

Desenvolvimento de Grandes Aplicações Web

Kyle Loudon

Introdução de Nate Koehley

Authorized Portuguese translation of the English edition of titled *Developing Large Web Applications*, First Edition, ISBN 978-0-596-80302-5 © 2010, Yahoo! Inc. This translation is published and sold by permission of O'Reilly Media, Inc., the owner of all rights to publish and sell the same.

Tradução em português autorizada da edição em inglês do título *Developing Large Web Applications*, First Edition, ISBN 978-0-596-80302-5 © 2010, Yahoo! Inc. Esta tradução é publicada e vendida com a permissão da O'Reilly Media, Inc., detentora de todos os direitos para publicação e venda desta obra.

© Novatec Editora Ltda. 2010.

Todos os direitos reservados e protegidos pela Lei 9610 de 19/02/1998. É proibida a reprodução desta obra, mesmo parcial, por qualquer processo, sem prévia autorização, por escrito, do autor e da Editora.

Editor: Rubens Prates
Tradução: Rafael Zanolli
Revisão gramatical: Patrícia Zagni
Editoração eletrônica: Camila Kuwabata e Carolina Kuwabata

ISBN: 978-85-7522-251-5

Histórico de impressões:

Setembro/2010 Primeira edição

Novatec Editora Ltda.
Rua Luís Antônio dos Santos 110
02460-000 – São Paulo, SP – Brasil
Tel.: +55 11 2959-6529
Fax: +55 11 2950-8869
E-mail: novatec@novatec.com.br
Site: www.novatec.com.br
Twitter: twitter.com/novateceditora
Facebook: facebook.com/novatec
LinkedIn: linkedin.com/in/novatec

**Dados Internacionais de Catalogação na Publicação (CIP)
(Câmara Brasileira do Livro, SP, Brasil)**

Loudon, Kyle
Desenvolvimento de grandes aplicações Web /
Kyle Loudon ; tradução Rafael Zanolli. -- São
Paulo : Novatec Editora ; Sebastopol, Calif :
O'Reilly, 2010.

Título original: *Developing large Web
applications.*
ISBN 978-85-7522-251-5

1. HTML (Hyper Text Markup Language) 2.
Internet (Rede de computadores) 3. Web sites I.
Título.

10-08752

CDD-004.67

Índices para catálogo sistemático:

1. Sites na Web : Criação : Internet : Redes de computadores : Processamento de dados 004.67
2. Web sites : Criação : Internet : Redes de computadores : Processamento de dados 004.67

ORGF20100903

Conforme as aplicações web se tornam cada vez maiores, de que forma os desenvolvedores lidam com sua crescente complexidade? Podemos fazê-lo voltando-nos a algumas das mesmas boas práticas utilizadas em outros tipos de desenvolvimento de software. Em termos gerais, essas práticas ainda não estão tão difundidas no desenvolvimento web, ou seja, no desenvolvimento de software que utiliza principalmente HTML, CSS, JavaScript e muitas outras linguagens de codificação lado-servidor (utilizaremos PHP para a codificação lado-servidor neste livro, mas os mesmo princípios podem ser aplicados às outras linguagens). Ademais, o caráter único das tecnologias existentes representa um desafio aos desenvolvedores que tentam aplicar boas práticas de modo coerente.

Um dos temas que você verá se repetir neste livro é a importância da extensão das práticas de desenvolvimento modular ao desenvolvimento web. Apresentaremos técnicas concretas e práticas para que se alcance a modularidade em grandes aplicações web. Nesse processo, exploraremos muitos dos aspectos mais sutis das linguagens HTML, CSS, JavaScript e PHP. Você verá que a maioria das técnicas é de aplicação relativamente simples e que nenhuma depende da utilização de frameworks específicos. É importante que se perceba que elas não impedem a utilização de frameworks diversos; ao contrário, tais técnicas criam um melhor ambiente no qual os muitos frameworks podem ser utilizados. Assim, veremos vários exemplos que utilizam a biblioteca JavaScript Yahoo! User Interface (YUI).

