



James A. O'Brien
George M. Marakas

ADMINISTRAÇÃO DE **SISTEMAS DE INFORMAÇÃO**

15ª Edição



**Mc
Graw
Hill**





O12a O'Brien, James A.

Administração de sistemas de informação [recurso eletrônico] / James A. O'Brien, George M. Marakas ; tradução: Rodrigo Dubal ; revisão técnica: Armando Dal Colletto. – 15. ed. – Dados eletrônicos. – Porto Alegre : AMGH, 2013.

Editado também como livro impresso em 2013.
ISBN 978-85-8055-111-2

1. Administração de empresas. 2. Sistemas de informação.
I. Marakas, George M. II. Título.

CDU 658:004



| Categoria do software | O que faz | Produto IBM | Clientes | Principal concorrente | Clientes |
|--------------------------------|---|-------------|--|------------------------------|--|
| Gerenciamento de rede | Monitora redes para controlá-las e mantê-las funcionando. | Tivoli | T. Rowe Price o utiliza para proteger registros de clientes. | HP OpenView | Amazon.com o utiliza para monitorar seus servidores. |
| Servidor de aplicativo | Transfere dados entre aplicativos de negócios e a web. | WebSphere | REI o utiliza no site da empresa e para distribuir dados. | BEA WebLogic | Washingtonpost.com o utiliza para desenvolver novas páginas. |
| Gerenciador de banco de dados | Proporciona armazenamento digital para dados da empresa. | DB2 | Mikasa o utiliza para ajudar os clientes a encontrar produtos <i>on-line</i> . | Oracle 11g | Southwest Airlines o utiliza para os programas de fidelidade do cliente. |
| Ferramentas de colaboração | Aumenta a capacidade de todos os recursos – do <i>e-mail</i> aos calendários eletrônicos. | Lotus | São utilizadas pela Sephora para coordenar a manutenção da loja. | Microsoft Exchange | Time Inc. o utiliza para fornecer <i>e-mail</i> para seus funcionários. |
| Ferramentas de desenvolvimento | Permite ao programador criar códigos de <i>software</i> com rapidez. | Rational | Merrill Lynch as usa para desenvolver códigos para transações <i>on-line</i> . | Microsoft Visual Studio .NET | Usado para desenvolver o sistema de gerenciamento. |

FIGURA 4.17 Comparação entre *softwares* de sistemas oferecidos pela IBM e pelos principais concorrentes.

Outros exemplos de programas de suporte de sistema incluem monitores de desempenho e de segurança. Os **monitores de desempenho** são programas que monitoram e ajustam o desempenho e a utilização de um ou mais sistemas de computador para manter a eficácia da execução. Os monitores de segurança são pacotes que monitoram e controlam o uso dos sistemas de computador e fornecem mensagens de alerta, além de registrarem evidências de uso não autorizado de recursos do equipamento. Uma tendência recente é a fusão dos dois tipos de programas em sistemas operacionais como o Windows 2008 Datacenter Server da Microsoft ou em *software* de gerenciamento de sistemas como o CA-Unicenter da Computer Associates, com capacidade para gerenciar tanto sistemas de grande porte como servidores de um centro de dados.

Outra tendência importante em termos de *software* é o uso do *software* de sistema conhecido como **servidores de aplicação**, que oferece interface *middleware*, entre sistema operacional e programas de aplicação dos usuários. **Middleware** consiste em um *software* que auxilia diversas aplicações de *software* e sistemas de computação em rede na troca de dados e no trabalho conjunto mais eficiente. Entre os exemplos, estão os servidores de aplicação, servidores web e *software* de integração de aplicação corporativa (*enterprise application integration* – EIA). Desse modo, por exemplo, os servidores de aplicação, como WebLogic da BEA e WebSphere da IBM, ajudam as aplicações de comércio e negócio eletrônico baseadas na web a executar com mais rapidez e mais eficácia em computadores com Windows, Unix e outros sistemas operacionais.

Linguagem de programação

Para entender melhor a respeito de *software*, é necessário conhecimento básico do papel da linguagem de programação no desenvolvimento dos programas do computador. A **linguagem de programação** permite ao programador criar o conjunto de instruções que constitui o programa do computador. Existem diversas linguagens de programação, cada uma com vocabulário, gramática e uso exclusivos.

Linguagem de máquina

Linguagem de máquina (ou *linguagem de primeira geração*) é o nível mais básico de linguagem de programação. Nos estágios iniciais do desenvolvimento do computador, todas as instruções de programação tinham de ser escritas usando códigos binários exclusivos para cada computador. Esse tipo de programação envolve a difícil tarefa de escrever instruções na forma de seqüências de dígitos binários (um e zero) ou de outros sistemas numéricos. Os programadores precisam conhecer em detalhes as operações internas do tipo específico de CPU que estão usando e eles

| Quatro níveis de linguagem de programação | |
|---|---|
| <ul style="list-style-type: none"> Linguagem de máquina: Uso de instruções de código binário 1010 11001 1011 11010 1100 11011 | <ul style="list-style-type: none"> Linguagem de alto nível: Uso de instruções curtas ou anotações aritméticas BASIC: $X = Y + Z$ COBOL: COMPUTE $X = Y + Z$ |
| <ul style="list-style-type: none"> Linguagem “montadora”: Uso de instruções codificadas com símbolos LOD Y ADD Z STR X | <ul style="list-style-type: none"> Linguagem de quarta geração: Uso de instruções naturais e não procedural SUM THE FOLLOWING NUMBERS (SOMAR OS SEGUINTE NÚMEROS) |

FIGURA 4.18 Exemplos de quatro níveis de linguagem de programação. Essas instruções em linguagem de programação devem ser usadas para calcular a soma de dois números conforme a expressão $X = Y + Z$.

devem escrever séries longas de instruções detalhadas para realizar cada tarefa simples de processamento. A programação em linguagem de máquina requer especificação de locais de armazenamento para cada instrução e item de dados utilizados, e é necessário incluir instruções para cada chave e indicador usados pelo programa. Esses requisitos tornam a programação de linguagem de máquina uma tarefa difícil e propensa a erro. Uma programação em linguagem de máquina para somar dois números juntos na CPU de um computador específico e armazenar o resultado deve ter a forma mostrada na Figura 4.18.

