

JavaScript Firme: Módulos com RequireJS e BDD com Jasmine

André Valenti

*#TDC2013 - The Developer's Conference
#QConSP*

São Paulo - SP

Sobre mim



André Valenti

Professor de Ensino Tecnológico no
IFSP São Carlos



Sobre mim



- Desenvolvedor durante 4,5 anos

- Principais experiências:

- Java
- JavaScript / Node.js / CoffeeScript
- Jogos HTML5
- Groovy / Grails
- PostgreSQL



Pergunta:

- Você tem medo de alterar seu projeto?

Estrutura

- Projetos crescem
- Alterações são inevitáveis
- Firme na base, flexível no topo

Firmeza

estrutura_firme \Rightarrow sossego

Firmeza

~~estrutura_firme \Rightarrow sossego~~

\neg estrutura_firme \Rightarrow \neg sossego

estrutura_firme tende a \Rightarrow sossego

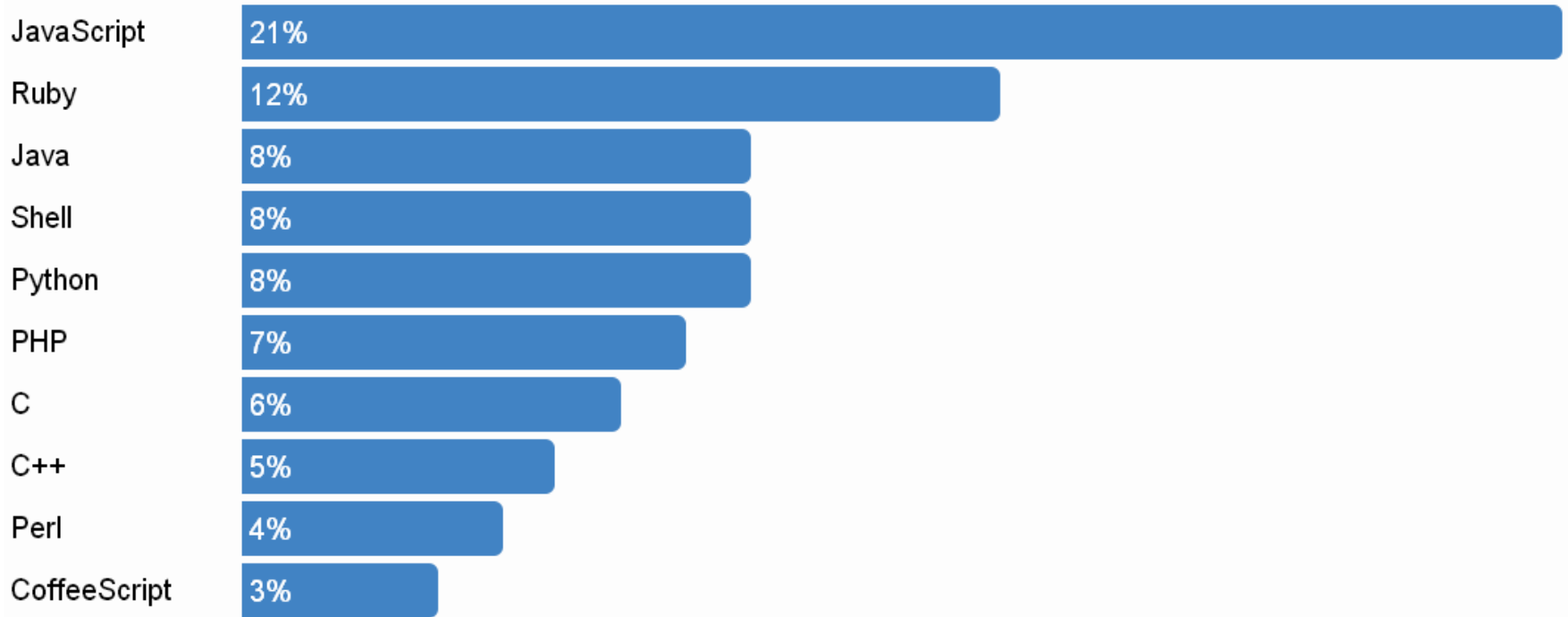
Estrutura

- Sintomas de base frouxa:
 - Criar novo módulo dá muito trabalho
 - Alterações de negócio quebram o projeto
- Sintomas de topo engessado:
 - Evoluir código dá muito trabalho
 - Refatorar dá muito trabalho

JavaScript

Linguagem de programação com uma certa importância...

