Universidade Norte do Paraná

SISTEMA DE ENSINO PRESENCIAL CONECTADO
TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

MARCELO EDUARDO ALVES PAIXÃO RESENDE

SISTEMA DE CONTROLE DE ENTREGA DE PIZZA

Uberlândia
2015

MARCELO EDUARDO ALVES PAIXÃO RESENDE

SISTEMA DE CONTROLE DE ENTREGA DE PIZZA

Trabalho Individual apresentado à Universidade Norte do Paraná - UNOPAR, como requisito parcial para a obtenção de média semestral nas seguintes disciplinas: Introdução ao Desenvolvimento de Sistemas Web; Processo de Negócio e Software; Lógica e Técnicas de Programação e Metodologia Científica.

Orientadores: Profª Merris Mozer
Prof. Marco Hisatomi
Prof. Anderson Macedo
Prof. Claudiney José de Souza

Uberlândia
2015

SUMÁRIO

1 INTRODUÇÃO

O uso de tecnologias web cresce num ritmo muito rápido. Em intervalos de meses, novas tecnologias surgem, algumas que despertavam interesse se tornam obsoletas, uma infinidade de tecnologias em uso e até outras que caíram em desuso recebem atualizações e são revitalizadas.

Neste cenário, profissionais que antes eram bem procurados pelo mercado também podem se tornar obsoletos caso não se atualizem no mesmo ritmo das mudanças do mercado.

O profissional não necessita saber apenas o conteúdo técnico para programação, precisando também se inteirar de como as organizações trabalham, quais são seus objetivos e como pode ajudar na missão da empresa.

Este trabalho se propõe a explicar como tecnologias atuais de desenvolvimento para a web surgiram, como evoluíram, como são utilizadas atualmente e quais seus aspectos técnicos. Para melhor entendimento de como as empresas operam, é sugerido o uso das técnicas fornecidas pelo *Enterprise Knowledge Development*.

2 OBJETIVO

O objetivo deste estudo, de forma geral, é entender como tecnologias web podem auxiliar organizações e quais são seus aspectos técnicos.

Para a linguagem HTML, será utilizada uma abordagem simples, objetiva e capaz de apresentar bases sólidas para compreensão do que é esta tecnologia e porque é tão importante para a web, incluindo trechos de código bem explicados e funcionais.

Um caso de estudo será analisado para melhor compreensão das alternativas existentes para criar HTML dinâmico, que depende de outras tecnologias para tal. As tecnologias a serem comparadas serão PHP e ASP.NET, fornecendo dados sobre suas histórias, dados de utilização e qual a melhor opção para o cenário apresentado.

Uma visão geral sobre o método Enterprise Knowledge Development será apresentada, exemplificando seus principais componentes e como se relacionam.

3 HTML

HyperText Markup Language (Linguagem de Marcação de Hipertexto), abreviado como HTML, é a linguagem de marcação padrão para construção de páginas para *web*. Foi criada por Tim Berners-Lee para auxiliar na sua comunicação com colegas de pesquisa e sem regras rígidas de uso.

A *World Wide Web*, desde seu surgimento, evoluiu com o objetivo de fornecer conteúdo cada vez mais rico. Inicialmente fornecendo meios de exibição de textos formatados e imagens, durante um processo de evolução contínuo foram o HTML ganhou suporte para inserção de vídeos, áudio, animações e muitas outros recursos que permitiram à *web* fornecer conteúdo rico.

Os documentos HTML, em sua essência, são documentos de texto puro, com o conteúdo que o criador ou interessado na sua publicação deseja exibir envolto em *tags*, comandos da linguagem que dão instrução aos navegadores do que é o conteúdo, como exibi-lo e, em alguns casos, como é o seu visual.

3.1 EVOLUÇÃO DA LINGUAGEM

Até 1993, a linguagem não tinha uma especificação formal. Em 1995, foi criado seu primeiro padrão oficial, o HTML 2.0. Em 1997 duas versões foram lançadas, o HTML 3.2 no início do ano e HTML 4.0 no fim. Com o avanço rápido da web, algumas mudanças foram introduzidas e, em 1999, o HTML 4.01 foi lançado, que esperava-se que fosse a última especificação da linguagem HTML.

Durante a década de 90, a linguagem sofreu com programadores que não aderiam às suas especificações e com as empresas que desenvolviam navegadores. Estas empresas, assim como os programadores, não seguiam todas as especificações da linguagens, ou as implementavam de forma inconsistente e até inseriam funcionalidades proprietárias que só funcionavam em seus navegadores.

Por volta da virada do século, uma reformulação chamada de XHTML começou a ser utilizada. Esta era inspirada no XML, outra linguagem de marcação para fins especiais com regras mais rígidas de sintaxe. A partir daí, web designer e desenvolvedores começaram a pressionar seus colegas para que desenvolvessem de acordo com as especificações, e ao mesmo tempo, pediam para

que as empresas que desenvolviam navegadores também aderissem melhor à tendência.

Por volta de 2008 a primeira especificação do HTML5 foi publicada. Esta nova tecnologia visa eliminar ou diminuir a necessidade do uso de *plug-ins* para exibição de conteúdo, melhor semântica e retro compatibilidade com especificações anteriores.