Logo de início é importante deixarmos claro por que as técnicas que exploraremos neste livro são especialmente úteis para os desenvolvedores web que trabalham em grandes aplicações. Começaremos examinando alguns dos fatores que contribuem para a complexidade de muitas dessas aplicações. Em seguida, exploraremos como a modularidade desempenha um importante papel no gerenciamento da complexidade. Por fim, examinaremos uma lista de princípios que deverão guiar-nos em nossas discussões ao longo do livro.

Gerenciamento da complexidade

Se você considerar quão diferente é a Internet de hoje daquela que existia dez anos atrás, ficará claro o grau de complexidade adquirido pelas aplicações web e a rapidez

com que as mudanças ocorreram. Com muita frequência, essa complexidade torna tais aplicações difíceis de serem mantidas e mais custosas durante sua existência. Vamos examinar alguns dos fatores que contribuem para a complexidade de muitas grandes aplicações da web. Geralmente, grandes aplicações seguem as seguintes características:

Disponibilidade contínua

A maioria das grandes aplicações deve rodar 24 horas por dia, sete dias por semana. Além disso, tempos de resposta devem ser rápidos em todos os momentos, mesmo nas horas de pico. Desenvolvedores web têm de escrever código que seja especialmente robusto.

Grande base de usuários

Grandes aplicações web normalmente apresentam volumes impressionantes de usuários. Isso requer o gerenciamento de um grande número de conexões simultâneas ou de várias camadas de caching. Os desenvolvedores muitas vezes têm de escrever códigos capazes de lidar com tais situações.

Entrega em partes

Enquanto muitos tipos de software são distribuídos como unidades completas, aplicações web têm muitas de suas partes entregues página a página, ou conexão a conexão, via Ajax. Como resultado, grandes aplicações web operam dentro de um ambiente compartilhado por um grande número de usuários.

Diversidade

Temos dificuldade em imaginar um negócio ou serviço que não tenha ao menos algum tipo de interface web. Por exemplo, existem aplicações financeiras, sites de venda de ingressos, aplicações que organizam volumes espantosos de dados (p. ex., mecanismos de busca), sistemas de mídia (p. ex., sites de notícias) e assim por diante. Desenvolvedores web devem estar prontos para escrever códigos que possam ser reutilizados em lugares inesperados.

Longevidade

As maiores aplicações web atuais, mesmo aquelas que existem há anos, estão apenas no começo de suas vidas. Desenvolvedores web têm de escrever código partindo do princípio de que ele será capaz de resistir a anos de alterações e manutenção.

Múltiplos ambientes

A web é um ambiente onde as mudanças acontecem muito rapidamente, repleta de velhos navegadores e outros dispositivos de difícil suporte. Usuários acessam grandes aplicações web a partir de todo tipo de ambiente e com telas

de tamanhos incrivelmente diferentes (incluindo dispositivos móveis). Desenvolvedores web devem ser capazes de escrever código que possa lidar com as inúmeras idiossincrasias que resultam desses fatos.

Atualizações em tempo real

Grandes aplicações web não são estáticas; na verdade, estão sempre em evolução, com diversas alterações sendo regularmente enviadas aos servidores. Os desenvolvedores web devem escrever códigos que possam se adaptar a essas partes móveis.

Com o passar do tempo, os desenvolvedores normalmente acabam tratando da complexidade presente nas grandes aplicações web por meio de correções e ajustes isolados, conforme suas aplicações atingem vários pontos de quebra (breaking points). Mas há um caminho melhor. Este livro irá mostrar-lhe como lidar com desafios, como os mencionados anteriormente, desde o início de seu código. A mitigação da complexidade desses desafios pode muitas vezes ser atribuída a um ou mais dos subprodutos da modularidade que examinaremos em breve. Por exemplo, mencionei que grandes aplicações web têm de estar sempre disponíveis. Da perspectiva do desenvolvedor web, trata-se de uma questão de confiabilidade e nesse aspecto a modularidade desempenha um papel vital.