JavaScript



<https://github.com/languages>

JavaScript

Orientação a Objetos em JavaScript:

JavaScript

MinhaClasse
- valor : int
+ metodo1() : void + metodo2(a : int, b : int) : int

JavaScript

```
function MinhaClasse(valor) {  
  this._valor = valor;  
}
```

MinhaClasse
- valor : int
+ metodo1() : void + metodo2(a : int, b : int) : int

JavaScript

```
function MinhaClasse(valor) {  
  this._valor = valor;  
}
```

```
MinhaClasse.prototype.metodo1 = function() {  
};
```

MinhaClasse
- valor : int
+ metodo1() : void + metodo2(a : int, b : int) : int

JavaScript

MinhaClasse
- valor : int
+ metodo1() : void + metodo2(a : int, b : int) : int

```
function MinhaClasse(valor) {  
  this._valor = valor;  
}
```

```
MinhaClasse.prototype.metodo1 = function() {  
};
```

```
MinhaClasse.prototype.metodo2 = function(a, b) {  
  return this._valor + a + b;  
};
```

JavaScript

MinhaClasse
- valor : int
+ metodo1() : void + metodo2(a : int, b : int) : int

```
function MinhaClasse(valor) {  
  this._valor = valor;  
}
```

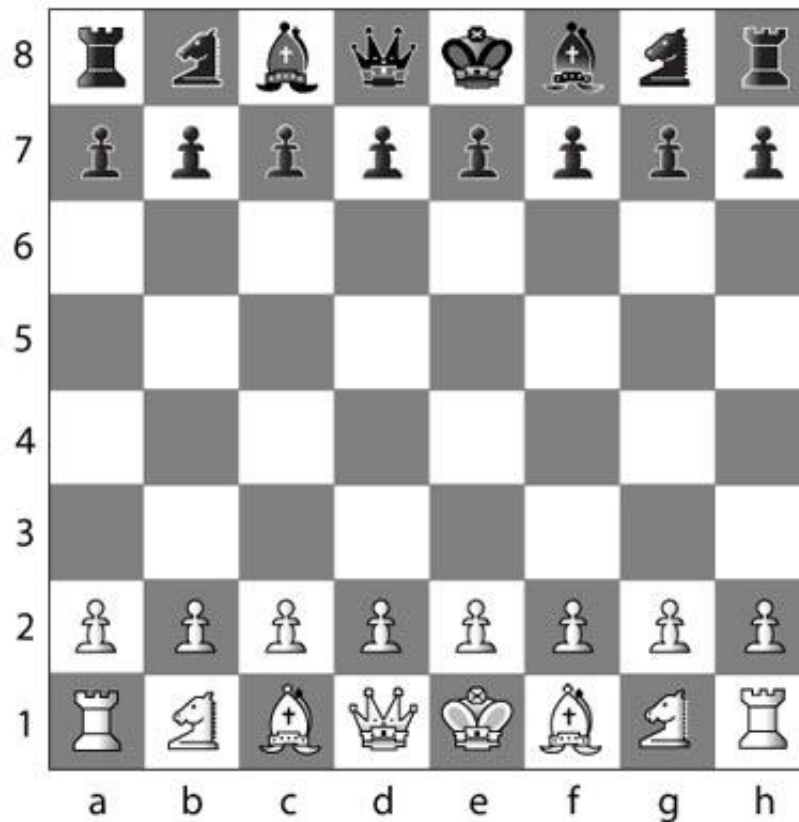
```
MinhaClasse.prototype.metodo1 = function() {  
};
```

```
MinhaClasse.prototype.metodo2 = function(a, b) {  
  return this._valor + a + b;  
};
```

```
new MinhaClasse(1).metodo2(3, 4);
```


JavaScript

Exemplo: jogo de xadrez



JavaScript

- Como organizar o código?
- Como estruturar o projeto?

