

# A importância da interoperabilidade.

Há algum tempo tenho tido muito contato com PHP como desenvolvedor e também como usuário de sistemas prontos. Porém depois de um tempo me deparei com um problema sério: nenhum desses sistemas é suficientemente interoperacional, e muitas vezes eu mesmo tenho deixado de produzir sistemas interoperabilizados. E é sobre isso que quero tratar nesse artigo, a necessidade de se produzir sistemas interoperabilizados.

Antes de continuar gostaria de colocar da forma mais clara que eu puder, o que é interoperabilidade.

Interoperabilidade é a capacidade de um sistema(informatizado ou não) de se comunicar de forma transparente(ou o mais próximo disso) com outro sistema (semelhante ou não). Para um sistema ser considerado interoperável é muito importante que ele trabalhe com padrões abertos. Seja um sistema de portal, seja um sistema educacional ou ainda um sistema de e-commerce, hoje em dia se caminha cada vez mais para a criação de padrões para sistemas.

Por mais que possamos pensar num "sistema revolucionário" toda vez que vamos desenvolver um novo sistema, é bom ter sempre em mente que alguém já passou por isso um dia, e essa pessoa também teve a idéia de fazer um "sistema revolucionário". O que fazer então? Você tem duas opções: optar por um sistema pronto (na maioria das vezes sem atender suas necessidades) ou desenvolver um próprio.

Se a sua opção for a primeira, você deve ter em mente que, por mais simples que seja o uso desse sistema em um dado momento, um dia ele poderá precisar se comunicar com outro sistema. Portanto, é bom que se escolha um sistema que trabalhe com padrões abertos, pois assim será mais fácil no momento em que você precisar interligar seus sistemas.

No entanto se a sua opção foi a segunda(desenvolver um próprio sistema), é hora de pensar no tipo do sistema que você deseja criar. Seja qual for a finalidade do seu sistema, deve haver um padrão aberto que define como implementar isso que você deseja criar. Por exemplo, digamos que você vá criar um sistema de notícias para o seu site. Ao criar esse sistema, você talvez pense que ele só será usado no seu site, mas digamos que no futuro você deseja liberá-lo com uma licença qualquer para que outras pessoas o usem. Pois bem, essas pessoas que um dia vão usar o seu sistema, talvez já tenham outros sistemas que elas desejem que se integrem com o que você desenvolveu e aí nós temos um problema: o seu sistema de notícias "conversa" em algum padrão aberto? Se não, provavelmente ele estará fadado ao fracasso, ou, na melhor das hipóteses, ele será usado por pessoas que não estão interessadas em contribuir para o desenvolvimento, provavelmente usando-o de forma muito restrita. Para um desenvolvedor que tem uma visão de Software Livre, ou que gosta de saber que seu software está sendo de alguma utilidade, isso é, no mínimo, frustrante.

Eu digo isso por experiência própria, pois como todo iniciante um dia já produzi software sem me preocupar com isso.

Alguns projetos começaram do nada, sem nenhum padrão e hoje caminham para uma padronização, porém de forma muito dolorosa e embaraçosa porque um dia não pensaram que isso seria importante, ou porque a emergência não deixou que fosse implementado. É o caso do PHPNuke, por exemplo. Apesar de, atualmente, apresentar alguma pouca padronização (por exemplo no que se refere a exportação de notícias em formato RSS), ainda tem muito a aprender.

Atualmente o esforço de grandes empresas e organizações como, por exemplo, a W3C para criar ou estabelecer padrões de comunicação e formatação para Web não tem sido em vão, e cada vez mais os desenvolvedores e empresas estão empregando esses padrões nos seus produtos. A seguir listo alguns exemplos de padrões para web (ou não ) que foram criado para auxiliar a comunicação entre softwares:

- Para conversa entre plataformas/tecnologias o RPC, hoje em dia implementado com XML, gerando o XML-RPC;

- Para exportação notícias temos alguns padrões: RSS/RDF/ATOM, todos implementados com XML;
- Para manipulação de blogs faz-se uso da Blogger-API, que usa XML-RPC;
- Para a padronização de cursos usados em sistemas de educação à distância há o SCORM e outros;
- Para formatação de conteúdos para web (páginas) o CSS associado com o XHTML (evolução do HTML);
- E para muitas outras coisas o uso do XML tem se tornado indispensável.

Esses são alguns poucos exemplos que tenho tido contato, com certeza há muitos outros, e alguns bem específicos. Se você vai desenvolver recomendo que você consulte alguma organização de referência para padronizações de software.

Entre os usuários de PHP eu tenho notado, no entanto, que poucas pessoas/grupos/empresas têm se preocupado com isso. Na minha humilde opinião alguns fatores determinantes para que isso ocorra são:

- Essência do PHP. Pelo fato do PHP ser uma linguagem e não um framework (por exemplo), ela não trata de assuntos específicos, e abrange muitas áreas, deixando os pontos específicos para o desenvolvedor. O que em si não é um defeito, pelo contrário. E como você já deve estar pensando, isso não é exclusividade do PHP, mas de qualquer linguagem. Mas em todo caso dificulta a aplicação de padrões quando deixa isso por conta do desenvolvedor.
- Inexperiência. Devido ao fato do PHP ser uma linguagem relativamente simples de se aprender e implementar algo com ela, muitas pessoas sem conhecimento acabam produzindo softwares sem nenhum estudo, o que acaba dando em um sistema único e não reutilizável (ou no nosso caso, sem interoperabilidade).
- Falta de uma comunidade séria. Apesar de haver uma grande quantidade de pessoas envolvidas com o PHP, não há uma comunidade séria que trabalha para tentar desenvolver soluções e suporte. O mais próximo disso é o site oficial, mas que passa longe de ser um centralizador de uma comunidade. Recentemente Chris Shiflett criou a PHP Community (<http://www.phpcommunity.org>), o que parece ser a primeira iniciativa séria da "profissionalização" do uso do PHP. Como voluntário lá, espero que consigamos relamente ajudar a comunidade.