3.2 COMANDOS HTML

Para utilizar as funcionalidades da linguagem HTML é necessário o uso de *tags*, que são os comandos da linguagem. A função das tags é separar conteúdos, delimitar onde começam e onde terminam, especificar seu tipo e até seu comportamento, como ligações para outras páginas. A função do navegador é interpretar estes comandos e exibir o conteúdo de acordo com o que foi especificado pela *tag*.

Uma *tags*, em sua utilização prática, é composta por um sinal de menor que (<), que faz sua abertura, seguido pelo nome do elemento a ser utilizado e um sinal de maior que (>), que faz seu fechamento. O nome da tag pode ser escrito em letras maiúsculas, minúsculas ou em uma combinação entre elas, como `<tag>`, `<TaG>` ou `<TAG>`, porém em algumas especificações, como o XHTML, e em recomendações de boas práticas de programação, o uso exclusivo de letras minúsculas é obrigatório.

A maioria das tags funcionam em par, que consiste em uma tag de abertura e uma tag de fechamento, que inclui uma barra (/) logo após o sinal de menor que. Por exemplo, a tag `<b>` é responsável por exibir o texto inserido em negrito:

```
<b>texto em negrito</b>
```

Algumas *tags* são *tags vazias*, e por isso não obrigam o uso em par com uma *tag* de fechamento. Entretanto, de acordo com a especificação utilizada, pode ser obrigatório o uso de uma *tag* de fechamento, como no XHTML. Um par de *tags* com seu conteúdo formam um *elemento*.

Como na maioria das linguagens, um documento HTML também pode ter comentários, que é iniciado por `<!--` e fechado por `-->`. O conteúdo dentro de um comentário não é interpretado pelos navegadores, e servem para

adicionar anotações conforme a necessidade do programador. Um documento básico em HTML5 pode ser visto no exemplo abaixo:

```
<!DOCTYPE html>
<html lang=”pt-BR”>
    <head>
          <title><!-- título da página --></title>
    </head>
    <body>
    <!-- conteúdo da página -->
    </body>
</html>
```

3.2.1 Atributos

A tag de abertura de um elemento pode incluir um ou mais atributos, a fim de modificar sua aparência ou comportamento. Um atributo consiste em um nome de atributo e pode ter um valor, podendo ser demonstrado da seguinte forma:

```
<input type=”checkbox” checked />
```

A *tag input* do exemplo acima possui um atributo *type* com o valor *checkbox*, que define o tipo do elemento de um formulário como uma caixa de seleção, e também possui o atributo *checked*, que neste caso não possui nenhum valor. Em XHTML, todos atributos devem ter um valor, que neste caso seria `checked=”checked”`, mas em outras especificações o exemplo é válido.

3.3 APARÊNCIA DO DOCUMENTO HTML

A linguagem HTML é utilizada para criação de páginas com conteúdo rico, chamado de hipertexto. Para proporcionar uma melhor experiência final, uma página pode ser incrementada com a inserção de conteúdo além de texto, como imagens, vídeos e áudio. Também é possível alterar a aparência de textos, agrupar conteúdo em tabelas ou seções e criar ligações, chamadas de links, para outras páginas do mesmo site ou ligações externas para outros sites das web.

3.3.1 Inserindo imagens

Imagens são parte fundamental de muitas formas de comunicação, e naturalmente, são suportadas pela linguagem HTML desde sua primeira versão. As imagens são inseridas em documentos HTML através da *tag img*. Os formatos mais utilizados são JPEG, PNG, GIF e recentemente, o formato SVG tem recebido ampla adoção, embora os navegadores consigam exibir vários outros formatos de imagem, variando conforme a decisão do desenvolvedor dos mesmos.

Uma *tag img*, com os principais atributos definidos, tem a seguinte estrutura:

```
<img src=”pasta/subpasta/praia.jpeg” alt=”Foto da praia”>
```

O atributo *src* define o local que a imagem está hospedada, podendo ser no mesmo servidor do documento HTML ou em um servidor externo, exigindo assim o endereço completo. O atributo *alt* define um título para a imagem, que só será exibido caso o usuário pare o mouse por alguns instantes sobre a imagem, ou será exibido no lugar da imagem caso o navegador encontre dificuldades para obter a imagem do local definido.

Vale lembrar que as imagens não estão tecnicamente inseridas no documento HTML, sendo apenas referenciadas no atributo *src*. É trabalho do navegador acessar o local definido, obter os dados da imagem e exibi-la conforme as definições da *tag*.

Atributos adicionais podem ser fornecidos pelo programador para alterar a aparência e comportamento do elemento.

Um atributo *align* pode receber os valores *top, right, bottom, left* e *middle*, alterando sua posição em relação aos outros elementos e ao espaço que ocupa.

Para mudar as dimensões da “caixa” do elemento que a imagem está inserida, é possível fornecer os atributos *width* e *height*, que alteram, respectivamente, a largura e altura da imagem.

O atributo *border* altera a aparência da “caixa” do elemento, adicionando uma borda externa com a largura especificada.

Os atributos *width*, *height* e *border* recebem um número inteiro como valor, que é interpretado como a quantidade de *pixels*.