A promotional image for the TV show MacGyver. Keanu Reeves, as the character MacGyver, is shown from the chest up, wearing his signature brown leather jacket over a red shirt and gold-rimmed aviator sunglasses. He is holding a large, white, cylindrical rocket or missile over his right shoulder. The rocket has a tan-colored nose cone and a small white wing. The background is a clear, bright blue sky. The overall tone is adventurous and heroic.

***Cria logo um HTML aí
e manda ver!***

MACGYVER

xadrez-macgyver-1

```
1 <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
2 .. "http://www.w3.org/TR/html4/loose.dtd">
3 <html>
4 <head>
5 .. <title>Título da Página</title>
6 </head>
7
8 <body>
9 .. <h1>Título da Página</h1>
10
11 .. <script>
12 ..
13 ..
14 .... // Seu código JavaScript aqui!
15 ....
16 ....
17 .. </script>
18
19 </body>
20 </html>
```

xadrez-macgyver-1

Em pouco tempo, seu projeto ficará assim:

xadrez-macgyver-1

JavaScript

- Dá para melhorar isso aí!
- De que jeito?

A promotional image for the TV show MacGyver. Keanu Reeves, as the character MacGyver, is shown from the chest up, wearing a brown leather jacket over a red shirt and gold-rimmed aviator sunglasses. He is holding a large, white, cylindrical rocket or missile over his right shoulder. The rocket has a tan-colored nose cone and a small white fin. The background is a clear blue sky. A dark blue banner is at the top, and the title 'MACGYVER' is at the bottom in large, stylized letters.

***Cria uns .js, mete uns
<script> e já era!***

MACGYVER

xadrez-macgyver-2

```
<html>
  <head> ... </head>
  <body> ...
    <script src="js/src/Jogador.js"></script>
    <script src="js/src/Posicao.js"></script>
    <script src="js/src/Xadrez.js"></script>
    <script src="js/src/xadrez-main.js"></script>
  </body>
</html>
```

xadrez-macgyver-(1|2)

Código dos projetos disponível no GitHub:

```
git clone https://github.com/awvalenti/javascript-firme.git
```

```
cd javascript-firme/xadrez-macgyver-1
```

```
cd javascript-firme/xadrez-macgyver-2
```

xadrez-macgyver-(1|2)

- Problemas:
 - Exige ficar acrescentando <script>
 - Cria globais (variáveis, funções, classes)
 - Dependências ficam mascaradas
 - Forte tendência a acoplamento
 - Não vai rodar no NodeJS

RequireJS

- <http://requirejs.org/>
- Gestor de módulos
- Compatível com navegadores e com NodeJS

RequireJS

- Deixa claras as dependências
- Evita globais
- Ajuda a reduzir acoplamento
- Carrega scripts automaticamente

Sem RequireJS

```
// MinhaClasse.js
```

```
function MinhaClasse() {  
    this._objeto1 = new Classe1();  
    this._objeto2 = new Classe2();  
}
```

```
// Problemas:
```

```
// - dependências implícitas: Classe1 e Classe2  
//  
// - obrigatoriedade de elementos <script> no HTML  
//  
// - globais: Classe1, Classe2 e MinhaClasse
```

Com RequireJS

```
// MinhaClasse.js
define(['Classe1', 'Classe2'], // dependências explícitas
function(Classe1, Classe2) {
    function MinhaClasse() {
        // tudo é importado automaticamente
        this._objeto1 = new Classe1();
        this._objeto2 = new Classe2();
    }

    MinhaClasse.prototype.meuMetodo = function() {};

    return MinhaClasse; // não gera globais
});
```