Linguagem “montadora” (ou *linguagem de segunda geração*) é o nível seguinte da linguagem de programação. Ela foi criada para reduzir as dificuldades de escrever os programas em linguagem de máquina. O uso da linguagem “montadora” requer um programa conversor de linguagem denominado “*montador*” (*assembler*) que permite ao computador converter as instruções dessa linguagem em instruções da máquina. A linguagem “montadora” frequentemente é chamada linguagem simbólica, porque símbolos são usados para representar códigos de operação e locais de armazenamento. Abreviações alfabéticas convenientes denominadas *mnemônicas* (que ajudam a memorizar) e outros símbolos representam os códigos de operação, locais de armazenamento e elementos de dados. Por exemplo, o cálculo $X = Y + Z$, em uma linguagem montadora, deve ficar conforme indica a Figura 4.18.

A linguagem “montadora” ainda é utilizada como método de programação em linguagem orientada à máquina. A maioria dos fabricantes de computadores fornece uma linguagem “montadora” que reflete o conjunto de instruções em linguagem de máquina exclusiva de uma linha específica de equipamentos. Essa característica é interessante para os *programadores de sistema*, que programam *software* de sistema (ao contrário dos programadores de aplicação, que programam *software* de aplicação), porque propicia mais controle e flexibilidade na criação de um programa para um computador específico. Eles conseguem produzir *softwares* mais eficientes, ou seja, programas que exijam mínimo de instruções, armazenamento e tempo de CPU para executar uma tarefa específica de processamento.

A **linguagem de alto nível** (ou *linguagem de terceira geração*) utiliza instruções, denominadas *sentenças*, que adotam frases curtas ou expressões aritméticas. Cada sentença em linguagem de alto nível consiste, na verdade, em *macroinstruções*, ou seja, cada uma gera diversas instruções para a máquina quando convertida em linguagem de máquina pelos programas conversores de linguagem de alto nível, denominados *compiladores* ou *interpretadores*. As sentenças de linguagem de alto nível lembram frases ou expressões matemáticas necessárias para expressar o problema ou o procedimento programado. A *sintaxe* (vocabulário, pontuação e regras gramaticais) e a *semântica* (significado) dessas sentenças não refletem o código interno de nenhum computador específico. Por exemplo, o cálculo $X = Y + Z$ seria programado em linguagem de alto nível Basic e Cobol como mostra a Figura 4.18.

As linguagens de alto nível, como Basic, Cobol e Fortran, são mais fáceis de aprender e programar que a linguagem “montadora”, porque as regras, formas e sintaxes são menos rígidas. No entanto, os programas de linguagem de alto nível normalmente são menos eficientes que os de linguagem “montadora” e levam mais tempo para serem convertidos em instruções da máquina.

Linguagem
“montadora”

Linguagem de alto
nível

Linguagem de quarta geração

Como a maioria das linguagens de alto nível é independente da máquina, os programas escritos nessa linguagem não precisam ser reprogramados quando um novo computador é instalado, e os programadores não precisam aprender uma linguagem diferente para cada tipo de computador.

A expressão **linguagem de quarta geração** descreve diversas linguagens de programação mais voltadas à conversação e menos aos procedimentos do que as linguagens anteriores. É chamada linguagem de quarta geração para ser diferenciada da linguagem de máquina (primeira geração), linguagem montadora (segunda geração) e linguagem de alto nível (terceira geração).

A maioria das linguagens de quarta geração consiste em *linguagem não procedural* que incentiva usuários e programadores a especificar os resultados desejados, enquanto o computador determina a sequência de instruções que produzirá tais resultados. Portanto, a linguagem de quarta geração simplifica o processo de programação. A **linguagem natural** às vezes é considerada linguagem de *quinta geração* e fica bem próxima do inglês ou de outras línguas humanas. A atividade de pesquisa e desenvolvimento de inteligência artificial está desenvolvendo linguagens de programação fáceis de usar, como as de uma conversa comum na língua materna de uma pessoa. Por exemplo, Intellect, uma linguagem natural, utilizaria uma sentença como “Qual é a pontuação média do teste MIS 200?” para programar uma simples tarefa de calcular a média de pontos de um teste.

No início da linguagem de quarta geração, os resultados indicaram que ela não comportava ambientes de processamento de alto volume de transações. Se, de um lado, a linguagem de quarta geração caracteriza-se pela facilidade do uso, do outro, é considerada menos flexível do que suas antecessoras, principalmente por causa da necessidade de mais capacidade de armazenamento e de velocidade de processamento. Em um ambiente de grande volume de dados como o de hoje, a linguagem de quarta geração é muito utilizada e não é mais vista com essa característica de alternância entre facilidade de uso e flexibilidade.

Geração automática de códigos

Vinte anos atrás, o engenheiro de *software* Fred Brooks celebrenemente observou que não havia nenhum bala de prata capaz de matar “o monstro de compromissos perdidos, orçamentos estourados e produtos defeituosos”. Hoje, a criação de *software* pode parecer tão cara, sujeita a erros e difícil como sempre, e ainda assim o progresso continua. Embora ainda não haja nenhuma bala de prata à vista, uma série de novas técnicas promete impulsionar ainda mais a produtividade de um programador, pelo menos em alguns domínios de aplicativos.