Esses pontos fracos que eu apresentei são apenas alguns poucos, com certeza há outros fatores para que não haja um uso tão grande de padrões, um deles é o mal que assola muitos bons desenvolvedores: a urgência de um projeto. Esse também tem sido um enorme problema, pois criou-se a idéia de que o PHP é uma linguagem de alta produtividade, o que não deixa de ser verdade; porém se esquece muitas vezes que essa produtividade está associada a prazo e bons profissionais. Logo temos um problema que não é fácil de resolver se não fazendo código de forma desorganizada ou sem que pensem muito no nosso problema. E então temos o que eu costumo "carinhosamente" chamar de toscódigo (código muito tosco).

Bom, para resolver esses problemas e termos um sistema minimamente interoperacional devemos ter algumas coisas em mente:

- Se você não é o desenvolvedor, você deve contratar um bom profissional, e isso não é tão difícil de se encontrar no mercado hoje em dia;
- Como desenvolvedor, primeiramente você deve ter um bom conhecimento do negócio do seu cliente, e também pensar numa futura integração de sistemas;
- Ainda como desenvolvedor, é bom conhecer um pouco de padrões abertos, e se o sistema do seu cliente puder fazer uso de algum, corra atrás e leia o máximo que puder sobre a aquele padrão;

Atualmente estamos com um volume de programadores em PHP muito grande, porém, como em todo processo evolutivo, somente os melhores sobreviverão nesse mundo (se quiserem permanecer como programador). Digo isso não apenas com relação ao PHP, mas se você é programador e quer sobreviver às custas do seu trabalho, é bom começar a se tornar um profissional cada vez melhor. Um diferencial será, com certeza, a capacidade de entender e trabalhar com padrões abertos pois como eu já disse, o mercado está correndo atrás de tornar as coisas interoperacionais, pois isso diminui em muito o custo de

produção de software.

Falei que há muitos projetos que engatinham ainda em termos de padronização, porém alguns já nasceram com isso em mente, e outros logo se adaptaram. Em se tratando de PHP temos um fantástico sistema de portais, estilo PHPNuke, chamado Drupal (<http://www.drupal.org>), que começou como blog e hoje é um completo CMS ( Content Management System ). Esse é um projeto que aprendeu logo o valor de se trabalhar com padrões (no início teve sérios problemas) , e hoje é um ótimo projeto apesar de ainda ficar devendo em alguns pontos(o sistema de templates ainda é meio complicado ).

Em se tratando de outras tecnologias há o OpenACS, por exemplo. Uma tecnologia muito forte e que está se consolidando cada vez mais principalmente fora do Brasil e que é um dos melhores exemplos de uso de padrões e tecnologias de ponta da atualidade.

Esses são exemplos de projetos que trabalham com padrões e que têm ganhado um grande reconhecimento na comunidade de desenvolvedores. Com certeza há muitos mais projetos que caminham no caminho certo, porém ainda há muito trabalho a se fazer e se você quer fazer parte dessa (r)evolução, recomendo fortemente que corra atrás de entender o mínimo possível de padrões abertos, pois fatalmente um dia você irá se deparar com algum cliente que deseja algo mais e que talvez você não consiga oferecer.

Alguns sites que recomendo que você consulte na hora de pensar num sistema para web:

<http://www.w3c.org> - maior organização para tratamento de padronização da web;

<http://www.alistapart.com> - um excelente site sobre acessibilidade e usabilidade

<http://www.tableless.com.br> - site brasileiro com a mesma ideia do A list apart

<http://www.rfc-editor.org> - um bom começo para quem precisa procurar um RFC

<http://www.xmlrpc.com> - site que trata do XML-RPC

<http://xmlrpc.free-conversant.com/docs/bloggerAPI> e <http://www.dentedreality.com.au/bloggerapi> - dois sites que tratam da blogger-api, o último fala de uma implementação em PHP.

<http://www.google.com> - não sabe por onde começar? Procure por aqui. Saber já é um bom começo para quem quer aprender. :-)

Sinceramente, desejo que a comunidade PHP brasileira (e mundial), se envolva com esse assunto que atualmente é de grande importância para o desenvolvimento da web. A interoperabilidade é fundamental para que um sistema ganhe visibilidade se desenvolva, pense nisso na próxima vez que você for desenvolver algum software. Vamos nos esforçar cada vez mais para produzir software de qualidade, pois se somos uma comunidade, cabe a nós a tarefa de nos ajudarmos, e uma grande ajuda é facilitar o nosso próprio trabalho.

Autor: Rafael Ferreira Silva - <http://www.webphp.com.br>

Originalmente escrito para o site PHPBrasil.com - <http://www.phpbrasil.com>