

DESVENDANDO O FRAMEWORK ANGULARJS¹

Ricardo Wiesner², Gerson Battisti³.

¹ Pesquisa desenvolvida em empresa de desenvolvimento de Software

² Egresso do Curso de graduação em Ciências da Computação da UNIJUI, rica.wiesner@gmail.com

³ Professor Doutor do Departamento DCEEng, battisti@unijui.edu.br

INTRODUÇÃO

Ao longo do tempo a internet evoluiu significativamente, passando de sites estáticos, que não tinham interação com os usuários, para transformar-se em algo altamente interativo, definido como aplicações web.

A medida em que as aplicações web tornam-se maiores, melhores e mais rápidas, aumentaram a complexidade para seu desenvolvimento. Uma alternativa bem conhecida para auxiliar na complexidade do desenvolvimento é a solução JavaScript/Jquery, no entanto, nem sempre apresenta a estrutura necessária para garantir alta produtividade no desenvolvimento e facilidade de manutenção.

Uma solução para fazer frente ao JavaScript/Jquery é a utilização do framework AngularJS. O AngularJS inicialmente chamado de Angular, era uma biblioteca de código aberto criado como hobby por Misko Hevery. Paralelamente, Shyam Seshadri (líder), junto a Brad Green e Misko, passaram seis meses desenvolvendo o projeto Google Feedback. Apesar de todo esforço para ter um código de fácil manutenção e testável, sentiam-se frustrados com a incapacidade de adicionar novas funcionalidades e muitas linhas de código não haviam sido testadas. A partir desse fato, Misko fez uma afirmação ousada dizendo que era capaz de reproduzir tudo o que haviam feito em duas semanas. Ao final Misko fez em três semanas o que haviam levado seis meses, além disso teve uma significativa redução de código de 18 mil linhas para 1500 linhas. Assim, organizaram uma equipe para desenvolver a ideia principal do AngularJS, tornando-se um dos principais frameworks JavaScript. (Seshadri, Shyam; Green, Brad; 2014)

Esta pesquisa tem como objetivo apresentar uma visão geral do AngularJS enfatizando os conceitos e as características presentes no framework. Com o propósito de proporcionar uma aproximação das tecnologias atuais utilizadas pelo mercado.

METODOLOGIA

Para apresentar os conceitos utilizados no AngularJS foi realizado um estudo da especificação disponível no site <http://angularjs.org>, além do livro “Desenvolvendo Com AngularJS” escrito pela equipe que criou o framework. Após a revisão dos conceitos realizou-se a aplicação dos conceitos

Modalidade do trabalho: Ensaio teórico
Evento: XXIII Seminário de Iniciação Científica

por meio de exemplos práticos e diagramação, com a finalidade de facilitar a compreensão. Na construção dos exemplos foi utilizada a versão 1.3.15 do AngularJS, além da IDE Netbeans 8.0.2 e o navegador Mozilla Firefox.

RESULTADO E DISCUÇÕES

O AngularJS é um framework JavaScript que é baseado no padrão de arquitetura MVC. O padrão MVC separa uma aplicação em três camadas distintas e modulares: O modelo (model) que normalmente representa os dados por trás da aplicação; A visão (view) corresponde o que usuário vê; O controlador (controller) que representa a lógica do negócio e a camada de apresentação. Além da arquitetura MVC o AngularJS segue como filosofia alguns princípios básicos que são:

- Orientação a dados: o framework utiliza o conceito de data-binding bidirecional, ou seja, os dados do modelo são refletidos automaticamente no DOM e vice-versa.
- Declarativo: uma aplicação web é composta por vários trechos de HTML e dados associados. Estes muitas vezes acabamos duplicando como, por exemplo, um campo datepicker. O AngularJS provê um mecanismo para criar uma forma de reutilizar trechos de HTML redundantes, sendo conhecido como diretivas.
- Separação de responsabilidades: como o framework adota o padrão MVC, para estruturar qualquer aplicação naturalmente acaba tendo a separação das responsabilidades.
- Injeção de Dependência (Dependency Injection): o AngularJS é um dos poucos frameworks que implementam um sistema de injeção de dependências. Utiliza-se um mecanismo para apenas solicitar a dependência e o mesmo se encarrega de criá-la.
- Extensível: como citado anteriormente, o AngularJS provê um mecanismo para criar diretivas. Assim, o próprio desenvolvedor pode criar suas próprias diretivas.
- Mecanismo para testes: o AngularJS proporciona um mecanismo que possibilita realizar testes automatizados utilizando Karma, Protractor, entre outros.