3.3.2 Inserindo vídeos

Com o rápido avanço nas conexões de internet e consequente aumento de velocidade de transferência de dados, surgiu a oportunidade de utilizar vídeos em páginas da web. Utilizando um navegador moderno, compatível com a especificação HTML5, é possível inserir vídeos sem a necessidade de programas externos utilizando a *tag video*, exemplificada a seguir:

```
<video width="320" height="240" controls>
  <source src="video.mp4" type="video/mp4">
  <source src="video.ogg" type="video/ogg">
  Mensagem exibida apenas se o navegador não for compatível
com a tag video.
</video>
```

O elemento *video* é composto por uma ou mais *tags source*, que indicam no atributo *src*, assim como na *tag img*, o local onde o vídeo está hospedado e no atributo *type* define o tipo do arquivo de vídeo. O navegador então irá escolher o formato de vídeo especificado que oferece compatibilidade, e caso não seja possível, irá exibir o texto inserido após as *tags source*.

É possível definir a altura e largura do elemento video utilizando os atributos *width* e *height*. Um atributo *controls* indica que o navegador deverá exibir controles, como botões que pausam ou continuam a reprodução.

A introdução da *tag video* trouxe uma enorme facilidade para a inserção de vídeos nos documentos HTML, mas até o lançamento do HTML5, vários outros recursos foram desenvolvidos e utilizados pelos desenvolvedores para o mesmo fim. A maioria dos desenvolvedores e grandes *sites* utilizava o *plugin Adobe Flash*, que fornecia funcionalidades avançadas em navegadores, para inserir um reprodutor de vídeos capaz de executá-los.

O site mais famoso de reprodução e compartilhamento de vídeos, o *YouTube,* fornece um formulário com opções para gerar um código pronto para ser utilizado em documentos HTML. A vantagem de utilizar o serviço do YouTube está na retro compatibilidade, pois o código fornecerá um vídeo pronto para ser reproduzido em navegadores modernos e antigos. Um código gerado pela ferramenta, com tamanho do vídeo de 640 x 480, selecionando apenas a opção *“Mostrar controles do player*” e desmarcando as demais pode ser visto a seguir:

```
<iframe width="640" height="480"
src="https://www.youtube.com/embed/jNQXAC9IVRw?rel=0&amp;showinfo=0"
frameborder="0" allowfullscreen></iframe>
```

3.3.3 Inserindo tabelas

As tabelas são utilizadas para agrupar, ordenar e alinhar um conteúdo e foram criadas para exibir dados tabulares. Podem conter textos, imagens, vídeos, *links* e vários outros elementos.

O código para uma tabela simples é demonstrado a seguir:

```
<table border="1">
    <caption>Membros</caption>
    <tr>
          <th>Nome</th>
          <th>Idade</th>
    </tr>
    <tr>
          <td>Joaquim</td>
          <td>15</td>
    </tr>
    <tr>
          <td>Lorena</td>
          <td>19</td>
    </tr>
</table>
```

O código demonstrado produz, visualmente, no navegador *Google Chrome*, a tabela da Figura 1.

Figura 1 – Tabela exibida pelo navegador

Membros

Nome	Idade
Joaquim	15
Lorena	19

Fonte: Elaborada pelo autor

A *tag table* define que o conteúdo é uma tabela, e sua estrutura é construída no exemplo pelos elementos *tr*, *th* e *td*. O elemento *table* organiza seu

conteúdo em linhas e colunas. No exemplo, um atributo *border* foi adicionado, exibindo uma borda que envolve cada célula da tabela.

A *tag caption* define um título para a tabela. Seu uso evita que uma título para a tabela tenha que ser inserido fora do elemento *table*.

Atributos adicionais podem ser incluídos na *tag table*, como cellspacing, que controla o espaçamento entre células; cellpadding, que controla espaçamento dentro das células; e *bgcolor*, que altera a cor de fundo da tabela.

Uma linha é definida pela *tag tr*. O conteúdo de uma linha pode ser formado de células comuns, definidas pela *tag td*, ou células de cabeçalho, definidas pela *tag th,* e que por padrão são exibidas em negrito. Várias células podem ser definidas dentro de uma mesma linha.

As células também podem ser controladas com atributos adicionais. Os atributos mais importantes são *cellspan* e *rowspan*, que fazem com que uma célula ocupe mais que um espaço em uma linha ou em uma coluna.

3.3.4 Inserindo links de navegação

As ligações para outras páginas e sites são partes essenciais da maioria dos documentos HTML. As ligações, também chamadas de *links*, dão a possibilidade de transformar um texto ou elemento em objeto clicável, que quando ativado direcionará o usuário para o destino indicado.

As ligações podem ser criadas pela *tag a* (anchor/âncora), envolvendo com sua *tag* de início e de fechamento o conteúdo que será transformado em objeto clicável.

As ligações recebem obrigatoriamente o atributo *href*, que indica para onde o navegador será direcionado assim que o usuário solicitar mediante o clique. O local pode ser uma página no mesmo servidor, o endereço de outro site ou uma página em outro site.

Exemplo de código de uma ligação interna:

```
<a href=”paginas/sobre.html”>Sobre</a>
```

Exemplo de código de uma ligação externa:

```
<a href=”http://www.brasil.gov.br” target=”_blank”>Portal Brasil</a>
```

O atributo *target* pode ser visto no último exemplo. Este atributo é responsável por dizer ao navegador como a ligação deverá ser aberta. No exemplo, foi configurada para abrir a página em uma nova janela utilizando o parâmetro `_blank`. Podem também ser utilizados outros parâmetros, e caso o atributo não seja especificado, o navegador abrirá a página na mesma janela do site atual.

Uma ligação pode também direcionar o usuário para uma seção no mesmo documento ou em documentos externos caso seja adicionado o símbolo “#” seguido do nome da seção.