Componentes modulares

Em uma grande aplicação web, para tratar da complexidade valendo-se de componentes modulares, ou módulos, você deve encapsular tudo de que o componente precisa dentro de partes pequenas e bem definidas da aplicação. Isso permite a divisão de uma grande aplicação em partes mais facilmente manuseáveis que podem ser construídas com um foco específico e reutilizadas sempre que necessário. Além disso, você deve esconder (ou abstrair) os detalhes da implementação de cada módulo e tomar medidas com o intuito de garantir que cada um deles possa operar de modo mais ou menos independente dos outros. Até mesmo as aplicações web relativamente pequenas de hoje podem ser os Googles, Yahoo!s e Amazons de amanhã. Ao criar suas aplicações de modo modular desde o princípio, você oferecerá uma fundação sólida capaz de suportar o crescimento do futuro.

A modularidade parece algo simples, mas sua realização de modo coeso pode ser particularmente difícil quando passamos por todos os scripts HTML, CSS, JavaScript e lado-servidor que os desenvolvedores terão de escrever para as grandes aplicações web. Vamos agora examinar mais detalhadamente o conceito de modularidade e algumas das ideias básicas que podem ajudá-lo a atingi-lo.

Atingindo a modularidade

A modularidade de grandes aplicações web pode ser atingida, assim como em outros tipos de software, por meio do *encapsulamento*, da *abstração* e da manutenção do maior número possível de acoplamentos fracos entre os módulos de uma aplicação.

O encapsulamento é o processo de agrupamento de tudo que é necessário para um módulo dentro de uma única unidade coesa. Módulos para aplicações web devem encapsular todo o HTML, CSS, JavaScript e PHP que for necessário para sua existência. O capítulo 7 mostra como isso pode ser feito por meio de PHP orientado a objeto. Também teremos de aplicar técnicas no HTML, no CSS e no JavaScript em si para que um módulo aceite isso. Tais técnicas serão apresentadas nos capítulos 3, 4 e 5, respectivamente.

Abstração

A abstração é o processo de ocultação de detalhes que não devem ser observados quando se trabalha com um módulo de um ponto externo à sua implementação. A definição de uma boa interface é a chave para a abstração. Nesse ponto, aplicações web apresentam desafios especiais, já que a linguagem HTML não foi projetada para esconder informações da forma como muitas das outras linguagens o fazem e também porque o CSS avança para além das fronteiras dos módulos em suas *cascatas*, devendo ser administrado com muito rigor. O capítulo 7 mostra como PHP orientado a objetos pode ser utilizado para definir interfaces que abstraem os detalhes do trabalho com seções de HTML e CSS. Gerenciadores de dados, apresentados no capítulo 6, são bons exemplos de interfaces que conseguem abstrair o trabalho com dados dinâmicos administrados pelo backend¹.

Acoplamentos fracos

Um acoplamento fraco entre módulos significa que um depende do outro o mínimo possível e apenas em formas definidas com clareza. No capítulo 2, você verá que é relativamente simples criarmos um gráfico das dependências entre os objetos em um sistema orientado a objetos. Quando a orientação a objetos é utilizada para implementar módulos em uma grande aplicação web, notamos mais facilmente como as mudanças em uma parte do sistema podem afetar as outras. As técnicas apresentadas para HTML e CSS modulares nos capítulos 3 e 4 também promovem acoplamentos fracos.

1 N.T.: No design de software, tem-se tanto um frontend quanto um backend para o banco de dados, que são, respectivamente, o usuário e os componentes de processamento de dados. A separação de sistemas de software em front e backends simplifica o desenvolvimento e separa a manutenção (fonte: Wikipédia).

