

Padrões de Projeto e Frameworks Web

APRESENTAR O TEMA PADRÕES DE PROJETOS E FRAMEWORKS PARA O DESENVOLVIMENTO DE PÁGINAS DINÂMICAS.

AUTOR(A): PROF. EDSON MELO DE SOUZA

Introdução

A criação de páginas Web dinâmicas vem cada vez mais exigindo dos desenvolvedores conhecimento nas novas tecnologias e também a adaptação em relação ao tempo de desenvolvimento. Atualmente a construção de interfaces Web tem ocorrido de forma frenética, ou seja, as empresas necessitam estar alinhadas com o que há de mais novo no mercado, pois precisam acompanhar a concorrência ou, então, estar na vanguarda da tecnologia, sendo uma das primeiras a utilizarem determinada tecnologia.

Seguindo este raciocínio, o desenvolvimento de páginas Web pode ser facilitado, com mais qualidade, mais eficiência e mais rapidamente se forem usados *frameworks* e padrões de projeto.

Este tópico aborda os conceitos de Padrões de Projeto e Frameworks que podem facilitar a criação de sites e páginas com mais facilidade e qualidade.

1. Padrões de Projeto (*Design Patterns*)

Um Padrão de Projeto pode ser definido como um conjunto de soluções que podem ser aplicadas na resolução de um problema comum conhecido, ou seja, resolver um problema que já foi enfrentado por muitos em situações semelhantes. Pode ser definido também como uma técnica que aborda o reaproveitamento ou reuso de *software*, oferecendo muitos benefícios práticos no momento do desenvolvimento.

Inicialmente os padrões de projeto foram concebidos para o desenvolvimento de aplicações orientadas a objetos, os quais estão diretamente relacionadas com a Engenharia de Software. Entretanto, devido ao crescimento dos tipos de sistemas a serem criados possuírem características semelhantes, a técnica passou a ser adotada também na construção de bancos de dados, interfaces Web, sistemas móveis, entre outros.

Um padrão de projeto é constituído de uma documentação que orienta como um determinado problema pode ser resolvido e, segundo o conceito criado por Alexander (1979), “Cada padrão é um relacionamento entre um certo contexto, um certo sistema de forças que ocorrem repetidamente nesse contexto (problema), e uma certa configuração (solução) que permite que essas forças se resolvam.”. Então um padrão é determinado pelo seu nome e pelos três pilares de forma que:

- O Contexto se refere às situações onde há repetição.
- O Problema se refere ao conjunto de forças que ocorrem dentro de um Contexto, ou seja, dos objetivos e das limitações impostas pelo sistema.
- A Solução é a “receita” ou descrição dos procedimentos que devem ser adotados para a resolução do problema.

A grande vantagem em utilizar padrões de projeto está no fato de que são soluções apresentadas e documentadas que foram utilizadas e testadas por muitos programadores ao redor do mundo, trazendo confiabilidade na sua eficácia (Design Patterns, 2012).

Para o desenvolvimento de interfaces Web, existem diversos padrões de projeto que são mantidos por grandes empresas como o Yahoo!, Welie e UI-patterns.

Os padrões de projeto aplicados no design de interface visam ajudar na construção de soluções resolvendo problemas conhecidos, fornecendo uma linguagem comum entre os desenvolvedores e orientando as boas práticas para criação de interfaces.

Esses padrões estão divididos em categorias, onde cada qual trata dos assuntos que estão dentro daquele domínio. Ainda, dentro das categorias, existem subcategorias. As categorias se dividem em:

- Entrada de Dados (*Getting Input*) - aborda os conceitos de como deve ser realizada a coleta de dados em um site, orientando para a criação dos formulários que tragam ferramentas como calendário para seleção de datas, sistema de Captcha para segurança contra Spam, teclas de atalho, etc.
- Tratamento de Dados (*Dealing with Data*) - está relacionado com a disposição tabular dos dados, orientando para construção correta de tabelas, formatação de dados, autocompletar, imagens, entre outros.
- Navegação (*Navigation*) - aborda as implementações acerca da construção de menus, atalhos inteligentes, divisão em abas, paginação, Breadcrumbs, etc.
- Social - orienta para a interação do usuário no que diz respeito de depoimentos, chat, faq's, envio de convites (divulgação), entre outros.
- Guia de Uso e Avisos (*Onboarding*) - esta categoria visa orientar para construção de tutoriais, inclusão de dicas em links, notificação de ocorrências, além de fornecimento de sistema de registro com opções de remover, entre outros.
- Diversos (*Miscellaneous*) - essa categoria dispõe de orientações para implementação de páginas de produtos e/ou serviços, e também para elementos que não estiverem inseridos nas demais categorias.

