

# Frameworks para Aplicativos Web

APRESENTAR OS PRINCIPAIS CARACTERÍSTICAS DOS FRAMEWORKS USADOS PARA O DESENVOLVIMENTO DE INTERFACES WEB.

AUTOR(A): PROF. PAULO RICARDO BATISTA MESQUITA

Retornando um pouco aos anos 1960, militares norte-americanos tiveram a idéia de criar uma base de informações descentralizada, pois no caso de um ataque nuclear, as informações não seriam perdidas, continuando disponíveis para qualquer base militar. Dessa idéia, surgiu a ARPANET, que foi construída por um consórcio de universidades, e que com o tempo, evoluiu para o que conhecemos hoje como Internet.

No início, era necessário somente apresentar as informações. Para que a apresentação fosse organizada, foi desenvolvida a HTML (Hyper Text Markup Language), uma linguagem de marcação que ao ser inserida no meio de um texto, faria com que qualquer browser mostrasse essa informação sempre da mesma forma.

Com o início do uso comercial da Internet, durante os anos 1990, quando o grande público teve acesso à rede, foi percebido o potencial para usá-la como plataforma de negócios. Para que isso fosse possível, foi necessário permitir que os usuários pudessem interagir com as páginas HTML. O próprio HTML teve que ser modificado para permitir isso. Como resultado surgiram o CSS (Cascade Style Sheets) e o JavaScript. O CSS é um padrão para o desenvolvimento de lay-outs de página web, e se for bem usado, pode mudar totalmente o visual de um sistema web e simular vários efeitos gráficos, como pode ser observado no site CSS Zen Garden (<http://www.csszengarden.com> (<http://www.csszengarden.com>)).

JavaScript foi a linguagem de programação definida pela W3C para controlar dinamicamente o conteúdo de uma página web a partir da interação com o usuário. Mas antes dela, era possível tratar dados enviados por formulários através de Applets escritos em Java, que logo foram substituídos por Servlets, e por CGI, que eram rotinas executadas num servidor web, escritos em C ou C++.

Até o início dos anos 2000, um site poderia ser de conteúdo estático (orientado à apresentação). Esse tipo de site costumava ser desenvolvido para apresentar as informações de uma empresa, ou similares. O outro tipo, eram os sites orientados a serviços. Nesses sites, o conteúdo apresentado variava de acordo com a forma que o usuário interagía com o site. No início, esse tipo de aplicação fazia parte de sites orientados à apresentação. Hoje em dia, são bem poucos os exemplos de sites orientados à apresentação. Redes sociais, portais de notícias, comércio eletrônico, etc? são vários dos exemplos para sites orientados a serviços.

Vários dos sistemas empresariais atuais usam o conceito de sites orientados a serviços, substituindo os aplicativos locais, desenvolvidos na estrutura cliente-servidor. O motivo para isso é que sob o ponto de vista operacional, é mais barato e simples ampliar o número de usuários do sistema, fornecendo apenas um browser e um ponto de rede, do que instalar localmente vários módulos do aplicativo, e somente depois disso, distribuir os computadores para os vários usuários. Além disso, ainda há a questão da manutenção do sistema, que passa a ser centralizada.

Em todos esses casos, a interface web é desenvolvida com base em documentos HTML, que têm adicionado a eles, arquivos CSS e JavaScript para melhorar a interação com o usuário. Assim, elas formam o núcleo das tecnologias web para exibir o conteúdo da página web.

A grande questão é quando é necessário controlar o conteúdo HTML a ser exibido pela interface web. Nesses casos, a plataforma Java EE disponibiliza um subconjunto de tecnologias denominado Java Web, formado por JSP, Servlets e JavaBeans. Há várias formas de usar essas tecnologias, desde escrever toda a regra de negócios em arquivos JSP, ou distribuí-los em classes do tipo JavaBeans e usar Servlets para conectar as páginas JSP e/ou HTML aos JavaBeans. Algumas formas são mais simples de serem desenvolvidas e mais complexas de serem atualizadas, ou mais complexas de serem desenvolvidas, e mais difíceis de serem atualizadas. A escolha é do desenvolvedor.