Sem RequireJS

```
<html>
  <head> ... </head>
  <body> ...
    <script src="js/src/Jogador.js"></script>
    <script src="js/src/Posicao.js"></script>
    <script src="js/src/Xadrez.js"></script>
    <script src="js/src/xadrez-main.js"></script>
  </body>
</html>
```


Com RequireJS

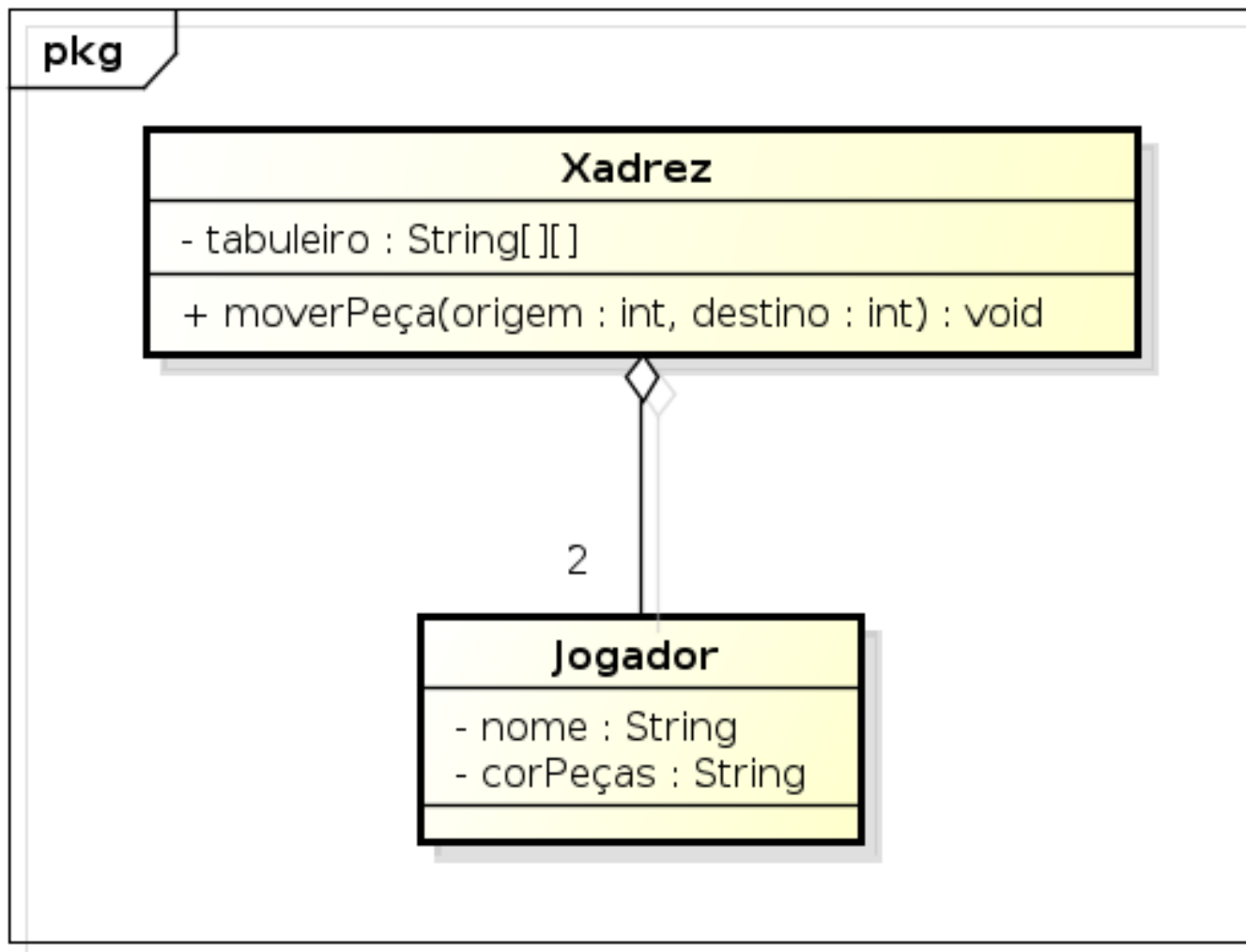
```
<html>
  <head> ... </head>
  <body> ...
    <script src="js/lib/require.js"
      data-main="js/src/xadrez-main.js"></script>
  </body>
</html>
```