As técnicas abrangem um amplo espectro de métodos e resultados, mas todas visam à geração automática de *software*. Normalmente, elas geram código a partir de projetos de alto nível e capazes de serem lidos por máquina ou a partir de linguagens de domínio – assistidos por compiladores avançados – que às vezes podem ser usados por não programadores.

Gordon Novak, professor de ciência da computação da Universidade do Texas, em Austin, e membro do Laboratório de Inteligência Artificial da universidade, está trabalhando em “programação automática” – usando bibliotecas de versões genéricas de programas, como algoritmos – para classificar ou encontrar itens em uma lista. Mas, ao contrário de sub-rotinas tradicionais, que possuem interfaces simples mas rígidas e são iniciados por outras linhas de código do programa, a sua técnica funciona em um nível superior e, portanto, mais flexível e fácil de usar.

Os usuários de Novak constroem “visões” que descrevem os dados e princípios do aplicativo e depois conectam as visões por setas nos diagramas que mostram as relações entre os dados. Os esquemas são, em essência, gráficos de altíssimo nível do programa desejado. Eles são compilados de forma que personalizam os algoritmos genéricos armazenados para o problema específico do usuário, e o resultado é um código fonte comum, como C, C++ ou Java.

Novak diz ter sido capaz de gerar 250 linhas de código-fonte de um programa de indexação em 90 segundos com o seu sistema. Isso é equivalente a uma semana da produtividade de um programador médio usando uma linguagem tradicional. “Você está descrevendo seu programa em um nível superior”, diz ele. “E o meu programa está dizendo o seguinte: ‘posso adaptar o algoritmo para a sua aplicação de forma gratuita’.”

Douglas Smith, cientista principal do Kestrel Institute, uma empresa de pesquisa de ciência computacional sem fins lucrativos de Palo Alto, Califórnia, está desenvolvendo ferramen-

tas para “automatizar conhecimentos e inseri-los no computador”. Um programador começa com o Specware da Kestrel, que é uma linguagem genérica de quinta geração que especifica as funções de um programa sem ligação com a linguagem de programação final, a arquitetura do sistema, os algoritmos, as estruturas de dados, e assim por diante. O Specware baseia-se em uma biblioteca de componentes, mas estes não são o código. Eles estão em um nível superior e incluem o conhecimento de design e princípios sobre algoritmos, estruturas de dados, e assim por diante. Smith os chama de “modelos abstratos”.

Além disso, o Specware pode produzir provas de que o código de trabalho está “correto” – isto é, que obedece aos requisitos colocados pelo usuário (que, evidentemente, podem conter erros). “Alguns clientes querem isso para aplicativos de alta garantia, sem falhas de segurança”, diz Smith. A Kestrel trabalha para a Nasa e agências de segurança militar dos Estados Unidos.

“É uma linguagem para escrever os requisitos do problema, uma instrução de alto nível daquilo que a solução deve ser, sem dizer como resolver o problema”, diz Smith. “Consideramos que isso é a fronteira final em engenharia de *software*. É o que os analistas de sistemas fazem.”

Fonte: Adaptado de Gary Anthes, “In the Labs: Automatic Code Generators”, *Computerworld*, 20 de março de 2006.

A **linguagem orientada a objetos**, como Visual Basic, C++, e Java, também é considerada linguagem de quinta geração e tornou-se ferramenta principal no desenvolvimento de *softwares*. Resumidamente, enquanto a maioria das demais linguagens de programação separa elementos de dados dos procedimentos ou das ações a serem executadas com base neles, a linguagem orientada a objetos une tudo e forma objetos. Desse modo, um objeto é constituído por dados e ações a serem executadas com base nesses dados. Por exemplo, um objeto pode ser um conjunto de dados sobre a conta poupança de um cliente do banco e as operações (como cálculo de juros) a serem executadas com base nesses dados; ou pode ser dados gráficos, como uma tela de vídeo e as ações a serem executadas nessa tela (ver Figura 4.19).

Na linguagem procedural, o programa é composto de procedimentos para executar ações com base em cada elemento de dado. No sistema orientado a objetos, estes orientam outros objetos a executar as ações com base em si mesmos. Por exemplo, para abrir uma janela na tela do computador, o objeto menu inicial envia ao objeto janela uma mensagem para se abrir, e a janela aparece na tela. Isso ocorre porque o objeto janela contém o código de programação para abrir a si mesma.