Para iniciar um projeto utilizando o AngularJS precisa-se seguir dois passos: carregar o código fonte e inicializá-lo. A Figura 1, apresenta um exemplo básico de Hello World. Para inicializar o AngularJS é necessário colocar a diretiva ng-app (linha 2), para indicar qual parte da página o AngularJS deve manipular. A linha 9 da Figura 1, representa o carregamento do AngularJS.

```
2 <html ng-app>
3 <head>
4 <title>Exemplo 01</title> <meta charset="UTF-8">
5 <link href="./css/bootstrap.css" rel="stylesheet"/>
6 </head>
7 <body>
8 <h3>{{'Hello World'}}</h3>
9 <script src="libs/angular/angular.min.js"></script>
10 </body>
11 </html>
```

Figura 1 - Hello World AngularJS

Modalidade do trabalho: Ensaio teórico
Evento: XXIII Seminário de Iniciação Científica

Um dos princípios do AngularJS é o mecanismo de data-biding que facilita o desenvolvimento de aplicações, evitando códigos maçantes para atualizar a view. A Figura 2, representa o mecanismo de data-biding. O AngularJS faz a compilação do template gerando a view juntamente com os dados, após mantêm os dados da view e do model sincronizados.

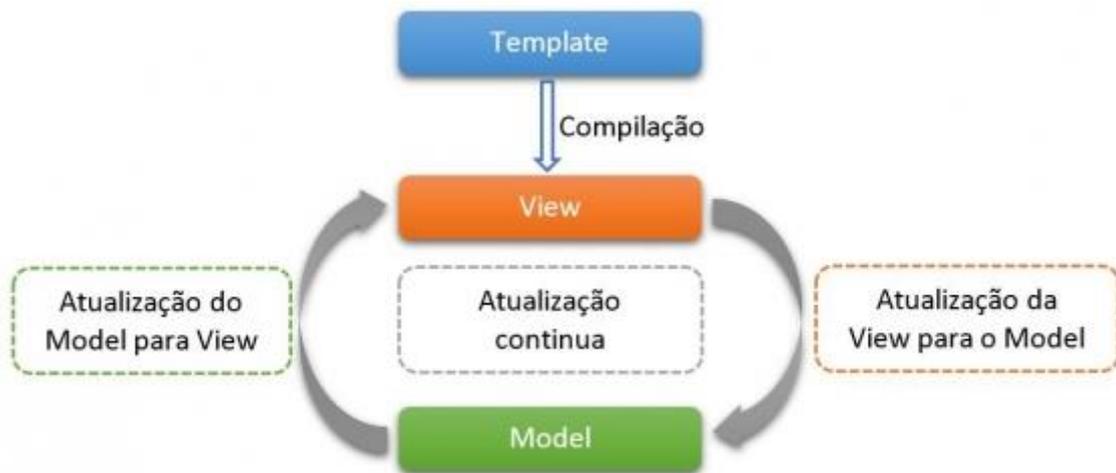


Figura 2 - Mecanismo de data-bing

Com o intuito de facilitar a compreensão do mecanismo de data-biding, será modificado o exemplo da Figura 1. Este exemplo, apresenta conceito de diretivas que consiste em um mecanismo com o objetivo de ampliar o vocabulário do HTML. Além das diretivas do próprio framework, se tem a possibilidade de criar mais diretivas de acordo com a necessidade. No exemplo da Figura 3, tem um campo de texto que ao digitar o valor no campo atualiza automaticamente o valor na variável que é apresentada na linha 11. Essa referência ao model é feita através da diretiva ng-model na linha 10.

```

2 <html ng-app>
3 <head>
4 <title>Exemplo 02</title> <meta charset="UTF-8">
5 <link href="/css/bootstrap.css" rel="stylesheet"/>
6 </head>
7 <body>
8 <div class="container">
9 <div class="page-header"> <h3>Exemplo 02</h3> </div>
10 <div class="col-lg-4"> <input class="form-control" ng-model="nome"/> </div>
11 <h3>Olá {{nome}}</h3>
12 </div>
13 <script src="libs/angular/angular.min.js"></script>
14 </body>
15 </html>