Em se tratando dos sistemas de alta complexidade, como os aplicativos corporativos mencionados anteriormente, Java WEB foi projetado para ser usado da seguinte maneira:

- JSP (Java Server Pages): misturado ao código HTML, em arquivos de extensão JSP, é usado com a função de controlar o conteúdo HTML a ser apresentado ao usuário. Ele pode ter seu layout aprimorado com o uso de CSS e JavaScript, como se fosse um arquivo HTML puro;
- Servlet: são classes que tratam requisições e geram respostas de aplicações web, elas podem ser usadas para controlar a navegação entre as várias páginas do site, e controlam o uso das classes JavaBeans;
- JavaBeans: são classes que executam as regras de negócio e as tarefas de infra-estrutura, como acesso a bases de dados;

Para funcionar corretamente, as aplicações JavaWeb necessitam de um arquivo chamado Descritor de Deployment (web.xml), que configura como os Servlets são mapeados e usados, e como o Servlet Container (ambiente de execução de arquivos JSP e Servlets) deve operar.

Voltando à idéia original da ARPANET: um sistema de armazenamento de informações, descentralizado, e que se for perdido um servidor, outro assume seu lugar. Ao analisar essa idéia, é possível encontrar algumas semelhanças com a idéia da computação em nuvem: acessar qualquer serviço disponível na rede, usando qualquer tipo de dispositivo que se conecte à rede.

Talvez os aplicativos do Google, como o Docs, Drive, Gmail, etc? sejam os exemplos mais comuns de computação em nuvem, e usam interfaces web. Aplicando o que já foi visto até agora, utilizaríamos Java Web para implementar a interface de usuário.

Na computação em nuvem, podem ser usados aplicativos executados em outros servidores. Para acessar esses aplicativos, podem ser usados tanto Servlets quanto JavaBeans. A comunicação entre servidores remotos exige algumas tecnologias adicionais de Java EE, como:

- JNDI: usado para endereçar o servidor, usuários ou os serviços que são executados remotamente;
- RMI: usado para estabelecer a comunicação entre objetos remotos;
- JTA: usado para controlar remotamente a execução de transações;

Todos esses sistemas funcionam através da sincronização de dados entre o cliente e o servidor, que normalmente assume duas formas de sincronizar os dados. Na primeira, os componentes de software que são executados do lado do servidor enviam os dados para os componentes executados do lado do cliente. Esse mecanismo é denominado PUSH, e há alguns frameworks que implementam esse tipo de operação, entre os mais conhecidos estão o Struts, Ruby on Rails e o Spring MVC. No segundo modo, os componentes de software executados do lado do cliente solicitam que vários componentes de software que funcionem como controles, lhe enviem as informações que serão mostradas ao usuário. Entre os frameworks mais conhecidos para esse mecanismo estão o JavaServer Faces (JSF), Struts2, ASP.Net Web Forms e Laravel.

Esses frameworks compreendem todos os componentes de software que compõem o aplicativo. Isso significa que eles são direcionados para facilitar o desenvolvimento do aplicativo como um todo, e não se preocupam com questões específicas, como a interação com o usuário, ou o gerenciamento dos dados usados pelo aplicativo. Esses frameworks podem ser complementados com os frameworks CSS, como o Bootstrap, que provê um mecanismo para o desenvolvimento de sites responsivos, ou a API de Persistência Java, que implementa o gerenciamento da base de dados.

## Frameworks para Design Responsivo

Uma necessidade que é bastante comum aos dias atuais, é a necessidade de uma interface web que se adapte automaticamente à resolução que o browser que está renderizando a página é capaz de mostrar. Essa necessidade ganhou força com os dispositivos móveis, principalmente os Smartphones, que possuem telas com vários padrões de resolução.

O principal objetivo da web é prover a informação de forma íntegra, deixando a formatação da apresentação da informação em segundo plano. É bastante comum ter elementos gráficos como menus, barras de rolagens, ícones e layers para acessar parte das informações. Dependendo do dispositivo que está sendo usado para visualizar esse conteúdo, é necessário redefinir esses elementos gráficos para garantir que a informação seja apresentada de modo lógico, e sem perdas de informação.