Linguagem orientada a objetos

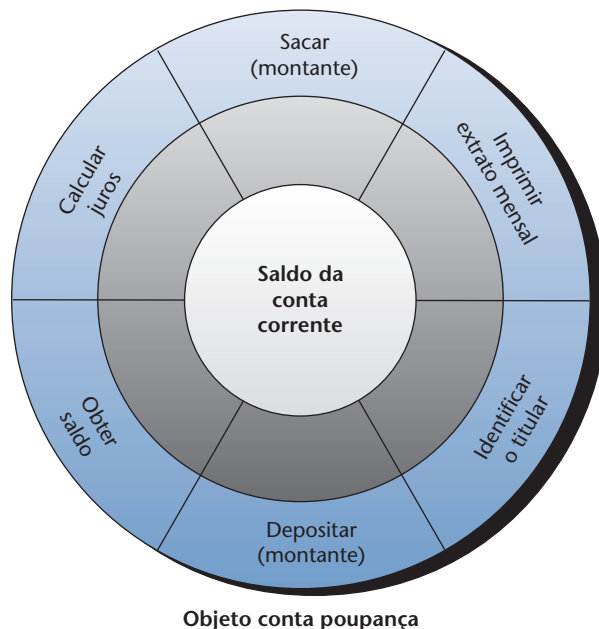
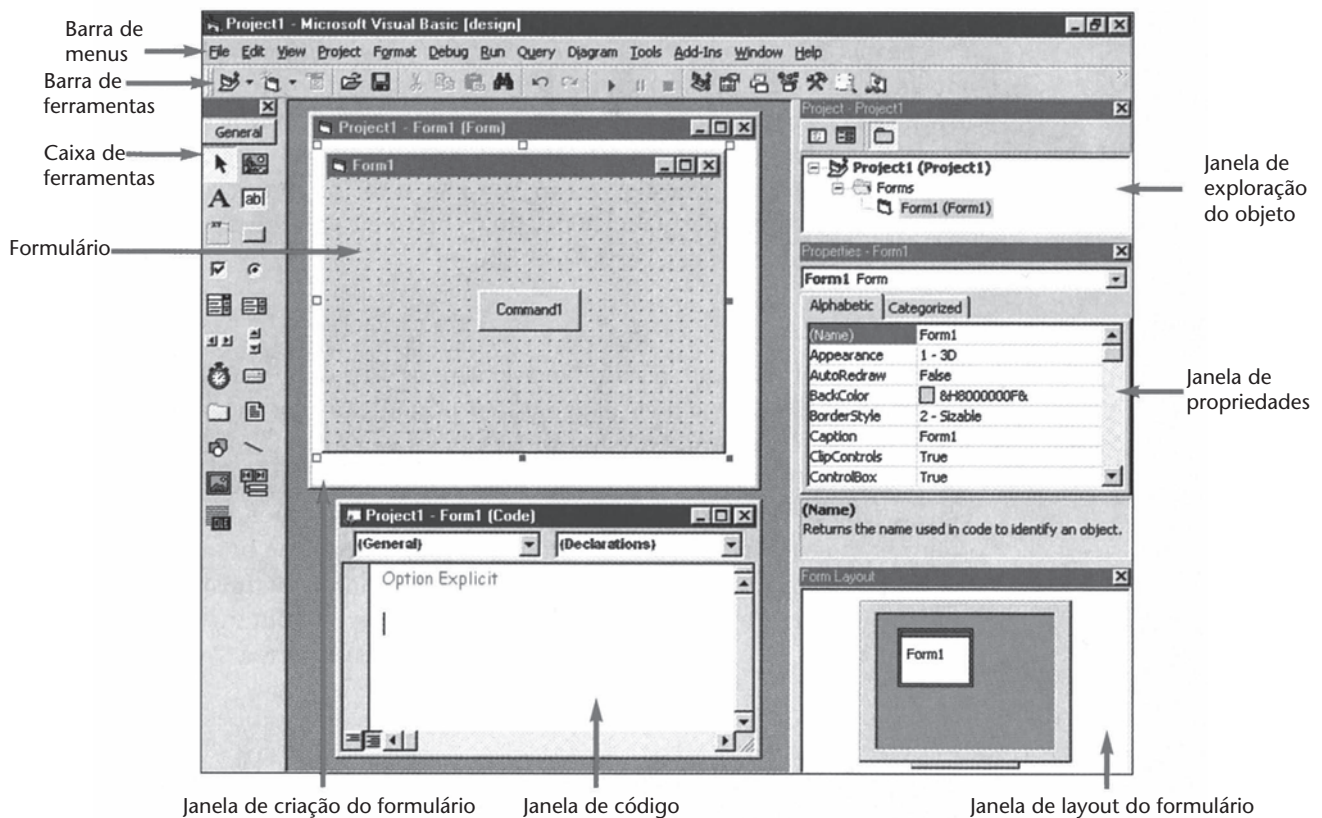


FIGURA 4.19 Exemplo de objeto conta poupança bancária. Esse objeto é constituído por dados sobre o saldo da conta do cliente e de operações básicas que podem ser executadas com base nesses dados.



Fonte: Cortesia da Microsoft®.

FIGURA 4.20 Ambiente de programação orientada a objetos Visual Basic.

A linguagem orientada a objetos é mais fácil de usar e mais eficiente na programação de interfaces gráficas do usuário necessárias em muitas aplicações. Portanto, é a linguagem de programação mais utilizada atualmente no desenvolvimento de *softwares*. Além disso, uma vez programados os objetos, eles podem ser reutilizados. Portanto, a reutilização de objetos é um grande benefício da programação orientada a objetos. Por exemplo, os programadores podem criar uma interface do usuário de um novo programa, montando objetos padrão, como janelas, barras, caixas, botões e ícones. Assim, a maioria dos pacotes de programação orientada a objetos oferece uma interface gráfica do usuário que suporta montagem visual de objetos por meio de operações de apontar e clicar, arrastar e soltar com o mouse, conhecida como *programação visual*. A Figura 4.20 mostra a tela de um ambiente de programação orientada a objetos Visual Basic. A tecnologia orientada a objetos será discutida no Capítulo 5, juntamente com os bancos de dados orientados a objetos.

Serviços e linguagens da web

As três linguagens de programação que consistem em ferramentas importantes para a criação de páginas multimídia, *sites* e aplicações baseados na web são: HTML, XML e Java. Além disso, XML e Java tornaram-se componentes estratégicos das tecnologias de *software* que estão auxiliando muitas das iniciativas de serviços na web nas empresas.

HTML

A **HTML** (*Hypertext Markup Language* – Linguagem de Marcação de Hipertexto) é uma linguagem de descrição de uma página para criar documentos de hipertexto ou hipermídia. A HTML insere códigos de controle em locais especificados em um documento que criam ligações (*hyperlinks*) com outras partes do documento ou com outros documentos em qualquer parte da World Wide Web. A HTML embute códigos de controle em texto ASCII de um documento para definir títulos, cabeçalhos, gráficos e componentes multimídia, além de *hyperlinks* dentro do documento.