Uma outra classe de padrões, mais voltada para o design persuasivo (*Persuasive Design*) também está disponível para aplicação em projetos e tem como objetivo criar um engajamento emocional através de experiências interativas e cognitivas que influenciam as ações das pessoas. Esse tipo de padrão é muito utilizado em jogos digitais e em mídias digitais com foco em vendas de impacto.

SAIBA MAIS!

UI-patterns - <http://ui-patterns.com/patterns> (<http://ui-patterns.com/patterns>)

2. Frameworks

Um *framework* nada mais é do que uma solução de *software* que traz diversas funcionalidades prontas, com o objetivo de aliviar a sobrecarga de desenvolvimento de uma aplicação.

Para o *design* de interfaces, existe uma problemática constante que é o relacionamento entre os desenvolvedores Front-End e Back-End, no que diz respeito a integração de soluções.

O profissionais de Front-End estão constantemente enfrentando situações onde há a necessidade de adaptar *layouts* (estruturas de telas) em ambientes diversos como: navegadores de diferentes fabricantes, os quais não são 100 % compatíveis; dispositivos móveis em suas mais variadas versões, entre outros. Considerando que atualmente os novos sites já nascem dinâmicos e há uma grande quantidade de outros sendo migrados, a utilização de um *framework* se faz quase que indispensável, uma vez que o tempo e a qualidade final devem ser rápido e alta, respectivamente.

Um *framework* Web fornece elementos ou componentes baseados em tecnologias conjuntas como o JavaScript e a jQuery para trabalhar com os objetos HTML e CSS e, desta forma, acelerar a criação de páginas por meio da utilização de componentes prontos.

Existem *frameworks* para cada tipo de camada, ou seja, *Front-End* e *Back-End*. No lado do cliente (*Front-End*) os mais conhecidos atualmente são o Bootstrap e Materialize. Esse *frameworks* oferecem diversos recursos para um rápido desenvolvimento, trazendo eficiência e estética muito satisfatórias.

2.1 Bootstrap

O Bootstrap se apresenta como um framework CSS muito poderoso, versátil e de fácil aprendizagem, colaborando para o desenvolvimento em equipe, já que há uma padronização dos elementos, além destes serem esteticamente atraentes.

Este *framework* foi desenvolvido pelo time de desenvolvedores do Twitter (www.twitter.com (<http://www.twitter.com/>)) para facilitar a manutenção dos códigos CSS, na criação de menus desdobráveis, animações, barras laterais, *banners*, entre outros elementos. Uma de suas características

marcantes é a responsividade, ou seja, a capacidade de uma página ou site ajustar-se automaticamente a um dispositivo e resoluções diferenciadas de telas. Fazendo uso de Javascript e jQuery, as aplicações ficam com uma interface muito mais eficiente, acessível e atrativa, além da rapidez no desenvolvimento. A seguir é mostrada uma página criada com o *framework* Bootstrap, utilizando alguns componentes mais comuns como caixas de texto, barra de navegação e painéis.

Exemplo Bootstrap

Link

Link 1

Lista ▾

Email

senha

Realizar login

Olá mundo!

Esta é uma página escrita com o *framework* Bootstrap. Apesar de simples, mostra como é fácil criar páginas com esta ferramenta. A divisão da página foi realizada usando apenas divs e as classes disponíveis.

Leia mais »

Título 1

Esta é uma coluna criada com o div utilizando a classe "col-md-4". Aqui podem ser inseridos mais textos, figuras ou vídeos.

Mais detalhes »

Título 2

Veja como é fácil fazer a divisão em colunas, basta repetir o bloco "col-md-4" para ter um novo espaço.

Mais detalhes »

Heading

Esta é a última coluna. Basta usar a imaginação para criar os blocos e utilizar.

Mais detalhes »

© Linguagem HTML, CSS e o Framework Bootstrap

Legenda: INTERFACE CRIADA COM O BOOTSTRAP

SAIBA MAIS!

Bootstrap Oficial - <http://getbootstrap.com/> (<http://getbootstrap.com/>)



Starter Template

A modern responsive front-end framework based on Material Design

GET STARTED



Speeds up development

We did most of the heavy lifting for you to provide a default stylings that incorporate our custom components. Additionally, we refined animations and transitions to provide a smoother experience for developers.