Benefícios da modularidade

Uma vez que você tenha criado módulos que aproveitam do encapsulamento, da abstração e dos acoplamentos fracos, terá como resultado os benefícios da *capacidade de reutilização*, da *sustentabilidade* e da *confiabilidade* em toda a sua aplicação web. Quando um módulo é reutilizável, fica claro como devemos proceder para fazê-lo operar e realizar novas tarefas. Você pode movê-lo, entre vários pontos da aplicação, certo de que tudo de que ele necessita para funcionar corretamente o acompanhará e de que você não encontrará consequências inesperadas quando reutilizá-lo. Quando um módulo é sustentável, fica clara qual área da aplicação deve ser alterada para atingir determinados atributos e como as alterações devem ser feitas. Módulos de fácil manutenção são mais confiáveis por reduzirem o risco de efeitos colaterais quando se realizam as mudanças.

Dez princípios para grandes aplicações web

Pensando na modularidade, utilizaremos a lista de princípios desta seção para guiarnos em nossas discussões nos capítulos que virão a seguir. Um princípio é uma ideia ou crença que aceitamos como verdadeira, ou seja, trata-se de uma asserção. Uma rápida leitura dessa lista poderá fornecer-lhe um bom entendimento do que queremos alcançar. Posteriormente, você verá essa lista como um conciso lembrete dos conceitos que discutimos. Cada princípio é examinado no capítulo com o número correspondente. Por exemplo, o princípio 4 descreve a utilização robusta da linguagem CSS e o capítulo 4 mostra como aplicar esse princípio quando estiver implementando suas CSS.

Princípio 1

Grandes aplicações web são construídas a partir de componentes modulares altamente reutilizáveis, sustentáveis e confiáveis.

Princípio 2

A utilização da orientação a objetos em JavaScript e nas linguagens de codificação lado-servidor acrescenta a capacidade de reutilização, a sustentabilidade e a confiabilidade das grandes aplicações web ao promover a modularidade.

Princípio 3

HTML em larga escala é semântico, destituído de elementos de apresentação, exceto por aqueles inerentes à arquitetura de informação, e conectável a uma grande variedade de contextos na forma de seções facilmente identificáveis.

Princípio 4

CSS em larga escala forma uma camada de apresentação que é separada da arquitetura de informação, aplicada de modo modular e livre de efeitos colaterais que surgem quando reutilizamos os módulos em vários contextos.

Princípio 5

JavaScript em larga escala forma uma camada de comportamento aplicada de modo modular e orientado a objetos, que impede efeitos colaterais quando os módulos são reutilizados em vários contextos.

Princípio 6

Dados dinâmicos transmitidos entre a interface do usuário e o backend são controlados por meio de uma interface de dados claramente definida. Páginas definem um único ponto onde os dados são carregados e um único ponto onde são salvos.

Princípio 7

Páginas são construídas a partir de módulos altamente reutilizáveis que encapsulam tudo que é necessário (p. ex., HTML, CSS, JavaScript e qualquer outro elemento necessário), tornando cada módulo uma unidade coesa e de funcionamento independente que pode ser utilizada em vários contextos e páginas.

Princípio 8

Ajax em larga escala deve ser portátil e modular, mantendo uma clara separação entre as transferências de dados e as atualizações à apresentação. A transferência de dados entre o navegador e o servidor é administrada por meio de uma interface claramente definida.

Princípio 9

HTML, JavaScript, CSS e PHP em larga escala oferecem uma firme fundação sobre a qual podemos construir grandes aplicações web de bom desempenho. Além disso, produzem um ambiente adequado à captura de métricas referentes ao site e à realização de testes.

Princípio 10

A organização dos arquivos no servidor para uma grande aplicação web reflete a arquitetura da aplicação em si, incluindo os escopos claramente demarcados nos quais cada arquivo será utilizado.