Como já foi mencionado, diversos programas incluídos nas principais suítes de *software* automaticamente convertem documentos em formato HTML. Entre eles, estão navegadores web, programas de processamento de texto e planilha eletrônica, gerenciadores de banco de dados e pacotes de apresentação gráfica. Esses e outros programas especializados de *publicação na web*, como o Microsoft FrontPage, Lotus FastSite e Macromedia DreamWeaver, oferecem ampla gama de recursos para ajudar o usuário a desenhar e criar páginas multimídia para a web sem a programação HTML formal.

A **XML** (*eXtensible Markup Language* – Linguagem de Marcação Extensível) não é uma linguagem de descrição de formato de página web, como a HTML. Ao contrário, a XML descreve o conteúdo das páginas web (como de documentos corporativos destinados ao uso na web), aplicando identificadores ou *marcadores contextuais* nos dados dos documentos da web. Por exemplo, a página de uma agência de viagens, com nomes de linhas aéreas e horários de voos, teria marcadores XML ocultos como “nome da companhia aérea” e “horário de voo” para classificar cada horário dessa companhia na página. Além disso, os dados de estoque de produtos disponíveis em um *site* poderiam ter marcadores como “marca”, “preço” e “tamanho”. Os dados colocados dessa forma facilitam a busca, classificação e análise de informações na web.

XML

Por exemplo, um *software* de busca compatível com XML facilmente localizará o produto exato pesquisado se os dados do item na web tiverem marcadores de identificação XML. E um *site* web criado com XML determinaria mais facilmente as características da página utilizadas pelo cliente e os produtos pesquisados. Portanto, a XML promete facilitar muito e tornar mais eficientes os processos de comércio e negócio eletrônico, auxiliando o intercâmbio eletrônico automático de dados empresariais entre companhias e seus clientes, fornecedores e outros parceiros comerciais.

Como mencionado no início do capítulo, este livro foi completamente revisado e editado para a edição atual, por meio de um aplicativo baseado em XML chamado PowerXEditor, da Aptara. Vamos concentrar nossa atenção nesse aplicativo singular de XML destinado a gerar ganhos de eficiência na indústria editorial.

Aptara, Inc.: revolucionando a indústria de editoração por meio da XML

A indústria editorial sofreu uma convulsão na última década. A “cauda longa” de vendas de livros já existentes por meio de vendedores *on-line* como a Amazon e a melhoria nas tecnologias de *hardware* e *software* capazes de replicar a experiência de ler um livro ou revista indicam que as editoras estão imprimindo e vendendo cada vez menos livros inéditos. Como resultado, muitas dessas empresas estão se aventurando na editoração digital.

“Todas as editoras estão mudando do impresso para o digital”, anunciou Dev Ganesan, presidente e CEO da Aptara, empresa especializada em transformação de conteúdo. “É uma mudança e tanto. As empresas de *software* precisam desenvolver plataformas para criação de conteúdos que atendam às necessidades de cada cliente. Ao mesmo tempo, os clientes estão considerando a editoração em termos de manipulação de conteúdo em relação a autores, editores e funcionários da produção. E, além disso, estão tentando automatizar partes do processo de produção. E as empresas devem estar dispostos a comercializar produtos usando a mídia tradicional e a nova para atingir o maior público possível. Assim, há uma série de desafios, mas também um monte de oportunidades.”

O resultado de tudo isto é que os profissionais de aprendizagem podem agora fornecer conteúdo de forma mais flexível e com menor custo. Eles podem transformar conteúdos estáticos em dinâmicos ao tomarem um volume de conhecimento impresso, como um livro, e convertê-rem-no em um formato digital. Podem, em seguida, dividir esse conteúdo em pedaços menores e organizar essas pepitas de informação de acordo com as necessidades dos alunos. Além disso, podem fazer que o conteúdo seja publicado e distribuído com muito mais rapidez por meio das mídias digitais *on-line*. Isso é crucial em uma indústria que, como a de saúde, enfrenta rápidas mudanças decorrentes das inovações e regulamentações tecnológicas, disse outra fonte da Aptara.

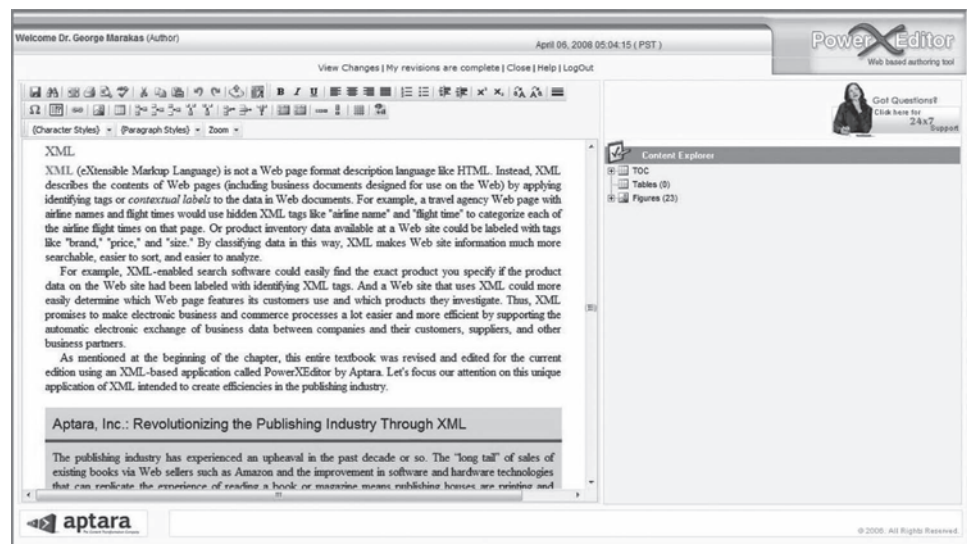
“Além da redução de custos, eles querem mudanças muito mais rapidamente”, afirmou. “O momento certo para comercialização torna-se fundamental porque há muita inovação acontecendo. Se eles não têm seus produtos de impressão prontos mais rapidamente, acabam ficando para trás”.