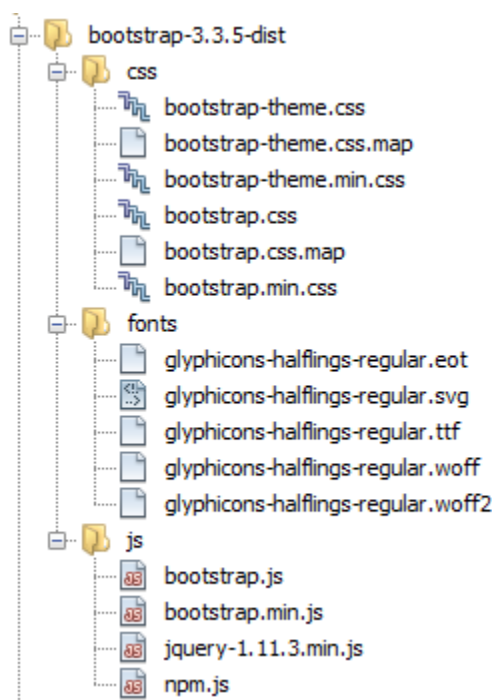
Para solucionar esse problema foi criado o Design Responsivo, que na prática, é a forma que o próprio aplicativo web tem para reconhecer qual é o tamanho da tela que o usuário está usando para visualizar o conteúdo, e realinhar, ou redesenhar, a formatação da tela, para se adequar à renderização que o browser

usado pelo usuário é capaz de executar. Isso permite que o aplicativo funcione bem tanto em browsers usados em PCs, quanto em browsers de dispositivos móveis.

Esses frameworks combinam HTML, JavaScript e CSS para elaborar um mecanismo que executa o design responsivo automaticamente, e na maioria das vezes, o desenvolvedor modifica somente o conteúdo a ser apresentado, e as imagens e cores que permitem alterar a identidade visual do site. Um dos frameworks CSS mais populares é o Bootstrap.

## Bootstrap

Como a maior parte das ferramentas usadas no desenvolvimento de aplicações web, o Bootstrap é o resultado do trabalho de uma equipe de desenvolvedores resolvendo um problema específico. No caso, a equipe que desenvolveu o Twitter desenvolveu um framework para facilitar o seu trabalho, e esse framework começou a ser usado em outras aplicações, até ser criada uma comunidade de desenvolvedores que continua o desenvolvimento do Bootstrap.



Legenda: FIGURA 1 - ESTRUTURA DE TRABALHO DO FRAMEWORK BOOTSTRAP.

O Bootstrap é um dos frameworks mais utilizados atualmente e um dos que mais possui derivados. Ele é tão utilizado que é possível encontrar facilmente ferramentas para auxiliar a utilização do próprio framework. É possível fazer download desse framework no site do projeto (<http://getbootstrap.com/>) e adicionado gratuitamente ao software sendo desenvolvido. Ele possui duas

versões, uma pré-compilada, que pode ser compilada pelo desenvolvedor, ou uma versão já compilada. Apesar de já ser um framework, o BootStrao usa outro framework, o jQuery, que o desenvolvedor precisa baixar e adicionar ao projeto.

## jQuery

O framework jQuery (<https://jquery.com/> (<https://jquery.com/>)) é provavelmente o framework mais usado nos aplicativos web. Ele é um framework que facilita o uso da linguagem de programação JavaScript, que foi originalmente desenvolvida para processar as informações que um usuário digitava num formulário web, diretamente no browser. Ele é um framework simples, mas bastante rápido e eficiente, que manipula o conteúdo HTML que o browser está renderizando. Ele ajuda a manipular a informação sendo exibida, animações e implementa o Ajax de modo mais simples do que é possível fazer com JavaScript. Uma das habilidades dele é detectar qual é o browser sendo usado pelo usuário, e ajustar o uso de JavaScript de acordo com o que o browser permite executar de JavaScript, uma vez que não uma padronização na forma como o browser pode interpretar os scripts escritos em JavaScript.

A possibilidade de ser processado por qualquer browser e de se escrever de modo simplificado os scripts que as interfaces web precisam, além da facilidade de obtenção e distribuição, e de permitir que o desenvolvedor adicione outras funcionalidades ao jQuery, tornaram esse framework bastante popular entre os desenvolvedores.