User Experience Focused

By utilizing elements and principles of Material Design, we were able to create a framework that incorporates components and animations that provide more feedback to users. Additionally, a single underlying responsive system across all platforms allow for a more unified user experience.



Easy to work with

We have provided detailed documentation as well as specific code examples to help new users get started. We are also always open to feedback and can answer any questions a user may have about Materialize.

Company Bio

We are a team of college students working on this project like it's our full time job. Any amount would help support and continue development on this project and is greatly appreciated.

Settings

Link 1
Link 2
Link 3
Link 4

Connect

Link 1
Link 2
Link 3
Link 4

Made by Materialize

2.2 Materialize

Assim como o Bootstrap, o Materialize é um *framework* que visa facilitar e agilizar o desenvolvimento de interfaces Web, pois, além da curva de aprendizagem ser baixa, possui uma muitas vantagens que permitem realizar todo um trabalho sem a necessidade de outros *frameworks* complementares. Tem como base o *Material Design*, desenvolvido pela Google, o que é uma das principais vantagens deste *framework*, uma vez que a Google sempre procura utilizar o que há sempre de melhor na tecnologia.

Este *framework* traz muitos recursos como a utilização da JQuery com os *plugins* Slider, Lightbox, Captions, Modais, Transitions, Waves entre muitos outros. Em termos estruturais, o Materialize trabalha também com sistemas de colunas, como o Bootstrap, o que facilita muito a migração entre os *frameworks*.

O Materialize apresenta alguns recursos que são diferenciais em relação ao Bootstrap, como os botões, o sistema de *modal*, os *cards*, as *collections* e o menu lateral que um grande destaque, pois está muito relacionado às aplicações móveis.

A seguir é mostrada uma página criada com o *framework* Materialize, utilizando alguns de seus componentes mais comuns.

Além dos *frameworks* Bootstrap e Materialize, que são baseados na construção do HTML, outros também se destacam tendo como base o JavaScript, ou seja, são ferramentas que ajudam a otimizar o desenvolvimento de interfaces dinâmicas. A seguir são apresentados os principais *frameworks*.

2.3 jQuery

A jQuery é uma biblioteca de funções JavaScript que trabalha diretamente com o HTML, manipulando componentes na página, a fim de proporcionar mais dinamismo. Ela é composta por funções que facilitam o desenvolvimento e a aplicação de recursos, sem a necessidade de muita programação para realizar as tarefas. Seu principal objetivo é otimizar, de maneira fácil, a interpretação dos *scripts* utilizados em uma página.

2.4 jQuery UI

É um complemento que permite implementar diversos diversos para gerar interfaces de usuario em páginas web, além de outras funcionalidades básicas para criar aplicações web enriquecidas (Criar Web, 2017).

2.5 JavaScriptMVC

JavaScriptMVC (JMVC) é um *framework* JavaScript que cria aplicativos de fácil manutenção, livres de erros e leves, com bastante rapidez. Este *framework* reúne as melhores bibliotecas e ferramentas que são garantidas para trabalharem juntas, sendo suportado por todos os navegadores compatíveis com jQuery.

2.6 Prototype JS

O prototype JS leva a complexidade de todo desenvolvimento web para fora do lado do cliente. "Foi criado para solucionar alguns problemas do mundo real e melhora a interface do Ajax e do DOM" (Linha de Código, 2017).

2.7 Script.aculo.us

Este *framework* permite criar efeitos visuais em Javascript com muita facilidade. "O script.aculo.us é uma biblioteca que permite a criação de efeitos de forma simples, sem precisar ser um desenvolvedor com conhecimentos avançados em Javascript (Linha de Código, 2017).

2.8 Kendo UI

Segundo Linha de Código (2017), "O Kendo UI além de ser um framework javascript, também é HTML5. Com ele é possível criar sites modernos e robustos, além de aplicativos *mobile*."

Como a tecnologia avança rapidamente, outros *frameworks* também podem ser encontrados na internet, oferecendo vantagens e facilidades.

SAIBA MAIS!