Um produto inovador da Aptara é o PowerXEditor (PXE). Aplicativo baseado em XML, o PXE permite que um editor faça upload de um layout de livro existente; edite ou revise todos os elementos do livro, incluindo a aparência e percepção dos textos, figuras, tabelas e outros elementos exclusivos para esse livro; e insira o livro em um programa de paginação que define a versão para impressão final. A questão importante é que tudo isso é feito em formato digital em vez do método anteriormente comum de dividir páginas e cortar e colar figuras e tabelas. Como o conteúdo do PXE é baseado em XML, o aplicativo pode ser acessado pela internet usando qualquer navegador web convencional. Isso significa que todos os que trabalham no livro podem ter acesso a vários capítulos e elementos onde quer que estejam. Adicione na administração do fluxo de trabalho os aspectos do PXE, e todas as fases de revisão, edição de texto e processos de verificação do livro podem ser manipuladas com facilidade.

A Figura 4.21 mostra uma típica tela do PXE. Você pode notar que se trata do processo de edição da página que você está lendo. A Figura 4.22 mostra o código XML da mesma página.

Fonte: Adaptado de Brian Summerfield. "Executive Briefings: Balancing Print and Digital Media". *Chief Learning Officer*, março de 2008. Disponível em <http://www.clomedia.com/includes/printcontent.php?aid=2133>

FIGURA 4.21 O PowerXEditor baseado em XML permite que todos os colaboradores do projeto de um livro tenham acesso aos elementos do livro por meio de um navegador web comum. Esta é uma tela do PXE da página que você está lendo agora.



Fonte: Cortesia da Aptara.

FIGURA 4.22 Esta é seção do código XML da página que você está lendo agora. Embora seja semelhante ao código-fonte em HTML, a XML é muito mais poderosa e complexa.

```
features to help you design and create multimedia Web pages without formal HTML programming.
</p>
<p class="Head3" id="head3_19">XML</p>
- <p class="Para_FL">
- <span class="Def_term">
  <b>XML</b>
</span>
(eXtensible Markup Language) is not a Web page format description language like HTML. Instead, XML
describes the contents of Web pages (including business documents designed for use on the Web) by
applying identifying tags or
<i>contextual labels</i>
to the data in Web documents. For example, a travel agency Web page with airline names and flight
times would use hidden XML tags like "airline name" and "flight time" to categorize each of the airline
flight times on that page. Or product inventory data available at a Web site could be labeled with tags
like "brand," "price," and "size." By classifying data in this way, XML makes Web site information much
more searchable, easier to sort, and easier to analyze.
</p>
<p class="Para_indented">For example, XML-enabled search software could easily find the exact product
you specify if the product data on the Web site had been labeled with identifying XML tags. And a Web
site that uses XML could more easily determine which Web page features its customers use and which
products they investigate. Thus, XML promises to make electronic business and commerce processes a
lot easier and more efficient by supporting the automatic electronic exchange of business data between
companies and their customers, suppliers, and other business partners.</p>
<p class="Head3" id="head3_20">Java and .NET</p>
- <p class="Para_FL">
- <span class="Def_term">
```

Fonte: Cortesia da Aptara.

Java é uma linguagem de programação orientada a objetos criada pela Sun Microsystems e que está revolucionando a programação de aplicações para a World Wide Web e intranets e extranets corporativas. A linguagem Java está relacionada com as linguagens de programação C++ e Objective C, mas é muito mais simples e mais segura, além de ser independente da plataforma de computação. A Java também se destina especificamente às aplicações de rede de tempo real, interativas e baseadas na web. As aplicações Java são compostas por pequenos programas de aplicação, denominados *applets*, e podem ser executadas por qualquer computador e em qualquer sistema operacional de qualquer local da rede.

Uma das principais razões da popularidade da linguagem Java é a facilidade de criação e distribuição dos *applets*, dos servidores da rede para os PCs-cliente e para os computadores da rede. Os *applets* podem consistir em pequenos programas de aplicação para fins especiais ou pequenos módulos de programas maiores de aplicações Java. Os programas Java também são independentes de plataforma – podem ser executados nos sistemas Windows, Unix e Macintosh sem nenhuma modificação.

A linguagem .NET da Microsoft é um conjunto de assistentes de programação para serviços na web, ou seja, um recurso para utilizar a web em lugar do próprio computador do usuário para acessar vários serviços (ver adiante). A linguagem .NET destina-se a oferecer a usuários individuais e empresariais interface interoperativa via web para aplicações e dispositivos do computador, além de tornar as atividades de computação cada vez mais orientadas a navegadores web. A plataforma .NET abrange servidores, serviços de componentes, como armazenamento de dados baseado na web; e *software* de dispositivos. Ela também inclui o Passport, serviço de verificação de identidade que requer preenchimento de formulários apenas uma única vez.

A plataforma .NET destina-se a permitir o funcionamento simultâneo de todos os dispositivos de computação e a atualização automática e sincronizada de todas as informações do usuário. Além disso, deverá oferecer serviços especiais de assinatura *on-line*. O serviço permitirá o acesso a produtos e serviços personalizados e a utilização destes de um ponto inicial central para gerenciamento de várias aplicações (por exemplo, *e-mail*) ou *software* (por exemplo, Office .NET). Para os desenvolvedores, a .NET oferece capacidade de criar e reutilizar módulos, aumentando, assim, a produtividade e reduzindo a quantidade de erros de programação.