4 ESTUDO DE CASO – SISTEMA DE CONTROLE DE ENTREGA DE PIZZA

A empresa “Meveana”, pizzaria em expansão, entrou em contato com a Software House SoftPlus informando da necessidade de automatizar seu processo de entrega. Com base nas necessidades identificadas durante as reuniões do analista de sistemas enviado pela SoftPlus com o proprietário da empresa, o Sr. Dorival, com os envolvidos no processo de entrega e demais interessados na automatização do processo, foi feito um estudo para seleção das tecnologias a serem utilizadas no desenvolvimento do software.

4.1 ESTUDO PRELIMINAR E SELEÇÃO DE TECNOLOGIAS APLICÁVEIS

De acordo com os requisitos que foram identificados pela empresa Meveana e a empresa SoftPlus, ficaram evidentes algumas características essenciais para a decisão de quais tecnologias são aplicáveis:

- Permitir que o sistema seja acessado de vários dispositivos;
- Permitir que diferentes empresas tenham acesso ao sistema sem precisar de mudanças em suas políticas de TI;
- Permitir que novos desenvolvedores trabalhem no projeto a longo prazo.

Para atender aos dois primeiros critérios, ficou claro que um sistema web iria suprir melhor às necessidades de flexibilidade no acesso e implantação. Para o último, o website w3techs.com, que realiza pesquisas e fornece informações sobre tecnologias para a web, foi consultado. As pesquisas sobre as linguagens de programação, tanto para servidor (server-side) quanto para cliente (client-side) forneceram tantos relevantes para este estudo, como pode ser visto no Quadro 1 e Quadro 2 mostrados a seguir.

Quadro 1 – Linguagens de programação server-side mais populares

Server-side Programming Languages

Most popular server-side programming languages

© W3Techs.com	usage	change since 1 September 2015
1. PHP	81.4%	+0.1%
2. ASP.NET	16.4%	-0.2%
3. Java	3.0%	
4. static files	1.6%	
5. ColdFusion	0.7%	

percentages of sites

Fonte: www.w3techs.com (2015)

Gráfico 2 – Linguagens de programação client-side mais populares

Client-side Programming Languages

Most popular client-side programming languages

© W3Techs.com	usage	change since 1 September 2015
1. JavaScript	90.3%	+0.4%
2. Flash	9.9%	-0.3%
3. Silverlight	0.1%	

percentages of sites

Fonte: www.w3techs.com (2015)

Levando em conta a popularidade de cada linguagem, tendências de crescimento, defasagem e disponibilidade de profissionais capacitados para trabalho com as mesmas, foram escolhidas as linguagens JavaScript, PHP e ASP.NET para estudo mais detalhado.

4.2 ANÁLISE DA LINGUAGEM PHP

PHP é uma linguagem de programação interpretada, livre, de tipagem fraca, que adota tanto o paradigma procedural quanto o orientado a objetos. Foi criada em 1995 por Rasmus Lerdorf para trabalhar junto com HTML na manipulação de formulários e como linguagem server-side para web básica, e desde então evoluiu com foco na web.

A linguagem é fácil de ser aprendida e utilizada. Deu origem a vários sistemas de sucesso como o Facebook, MediaWiki e Invision Power Board e

continua sendo amplamente utilizada, reformulada e desenvolvida. O desenvolvedor que utiliza PHP pode escolher entre uma variedade de frameworks para facilitar seu trabalho, como Laravel, Zend, Symphony e CodeIgniter. Várias bibliotecas estão disponíveis para as mais variadas tarefas, também reunidas sob o gerenciador de dependências Composer.

Por dar uma liberdade muito grande ao desenvolvedor em relação a forma de escrita do código, a linguagem é frequentemente criticada. Para estes problemas, os desenvolvedores PHP utilizam uma variedade de ferramentas como PHPSniffer e se guiam pelos padrões PSR, que ditam boas práticas para desenvolvimento com a linguagem.

O PHP também conta com vários frameworks de teste unitário e suporte em uma grande diversidade de IDEs, com IDEs criadas até para uso exclusivo com a linguagem.

É fácil de implantação e o interpretador PHP já vem disponível em quase todos os servidores de hospedagem disponíveis, desde os de baixo custo até poderosos servidores dedicados e possui versões nativas para os mais diversos sistemas operacionais. É difícil encontrar um SGBD ao qual o PHP não ofereça conexão fácil e estável.

4.3 ANÁLISE DA TECNOLOGIA ASP.NET

A tecnologia ASP.NET foi criada pela Microsoft como sucessora da tecnologia ASP (também chamada de ASP Clássico) para desenvolvimento de aplicações web. Oferece uma abordagem baseada na orientação a objetos e, embora seja vista como concorrente da linguagem PHP, é uma plataforma que permite que qualquer linguagem suportada do ambiente .NET seja utilizada, como C#.

Enquanto o desenvolvedor PHP tem como apoio para seu aprendizado fóruns de discussão e livros, a Microsoft criou uma grande estrutura de apoio para os programadores que utilizam ASP.NET.

Ao contrário do PHP, que não adota uma IDE como oficial ou principal, o ASP.NET está intimamente ligado ao uso da IDE Visual Studio .NET, trazendo benefícios ligados à padronização de tecnologias entre programadores e melhor suporte por se tratar de um produto mais específico.

