainda estão vinculadas ao cliente para o qual foi desenvolvido inicialmente. Ou seja, não tem uma interface universal que possa atender a qualquer tipo de usuário sem causar frustrações.

Mas, de qualquer forma, utilizar um sistema de gerenciamento de conteúdo no novo *site*, facilitou muito a realização das tarefas dos usuários do setor de Comunicação da Faperj. A coleta, administração e publicação do conteúdo estão sendo feitas pelos jornalistas de forma mais dinâmica e independente de intervenção do web designer.

5.8

Referências bibliográficas

FRIEDLEIN, Ashley. **Como gerenciar sites Web de sucesso**. Tradução Daniel Vieira. Rio de Janeiro: Campus, 2003.

Manuais do Gutenberg, 2003 e 2004.

NIGRI, Ariel – diretor da Winco. Entrevista realizada em março de 2005.

Site da WINCO – Tecnologia e Sistemas. <http://www.winco.com.br>. Acesso em 15/07/04.