A expectativa para o lançamento completo da .NET é de alguns anos, com lançamento periódico de produtos, como serviço de segurança pessoal e novas versões do Windows e Office, chegando separadamente ao mercado para implementar a estratégia da .NET. Hoje, já está disponível o ambiente de desenvolvimento Visual Studio .NET; além disso, o Windows XP possui algumas capacidades de .NET.

A versão mais recente da linguagem Java é a Java Enterprise Edition 5 (Java EE 5), que se tornou principal alternativa à plataforma de desenvolvimento de *software* .NET da Microsoft para muitas organizações que pretendem capitalizar o potencial comercial das aplicações e dos serviços baseados na web. A Figura 4.23 compara as vantagens e desvantagens da utilização da Java EE 5 e .NET no desenvolvimento de *software*.

Java e .NET

Serviços na web são componentes de *software* baseados na estrutura da web e nas tecnologias e nos padrões orientados a objetos para utilização da web para ligar eletronicamente as aplicações de diferentes usuários e diferentes plataformas de computação. Portanto, os serviços na web ligam funções empresariais básicas para compartilhar dados, em tempo real dentro das aplicações baseadas na web, por uma empresa com seus clientes, fornecedores e outros parceiros. Por exemplo, os serviços na web permitem à aplicação de compras de uma empresa verificar rapidamente o estoque de um fornecedor antes de emitir um grande pedido, enquanto a aplicação de vendas do fornecedor utiliza os serviços para verificar automaticamente a classificação de crédito da empresa em um serviço de informação de crédito antes de aprovar a compra. Desse modo, tanto entre as empresas como entre os profissionais de TI, a expressão *serviços na web* é comumente utilizada para descrever as funções de computação e negócios baseadas na web ou de serviços realizados pelas tecnologias e pelos padrões de *software* da web.

A Figura 4.22 mostra o funcionamento dos serviços na web e identifica algumas das tecnologias e dos padrões básicos envolvidos. A linguagem XML é uma das principais tecnologias que viabilizam o funcionamento das aplicações dos serviços na web entre diferentes plataformas de computação. São importantes também a **UDDI** (Universal Description and Discovery In-

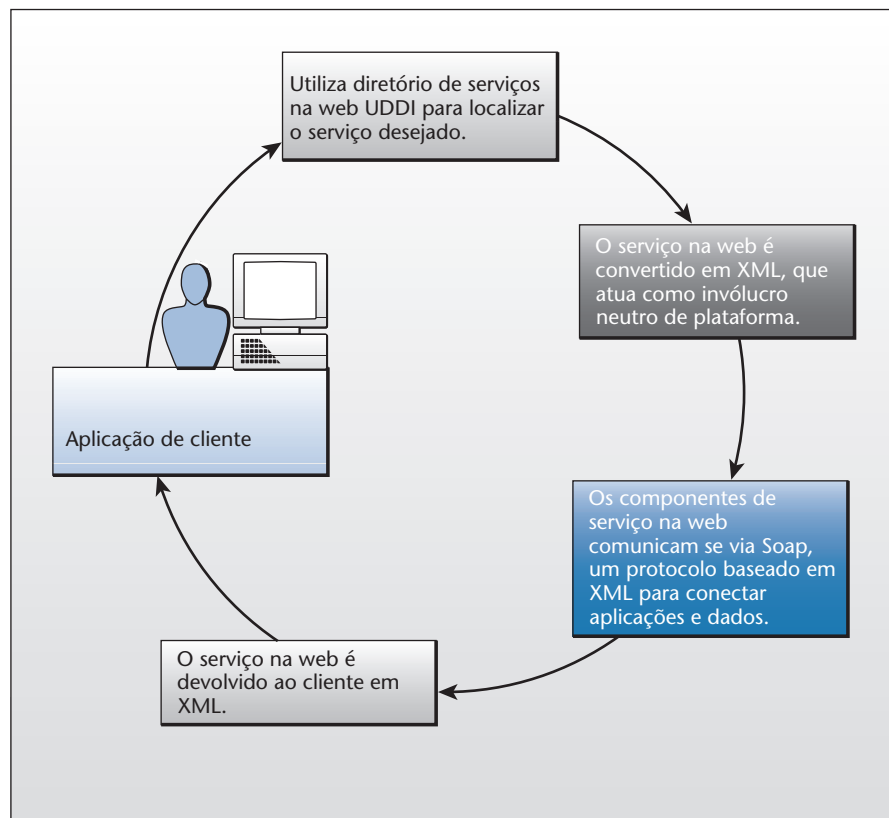
Serviços na web

| Java EE 5 | | .NET | |
|--|--|--|--|
| Prós | Contras | Prós | Contras |
| <ul style="list-style-type: none"> • Pode ser executada em qualquer sistema operacional e servidor de aplicativo (pode precisar de ajustes). • Comporta aplicações complexas de alto volume e de grandes transações. • Possui mais recursos empresariais para gerenciamento de sessões, solução de falhas, equilíbrio de carga e integração de aplicativo. • Conta com fornecedores experientes, como IBM, BEA, SAP e Oracle. • Oferece ampla gama de escolha de fornecedores de ferramentas e fornecedores de aplicativos. • Possui histórico comprovado. | <ul style="list-style-type: none"> • O ambiente de desenvolvimento do aplicativo é complexo. • As ferramentas às vezes são de difícil utilização. • A capacidade do ambiente Java Swing para criar interfaces gráficas do usuário tem limitações. • O custo de criação, desenvolvimento e controle de aplicativos pode se elevado. • Não contém suporte incorporado para padrões de serviço da web. • É difícil de utilizar em projetos de mercado de massa de alta rotatividade e de baixo custo. | <ul style="list-style-type: none"> • Ferramenta de fácil utilização aumentam a produtividade do programador. • Possui uma estrutura sólida para a criação de sofisticadas interfaces gráficas do usuário. • Oferece aos desenvolvedores opções para trabalhar com mais de vinte linguagens de programação. • Está firmemente integrada com o sistema operacional e o <i>software</i> de servidor corporativo da Microsoft. • Pode custar mais barato por causa, em parte, do servidor de aplicativo incorporado ao Windows, do gerenciamento unificado e da ferramentas mais baratas. • Contém suporte incorporado para padrões de serviço da web. | <ul style="list-style-type: none"> • A estrutura é executada apenas no Windows, restringindo as opções de fornecedor. • Os usuários das ferramentas e tecnologia anteriores enfrentam aprendizagem potencialmente intensa. • A nova infraestrutura de programas de execução não está madura. • Ainda há dúvida sobre a escalabilidade e capacidade de transação da plataforma do Windows. • As opções do ambiente de desenvolvimento integrado são limitadas. • A execução de aplicativos antigos no novo ambiente .NET pode ser complicada. |