Aplicações desenvolvidas em ASP.NET, normalmente, são executadas em servidores IIS (Internet Information Services), com o Framework .NET hospedados na plataforma Windows. Todos esses softwares são proprietários da Microsoft e devem ser devidamente licenciados, o que encarece os custos de hospedagem e limita opções. Outros projetos como Mono e mod_aspdotnet tem como objetivo fornecer meios para execução de aplicações .NET em plataformas Linux, porém não tem popularidade tão grande em comparação com as soluções da Microsoft e, por isso, tem menor suporte.

O Gráfico 3 e Gráfico 4 mostram, respectivamente, os softwares de servidor web e sistemas operacionais mais utilizados no meio.

Gráfico 3 – Servidores Web mais populares e em rápido crescimento

Web Servers

Most popular web servers

© W3Techs.com	usage	change since 1 September 2015
1. Apache	56.2%	-0.4%
2. Nginx	25.9%	+0.7%
3. Microsoft-IIS	12.8%	-0.2%
4. LiteSpeed	2.3%	
5. Google Servers	1.3%	

percentages of sites

Fastest growing web servers since 1 September 2015

© W3Techs.com	sites
1. Nginx	1,354
2. LiteSpeed	88
3. Google Servers	56

daily number of additional sites in the top 10 million

Find more details in the web server surveys

Fonte: www.w3techs.com (2015)

Gráfico 4 – Sistemas operacionais mais populares

Operating Systems

Most popular operating systems

© W3Techs.com	usage	change since 1 September 2015
1. Unix	67.2%	+0.1%
2. Windows	32.8%	-0.1%

percentages of sites

Fonte: www.w3techs.com (2015)

4.4 ANÁLISE DA LINGUAGEM JAVASCRIPT

JavaScript é uma linguagem criada em 1995 por Brendan Eich enquanto trabalhava na empresa Netscape. É interpretada, multi-paradigma e originalmente criada para uso em navegadores *web*.

Durante muitos anos, a linguagem foi menosprezada por vários desenvolvedores, principalmente por considerarem a linguagem fraca. Não só isso, como também sofreu os mesmos efeitos que o HTML durante a guerra dos navegadores, pois cada navegador desenvolveu sua própria implementação da linguagem: a Netscape inicialmente criou a linguagem como *LiveScript* e a Microsoft como *JScript*.

Com o passar dos anos, a linguagem começou a ser cada vez mais levada a sério, principalmente após a empresa Google demonstrar o que era possível realizar com ela após o lançamento da aplicação web Google *Maps*. Hoje, a linguagem é utilizada por uma variedade enorme de profissionais: desde não-especialistas em programação, que precisam da linguagem para enriquecer suas páginas *web*, até engenheiros de software que criam aplicações complexas para navegadores *web,* e mesmo do lado do servidor, impulsionado pelo sucesso do *Node.js*.

Para nosso estudo e possível aplicação, a linguagem foi escolhida por sua versatilidade, maturidade a ampla adoção na criação de aplicações *web*. Dispõe de vários *frameworks* bem desenvolvidos e apoiados por grandes empresas, como o *Angular.js* e *jQuery*. Executa nativamente com alta performance nos navegadores de uso comum e atualmente dispensa o uso de plug-ins como Adobe

Flash, que não tem bom suporte em dispositivos móveis, que são parte essencial dos requisitos de desenvolvimento do software.

4.5 CONCLUSÃO DO ESTUDO DE CASO

De acordo com os dados expostos na análise das tecnologias disponíveis, foi possível escolher as linguagens a serem utilizadas para programação do servidor e do cliente.

A linguagem *JavaScript* oferece todo apoio para uma experiência rica, moderna e fluida do lado do cliente, com uma alta disponibilidade de profissionais e maiores chances de que a maioria dos envolvidos no desenvolvimento da aplicação, mesmo que não sejam programadores, tenham familiaridade com o uso de linguagem.

Ficou decidido também que, entre as linguagens PHP e ASP.NET, a primeira atende melhor às demandas de desenvolvimento, implantação e de custo. Assim como no uso do *JavaScript*, é muito fácil encontrar profissionais capacitados em PHP.

Os estudo forneceu resultados satisfatórios e a partir dos dados obtidos é possível prosseguir para análises de mais alto nível.

5 MÉTODO ENTERPRISE KNOWLEDGE DEVELOPMENT – EKD

5.1 CONCEITO

Para uma implantação correta de equipamentos e sistemas de informação em empresas, é necessário que seus processos sejam atualizados. Para auxiliar nesta tarefa, a Modelagem Organizacional fornece meios de entender melhor os requisitos e propor alternativas para os processos da organização. Uma das metodologias disponíveis é o EKD.

O EKD é uma técnica de modelagem organizacional que fornece uma forma sistemática de analisar uma organização. É uma atividade valiosa para o desenvolvimento de software, mais precisamente para a engenharia de requisitos, permitindo o “estreitamento dos laços” entre as áreas de negócio e de tecnologia da informação. Além disso, de acordo com Pádua, “O EKD fornece uma base para o entendimento e apoio às mudanças organizacionais e ajuda o desenvolvimento de sistemas de informação que apoiará à organização”.

5.2 OBJETIVOS

É importante que os processos sejam também mapeados e documentados para que a empresa possa se preparar, reagir e tirar proveito de mudanças.