```

Figura 3 - Exemplo Data-Binding

Modalidade do trabalho: Ensaio teórico
Evento: XXIII Seminário de Iniciação Científica

Outro conceito utilizado pelo AngularJS são os módulos. Consistem em um modo de agrupar códigos relevantes por meio de um nome. Pode-se fazer uma analogia aos pacotes da Linguagem Java. Um módulo do AngularJS contém duas partes:

- Um módulo pode ter seus próprios controller, services, factories, diretivas. Estes podem ser acessadas por outros módulos.
- Um módulo também pode ter a dependência de outros módulos.

Para criar um módulo utilizando o AngularJS, pode ser feita da seguinte forma:
`angular.module('primeiroModulo',[]);`

O primeiro argumento da função é o nome do módulo propriamente dito. Neste caso define-se que o módulo tem o nome de "primeiroModulo". O segundo argumento é um array de módulos que são as dependências que serão carregadas. Um módulo com dependência pode ser feito da seguinte forma:

```
angular.module ('segundoModulo', ['primeiroModulo']);
```

Assim foi definida um modulo chamado 'segundoModulo' e tem a dependência do modulo 'primeiroModulo'. É importante salientar, caso seja esquecido de passar os dois argumentos para função, ao invés de ser criado um módulo será retornado o módulo que tenha o mesmo nome, quando não tiver o módulo registrado será lançado um erro, dizendo que o módulo não foi encontrado. Para chamar o módulo registrado é feito da seguinte forma:

```
angular.module('segundoModulo');
```

A partir dos módulos são criado os controllers. Os controllers do AngularJS são as funções que estão diretamente ligadas a view. Ou seja, são elas que irão dizer quais dados vão aparecer para o usuário, tratar a interação com usuário etc. Para definir um controller é necessário ter um módulo criado. Isso pode ser feito da seguinte forma:

```
angular.module('primeiroModulo',[]).controller('primeiroController' , function (){});
```

Por intermédio da função ".controller" criou-se o controller chamado 'primeiroController'. A função recebe dois argumentos, o primeiro é referente ao nome e o segundo é a função propriamente dita. O AngularJS, existem outros recursos que podem ser utilizados para organizar, reutilizar, agilizar o desenvolvimento.

CONCLUSÃO

Neste trabalho foi abordada uma visão geral do AngularJS. Um framework possui vários recursos como criar diretivas próprias, criar controller, services e factory que proporcionam

Modalidade do trabalho: Ensaio teórico
Evento: XXIII Seminário de Iniciação Científica

reaproveitamento de código e conseqüentemente maior produtividade. O AngularJS não é nenhuma solução que resolve todos os problemas do desenvolvimento web, porém possui características interessantes que atendem um conjunto de necessidades até então sem solução. A aplicação do framework em projetos atuais deve ser decidida baseada em análises específicas considerando e compreendendo o propósito do mesmo, ou seja, nem todos os casos é benéfico migrar para o AngularJS considerando o tempo de aprendizagem e a aplicabilidade.

Um das características mais importantes do AngularJS é a possibilidade de usar a mesma aplicação Web (que é utilizada por meio de um navegador), em tablets ou celulares, utilizando a phonegap para converter a aplicação em uma app nativa.

Os códigos fontes dos exemplos estão disponíveis para download no link:
https://www.dropbox.com/sh/yqqfh44s67jgw1o/AAA0pw85YRj_zts4yFHwf2Sfa?dl=0

REFERÊNCIAS BIBLIOGRÁFICAS

PEREIRA, Michael Henrique R. AngularJS: uma abordagem prática e objetiva. 1.ed. São Paulo: Novatec Editora Ltda, 2014.