Fonte: Carol Silva, ".NET versus Java", *Computerworld*, 20 de maio de 2002, p. 31.

FIGURA 4.23 Os benefícios e as limitações das plataformas de desenvolvimento de *software* Java Enterprise Edition 5 (Java EE 5) e Microsoft.NET.

FIGURA 4.24 As etapas básicas para a realização de um aplicativo de serviços da web.



Fonte: Adaptada de Bala Iyer, Jim Freedman, Mark Gaynor e George Wyner. "Web Services: Enabling Dynamic Business Networks", *Communications of the Association for Information Systems*, 11 2003, p. 543.

tegration), a lista de “páginas amarelas” de todos os serviços na web que permite localizá-los e utilizá-los, e o **Soap** (Simple Object Access Protocol), protocolo baseado em XML de especificações para conectar as aplicações com os dados necessários.

Os serviços da web são uma promissora tecnologia de *software* para automação do acesso aos dados e de funções de aplicação entre uma empresa e seus parceiros comerciais. As companhias estão realizando cada vez mais negócios pela web, portanto, se os serviços estão se tornando fundamentais para atender às necessidades de aplicações fáceis e eficientes de comércio e negócios eletrônicos. A flexibilidade e interoperabilidade dos serviços na web também serão essenciais para lidar com as rápidas mudanças nas relações entre companhia e parceiros comerciais, cada vez mais comuns no ambiente empresarial dinâmico e global de hoje.



Airbus: voando em SAP e serviços da web

A construtora europeia de aeronaves Airbus tem implementado um aplicativo de gerenciamento de viagens baseado em serviços da web da SAP como um primeiro passo para uma migração de toda empresa para uma arquitetura orientada a serviços (SOA). A fabricante de aviões está instalando o componente de gerenciamento de viagens do *software* ERP da SAP – o mySAP –, que usa a tecnologia SOA. “O novo sistema substitui um sistema desenvolvido na fábrica da empresa na França, um sistema baseado em Lotus, em suas operações na Espanha e versões anteriores do SAP nas instalações na Alemanha e no Reino Unido”, informa James Westgarth, gerente de contratos da tecnologia de viagens da Airbus.

“Gostamos da ideia de uma arquitetura aberta, que a tecnologia SOA permite”, diz Westgarth. “Gostamos de gerir tudo internamente e de escolher a melhor solução em cada classe.” Componentes adicionais, como reserva *on-line*, também poderão vir da SAP, se o fornecedor de *software* tiver um produto superior para essa aplicação”, conclui Westgarth.

A decisão de implantar um novo sistema de gestão de viagens baseado em serviços da web foi orientada, em grande parte, pela necessidade de reduzir os custos administrativos e melhorar os processos de negócios.

A Airbus tem um orçamento de viagem de 250 milhões de euros, que é usado para pagar mais de 180 mil viagens por ano. A empresa pretende reduzir custos, eliminando o processo atual de reembolso em papel, que consome tempo e trabalho, com um sistema que permite aos funcionários processar suas próprias despesas de viagem *on-line* a partir de seus computadores ou dispositivos móveis.

Um dos principais benefícios para os funcionários é que o período de reembolso será reduzido de dez para cerca de três dias. Além disso, o novo sistema permite à Airbus integrar novos prestadores de serviços com mais facilidade em suas operações, observa Westgarth. O fabricante terceirizou sua atividade de recuperação de impostos para empresas especializadas na tarefa. Com a ajuda de dinamizadores de *links* de aplicativos, Westgarth e sua equipe são capazes de conectar o seu sistema de gestão de viagens a outros aplicativos SAP da empresa, incluindo finanças e recursos humanos. A Airbus tem uma estratégia para, eventualmente, migrar para o ERP mySAP através de vários sistemas e países ao longo de vários anos.

“A empresa escolheu o gerenciamento de viagens como teste para o ERP mySAP”, diz Westgarth, que admite que houve alguns problemas com a implantação do aplicativo de gerenciamento de viagens: “Como somos a primeira grande empresa a implementar essa tecnologia, temos tido dificuldade em encontrar um número suficiente de pessoas qualificadas no mercado. E foi necessário algum trabalho para a integração da interface web em nosso portal.”

Entretanto, os funcionários da Airbus, segundo Westgarth, gostam da nova interface de usuário do aplicativo baseado na web, o log-in único e as orientações passo a passo. E a empresa gosta de flexibilidade. “Há cinco anos, ninguém falava sobre as empresas aéreas de baixo custo”, disse ele. “Precisamos nos adaptar ao mercado e às novas necessidades.”

Fonte: Adaptado de John Blau. “Airbus Flies on Web Services With SAP”. *IDG News Service/CIO Magazine*, 8 de junho de 2006.

Encerra aqui o trecho do livro disponibilizado para esta Unidade de Aprendizagem. Na Biblioteca Virtual da Instituição, você encontra a obra na íntegra.