Esta técnica também permite representar de forma estruturada o conhecimento organizacional, apresentando de forma detalhada os aspectos relacionados a funcionalidade dos processos e que passos levam a sua conclusão. Também permite representar e agregar os objetivos, restrições e regras de negócio da organização.

Assim, aplicada a técnica de modelagem e documentadas as informações da empresa, é possível recorrer a estas informações quando surge uma necessidade de mudança, para detectar problemas, na criação de novos sistemas de negócio ou para entender requisitos na engenharia de software. O objetivo então é chegar à criação de um documento chamado de repositório de conhecimento.

5.3 SUBMODELOS DO EKD

O Modelo Organizacional resultante da técnica utilizada pelo EKD é dividido em alguns submodelos, que podem ser conferidos a seguir.

5.3.1 MO – Modelo de Objetivos

Este modelo identifica e descreve os objetivos da organização e as questões associadas a fim de atingi-los. Descreve a razão para a atividades e entidades de outros modelos e explica porque objetivos existem ou não existem. Contém os seguintes componentes: objetivo, problema, causa, restrição e oportunidade.

Segue uma breve descrição de cada componente:

- Os objetivos especificam o que a organização e seus empregados desejam alcançar ou evitar.
- Problemas especificam estados que o ambiente está ou pode ficar que possam dificultar que a empresa alcance os objetivos.
- Causas expressam razões ou explicações para os problemas, geralmente fora do controle do projeto, processo ou organização.
- Restrições expressam restrições do negócio ou políticas **externas** que afetam componentes e ligações do Modelo Organizacional.
- Oportunidades expressam recursos que podem facilitar o processo para alcançar os objetivos, como novas tecnologias.

5.3.2 MRN – Modelo de Regras de Negócio

O MRN define e mantém regras de negócio formuladas e consistentes com o Modelo de Objetivos e como se relacionam com ele.

A modelagem das regras estão intimamente relacionadas com o Modelo de Objetivos, pois são definidas por objetivos e podem prejudicar outros objetivos. Fazem referência a conceitos definidos no Modelo de Conceitos, regras de

negócio são definidas por atores Modelo de Atores e podem ser motivadas por componentes do Modelo de Requisitos e Componentes Técnicos.

5.3.3 MC – Modelo de Conceitos

O Modelo de Conceitos tem uso mais restrito e serve para definir fenômenos e coisas que estão em outros modelos. Se um objetivo é definido como “agilizar os processos de entrega do setor de entregas”, aqui teremos a definição do “setor de entregas”.

5.3.4 MPN – Modelo de Processo de Negócio

Este modelo é usado para analisar e definir processos de negócio, como interagem e como lidam com a informação e material. Presume-se que um processo de negócio consome uma entrada em termos de informação ou material e produz uma saída, também sendo informação ou material.

De forma geral, o MPN é bem similar aos tradicionais diagramas de fluxo de dados (DFD).

5.3.5 MAR – Modelo de Atores e Recursos

Este modelo define quais atores e recursos estão envolvidos nas atividades organizacionais, quais deles se relacionam entre si e com os componentes do Modelo de Objetivos e do Modelo de Processo de Negócio. Geralmente podem ser identificados com perguntas, como: quem inicia, controla ou encerra determinadas as atividades? Quais são as responsabilidades de se reportar entre eles?

5.3.6 MRCT – Modelo de Requisitos e Componentes Técnicos

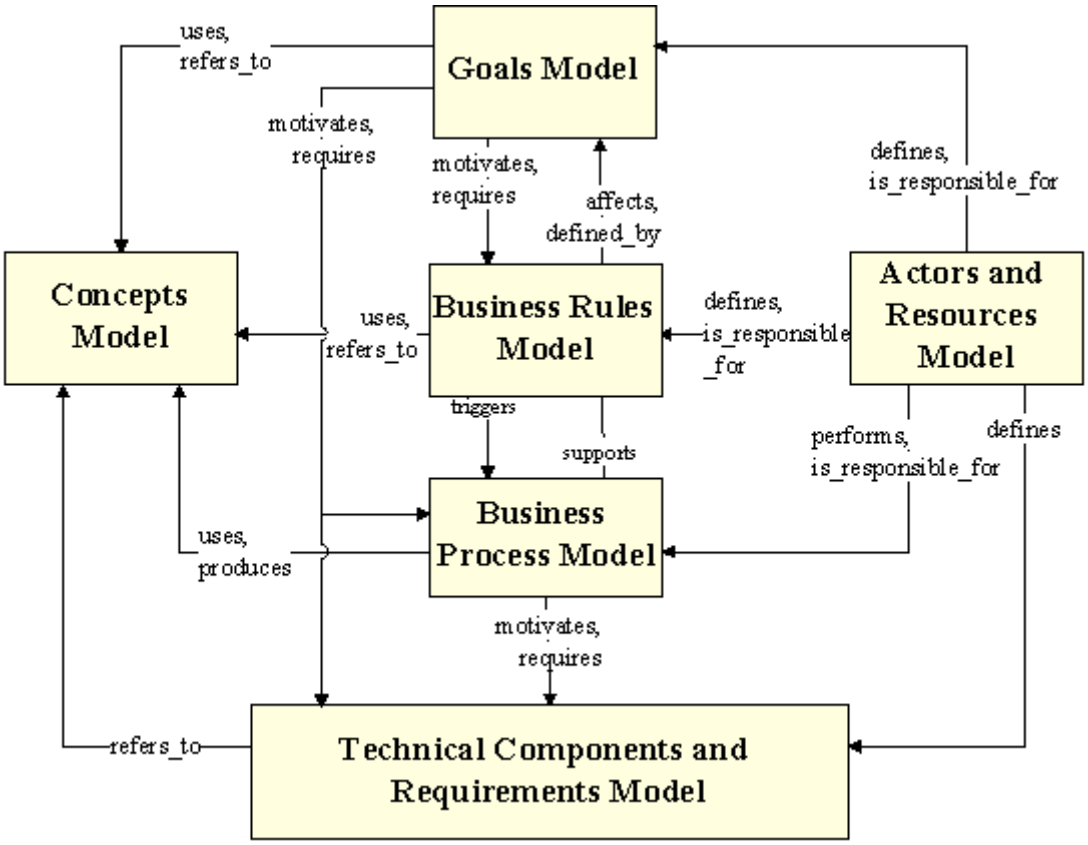
O Modelo de Requisitos e Componentes Técnicos se torna relevante quando o objetivo do uso do EKD é apoiar a definição de requisitos técnicos para a elaboração e criação de um sistema de informação. Normalmente responde às perguntas: quais são os requisitos para que o sistema seja desenvolvido, quais