Legenda: FIGURA 2 - EXEMPLO DE VISUAL DE FORMULÁRIO DESENVOLVIDO COM A COMBINAÇÃO DE BOOTSTRAP E JQUERY

## Ajax

O framework Ajax (ou AJAX), sigla para XML e JavaScript Assíncrono (do inglês Asynchronous JavaScript and XML) é um conjunto de técnicas de desenvolvimento web combinada a um conjunto de tecnologias, que são executadas no browser para estabelecer aplicações web que permitam atualizar o browser com as atualizações de conteúdo que estão sendo realizadas pelo servidor, de modo a diminuir o tráfego de informações entre o browser e o servidor. Como as atualizações podem não ocorrer no servidor, o Ajax não depende das atualizações efetuadas pelo servidor, e por isso, essa comunicação não precisa ser sincronizada.

Usando o Ajax, as aplicações web podem enviar dados ao servidor e verificar se há resposta do servidor. Tudo isso ocorre em background, sem interferir no que está acontecendo com o usuário enquanto ele usa o browser. Por haver esse tipo de desacoplamento, o Ajax permite que as páginas web modifiquem dinamicamente, de modo parcial, o conteúdo sendo mostrado, e assim, o tráfego de dados é reduzido.

# Hibernate

O Hibernate é um dos frameworks mais populares para o desenvolvimento de aplicações web. Os trabalhos realizados no desenvolvimento desse framework e de soluções derivadas desse projeto, podem ser encontrados em seu site (<http://hibernate.org/> (<http://hibernate.org/>)). Ele foi inicialmente desenvolvido para os aplicativos Java Web, se tornou popular entre os desenvolvedores dessa linguagem de programação, e mais tarde, foi adaptado para ter uma versão para a plataforma .Net.

A motivação do Hibernate foi prover uma ferramenta que permitiria desenvolver aplicativos web, baseado em classes que representavam as entidades e relacionamentos definidos no banco de dados, classes que controlam a execução de transações SQL em bases de dados, ele cria formulários baseados em páginas html ou jsp, e classes que conectam as classes que controlam a execução das transações SQL com as páginas HTML e JSP. Todos os arquivos criados pelo Hibernate são estruturados de acordo com o padrão de projeto MVC, e são projetados para prover as seguintes funcionalidades a quem o usa:

- Mapa Objeto/Relacional: o Hibernate constrói seu código-fonte para prover as transações SQL básicas, eliminando essa responsabilidade do desenvolvedor;
- Persistência de dados: o Hibernate possui uma API própria, baseada em JDBC, para persistir os dados usado pelo aplicativo, mas também ele também provê a implementação da persistência através de JPA;
- Garante ao desenvolvedor a possibilidade de implementar código usando os conceitos de orientação a objetos, como herança, polimorfismo, etc...;
- Escalabilidade: permite que o desenvolvedor crie e adicione novas funcionalidades ao seu aplicativo, sem interferir no que já existe;
- Confiabilidade: possui uma grande base de usuários, e um grande número de aplicações desenvolvidas com ele.

## ATIVIDADE FINAL

Para usar um mecanismo que atualize uma página web, sem que seja necessário fazer novamente a carga de toda a página, o desenvolvedor deve usar:

- A. BootStrap
- B. jQuery
- C. Ajax
- D. Hibernate

Para usar um mecanismo que permite atualizar a página web, usando os dados que o usuário preencheu na própria página web, sem haver comunicação com o servidor, deve ser usado:

- A. BootStrap
- B. jQuery
- C. Ajax
- D. Hibernate

Para agilizar o desenvolvimento do aspecto visual das páginas web, o desenvolvedor deve usar:

- A. BootStrap
- B. jQuery
- C. Ajax
- D. Hibernate

Para desenvolver um protótipo de uma aplicação, a partir do modelo de entidades usadas pelo aplicativo, o desenvolver deve optar por:

- A. BootStrap
- B. jQuery
- C. Ajax
- D. Hibernate

## REFERÊNCIA

LINWOOD, J., MINTER, D., OTTINGER, J. *Beginning Hibernate*. APRESS. 2014.

FLORES, R. *Getting Started With Bootstrap 3*. SMASHWORD. 2015.

SILVA, M. S. *jQuery - A Biblioteca do Programador JavaScript*. NOVATEC. 2013.