Materialize CSS - <http://materializecss.com/about.html> (<http://materializecss.com/about.html>)

Tableless - <https://tableless.com.br/material-design-com-materialize/>
(<https://tableless.com.br/material-design-com-materialize/>)

jQuery - <https://jquery.com/> (<https://jquery.com/>)

jQuery UI - <http://jqueryui.com/> (<http://jqueryui.com/>)

JavaScriptMVC - <http://www.javascriptmvc.com/> (<http://www.javascriptmvc.com/>)

Prototype JS - <http://prototypejs.org/> (<http://prototypejs.org/>)

Script.aculo.us - <http://script.aculo.us/> (<http://script.aculo.us/>)

Resumo

Neste tópico foram introduzidos os conceitos de Padrões de Projeto (*Design Patterns*), *Frameworks*, além de apresentar os dois *frameworks* mais conhecidos para desenvolvimento de interfaces dinâmicas.

Conclusão

Utilizar padrões de projeto e *frameworks* no desenvolvimento de interfaces dinâmicas, faz com que haja um aumento na produção e na qualidade dos projetos concebidos, além de manter harmonioso o desenvolvimento com as tecnologias mais modernas.

ATIVIDADE FINAL

Qual a definição de um Padrão de Projeto?

- A. É um conjunto de soluções que podem ser aplicadas na resolução de um problema comum conhecido.
- B. É um conjunto de *frameworks* para facilitar o desenvolvimento de uma interface.
- C. É uma solução de *software* que traz diversas funcionalidades prontas.
- D. É uma forma de acelerar o processo de desenvolvimento de uma interface dinâmica.

Os frameworks são divididos em?

- A. *Front-End* e *Back-End*.
- B. *Front-End* e *Back-Front*.
- C. *End-Front* e *Back-End*.
- D. *Front-End* e *End-Back*.

Quais são os frameworks mais conhecidos atualmente para o desenvolvimento de interfaces Web?

- A. Bootstrap e Materialize.
- B. Bootstrap e Material Design.
- C. Twitter e Bootstrap.
- D. Twitter e Google Material.

REFERÊNCIA

ALEXANDER, Christopher. The timeless way of building. New York: Oxford University Press, 1979.

BIBEAULT, B.; KATZ, Y. *jQuery in Action*. 2 ed. Greenwich, CT: Manning Publications Co., 2008. p. 475. ISBN 1-935182-32-5.

Criar Web. Primeiros passos com jQuery UI. Disponível em: <www.criarweb.com/artigos/primeiros-passos-jquerui.html>. Acesso em: 4 mai. 2017.

DE SOUZA, Edson Melo. *Desenvolvimento de um Sistema de Apoio à Decisão e Operacional para a Otimização dos Parâmetros de Corte em Usinagem (COPPISYS)*. 2013. 149 f. Dissertação (Mestrado em Engenharia de Produção) ¿ Programa de Mestrado em Engenharia de Produção, Universidade Nove de Julho, São Paulo, 2013.

MARTINS, Cláudio. disponível em: <<http://www.facom.ufu.br/~bacala/DAW/artigo2012-javamagazine107-PadreesProjetoWeb.pdf>> (<http://www.facom.ufu.br/~bacala/DAW/artigo2012-javamagazine107-PadreesProjetoWeb.pdf>)>. Acesso em: 8 de abr. 2017.

O que são Design Patterns. 2012. Disponível em: <<http://www.princiweb.com.br/blog/programacao/design-patterns/o-que-sao-design-patterns.html>> (<http://www.princiweb.com.br/blog/programacao/design-patterns/o-que-sao-design-patterns.html>)>. Acesso em: 7 de abr. 2017.

Principais Frameworks de Javascript. 2017. Disponível em: <<http://www.linhadecodigo.com.br/artigo/3637/principais-frameworks-de-javascript.aspx>> (<http://www.linhadecodigo.com.br/artigo/3637/principais-frameworks-de-javascript.aspx>)>. Acesso em: 5 de mai. 2017.

RICHARDS, G.; LEBRESNE, S.; BURG, B.; VITEK, J. *An analysis of the dynamic behavior of JavaScript programs. 2010 ACM SIGPLAN conference on Programming language design and implementation PLDI 10*, v. 45, n. 6, p. 1, 2010.

TUTORIAL WEB DESIGN. Tutorial Bootstrap. Disponível em: < <http://www.tutorialwebdesign.com.br> (<http://www.tutorialwebdesign.com.br/>)>. Acesso em 7 de mai. 2017.

WAHEED, Adil *et al.* Discovery of design patterns variants for quality software development. In: Intelligent Systems Engineering (ICISE), 2016 International Conference on. IEEE, 2016. p. 185-191.

W3C. HTML, The Web's Core Language. Disponível em: <<http://www.w3.org/html/> (<http://www.w3.org/html/>)>. Acesso em: 7 de mai. 2017.