requisitos são gerados por quais processos de negócio e qual o potencial de novas tecnologias para melhorar os processos.

5.4 RELACIONAMENTO ENTRE OS SUBMODELOS DO EKD

Os submodelos do EKD estão intimamente relacionados, afetando uns aos outros e se complementando. A Figura 2 ilustra como os submodelos estão relacionados.

Figura 2 – Relacionamento entre submodelos do EKD



Fonte: http://people.dsv.su.se/~js/ekp/ekd_method.html (2015)

6 RESENHA

REENLSOBER, Daniel. Análise Comparativa entre as Linguagens de Scripting ServerSide PHP e ASP .NET. Jaguariúna, 2006. 38 p.

A obra – A obra é uma monografia apresentada pelo autor Daniel Reenlsober, durante seu curso de graduação em Ciência da Computação da Faculdade de Jaguariúna no ano de 2006, cidade de Jaguariúna, apresentada num total de 38 páginas e publicada na internet.

Autor – Daniel Reenlsober foi aluno da Faculdade de Jaguariúna, se formando no ano de 2006 em que escreveu esta obra. Durante o desenvolvimento da obra teve como professor orientador o Prof. Ademário Araújo Jr.

Resumo – A obra tem como objetivo expor os resultados da pesquisa do autor sobre as linguagens PHP e ASP.NET, utilizando-se de exemplos das duas linguagens para efeitos de comparação. Também analisa fatores externos e ferramentas não-oficiais produzidas por terceiros que podem ser utilizadas para auxiliar e agilizar na criação de aplicativos utilizando estas linguagens.

Inicia o capítulo 1 com uma introdução onde descreve rapidamente os primórdios da World Wide Web e a revolução que as linguagens server-side trouxeram, transformando páginas estáticas em dinâmicas. Também introduz as linguagens comparadas pela obra, informando suas principais características, processo de evolução e popularidade.

No capítulo 2 o autor introduz os objetivos de seu estudo, listando os tópicos a serem abordados. O objetivo da obra é indicar qual linguagem é apropriada para diferentes tipos de projetos para web.

No capítulo 3, metodologia, é explicada a metodologia utilizada no estudo. O trabalho foi iniciado por meio de pesquisas e criação de exemplos para PHP e ASP.NET. Após a concussão dos projetos, é feita uma comparação entre as linguagens quanto a facilidade de aprendizado, complexidade do código, desempenho no servidor, segurança, implementação da aplicação junto ao design e tamanho do código. Logo depois, é feita uma análise do custo de desenvolvimento de cada linguagem.

O capítulo 4 apresenta os resultados do estudo, introduzindo uma comparação entre as configurações mínimas para uso das linguagens em ambiente de produção e de desenvolvimento.

Na comparação do custo de aquisição, são levantados os aspectos sobre custo de hardware e de software. Relaciona as configurações mínimas com custos de hardware, como também analisa custos de licenciamento de sistema operacional e ferramentas de programação: o PHP sai na frente, podendo ser utilizado com nenhum custo de aquisição de ferramentas nem de servidor, pois todo matéria necessário é gratuito.

O custo de hospedagem também é comparado, relacionando a possibilidade de, com PHP, utilizar Windows ou Linux, enquanto que com ASP.NET só é possível utilizar hospedagens Windows. O autor levanta dados mostrando que, em média, hospedagens com tecnologia ASP.NET são cerca de 6% a 29% mais caras que hospedagem PHP. O preço mais alto é atribuído também ao fato do Windows ser um software proprietário pago e ser mais inseguro que o Linux, sendo frequente a adição de um firewall, muitas vezes também pago.

Na comparação de custo de treinamento e suporte, o autor enfatiza o suporte que a Microsoft dá para desenvolvedores que trabalham com sua plataforma, auxiliando-os em vários meios de comunicação. Dá ponto negativo ao PHP por concluir que os desenvolvedores que utilizam esta linguagem, geralmente, dependem muito de fóruns de discussão e livros.

O autor também compara os salários utilizando como base o site curriculum (www.curriculum.com.br) na data de 1 de novembro de 2006, concluindo que o programador ASP.NET tem um salário médio maior que desenvolvedor PHP, acarretando em custos maiores para as empresas. Justifica esta diferença citando o crescimento do uso do ASP.NET e pelo fato dos programadores poderem certificar-se junto a Microsoft.

Comparando as ferramentas de desenvolvimento, o autor faz uma análise detalhada do Visual Web Developer da Microsoft, utilizado no desenvolvimento com ASP.NET, citando recursos visuais e de geração automática de código. Usa como exemplo de ferramenta para PHP o software Dreamweaver, da Adobe, concluindo que neste quesito a ferramenta da Microsoft é superior, dando uma vantagem para a linguagem da empresa.

Na comparação de segurança, é lembrado mais uma vez que pelo fato do Windows ser um sistema mais visado por hackers, o uso de ASP.NET necessita de mais cuidados com segurança. São citados os servidores principais utilizados em cada linguagem: Apache para PHP e IIS para ASP.NET. O ASP.NET é elogiado por fornecer código seguro através de geração automática de código e assistentes de configuração, enquanto no PHP o desenvolvedor precisa codificar manualmente ou utilizar código de terceiros, ambas opções oferecem riscos. Vários trechos de código das duas linguagens e capturas de tela são fornecidos.

Em comparação sobre o reuso de código da linguagem, é elogiada a facilidade de codificação do PHP junto ao HTML, porém o risco do código ficar extenso e de difícil manutenção, citando o framework de *template* Smarty como solução. As ferramentas visuais e organização de código em arquivos separados do ASP.NET dão pontos positivos para a linguagem.

Na implementação junto ao design, são usados como exemplo de ferramentas, novamente, o Dreamweaver e o Visual Web Developer, sem detalhar pontos negativos ou vantagens entre elas.

Conclusão do autor – Finalmente, o autor pôde concluir que não é possível comparar o PHP com ASP.NET na forma de programar, pois utilizam paradigmas de programação diferentes, sendo o ASP.NET orientado a objetos. Neste caso, indica o PHP para desenvolvedores com mais familiaridade com o paradigma de programação estruturada.

Enalteceu o fato do PHP ter menor custo para desenvolvimento, ao contrário do ASP.NET que demanda maior investimento, mas deu destaque ao ASP.NET quanto à segurança, lembrando que a ferramenta ajuda, mas é necessário competência do desenvolvedor.

Por fim, conseguiu atingir o objetivo de seu estudo, recomendando o ASP.NET em detrimento do PHP para projetos maiores.

Crítica – Apesar de ser um estudo desatualizado, uma boa parte dos profissionais atuais ainda concordam com a conclusão do autor, recomendando, entre as duas linguagens, ASP.NET para projetos maiores.

O autor tropeça em alguns termos, confundindo recursos de desenvolvimento de telas com recursos visuais e paradigmas de programação.

Alguns aspectos são meramente citados e não são comparados e o estudo compara a linguagem PHP, que de fato é uma linguagem de programação, com ASP.NET, que não é uma linguagem, mas sim linguagem de programação.

Indicação de leitura – A obra, apesar de fornecer informações defasadas e algumas confusas, serve bem para iniciantes que, em dúvida de qual tecnologia estudar, quiserem uma comparação superficial do PHP e ASP.NET. Entretanto, no aprendizado e estudo para mercado de trabalho, é recomendado procurar fontes mais atualizadas sobre o assunto.

7 CONCLUSÃO

Durante o estudo e pesquisa sobre os assuntos deste trabalho, foi possível demonstrar como algumas das principais tecnologias utilizadas para desenvolvimento web surgiram e são utilizadas atualmente.

É necessário analisar, para cada projeto, quais tecnologias dentre um leque gigantesco de opções podem ser escolhidas para uso, levando em conta aspectos de custo, tempo e disponibilidade de profissionais capacitados. A metodologia do estudo, com dados estatísticos sobre as tecnologias estudadas, pôde ajudar na solução do problema apresentado pelo estudo de caso.

REFERÊNCIAS

COOK, Graig; GARBER, Jason. **Foundation HTML5 with CSS3:** A Modern Guide and Reference. Edição 2012. Apress, 2012

EKD - Enterprise Knowledge Development. Publicação em inglês sobre o método EKD Disponível em: < http://people.dsv.su.se/~js/ekp/ekd_method.html>. Acesso em: 8 out. 2015.

FREEMAN, Eric T.; ROBSON, Elisabeth. **Head First JavaScript Programming.** O'Reilly Media, 2014

PÁDUA, S.I.D. **Investigação do processo de desenvolvimento de software a partir da modelagem organizacional, enfatizando regras de negócio.** 2000. Dissertação(Mestrado). Escola de Engenharia de São Carlos - USP. Disponível em: <http://www.teses.usp.br/teses/disponiveis/18/18140/tde-09122008-154855/publico/DissertacaoSilviaInesDallavallePadua.pdf>. Acesso em: 8 out. 2015.

REENLSOBER, Daniel. **Análise Comparativa entre as Linguagens de Scripting ServerSide PHP e ASP .NET.** 2006. Monografia (Bacharelado em Ciência da Computação) – Curso de Ciência da Computação da Faculdade de Jaguariúna, Jaguariúna. Disponível em: <http://bibdig.poliseducacional.com.br/document/?view=103>. Acesso em: 8 out. 2015.

SKLAR, David; TRACHTENBERG, Adam. **PHP Cookbook**: Solutions & Examples for PHP Programmers. 3. ed. O'Reilly Media, 2014