xadrez-modular

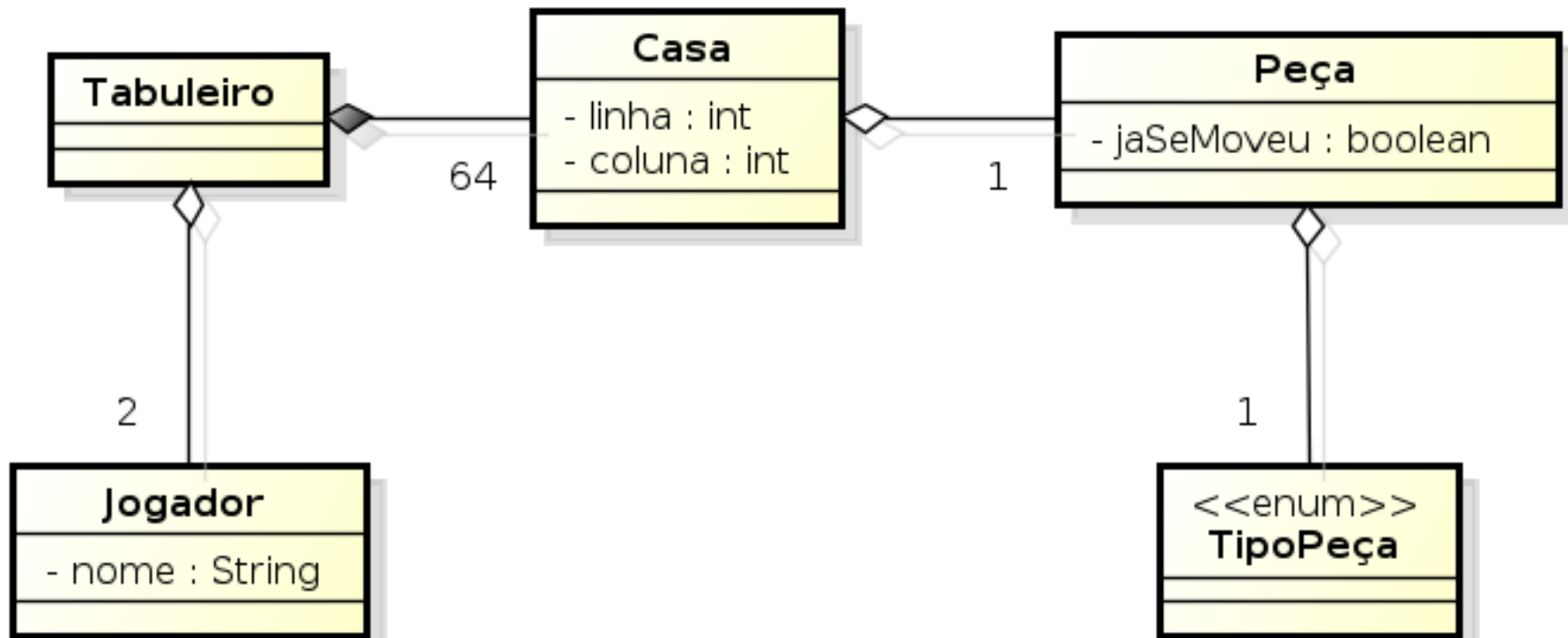
```
git clone https://github.com/awvalenti/javascript-firme.git  
cd javascript-firme/xadrez-modular
```

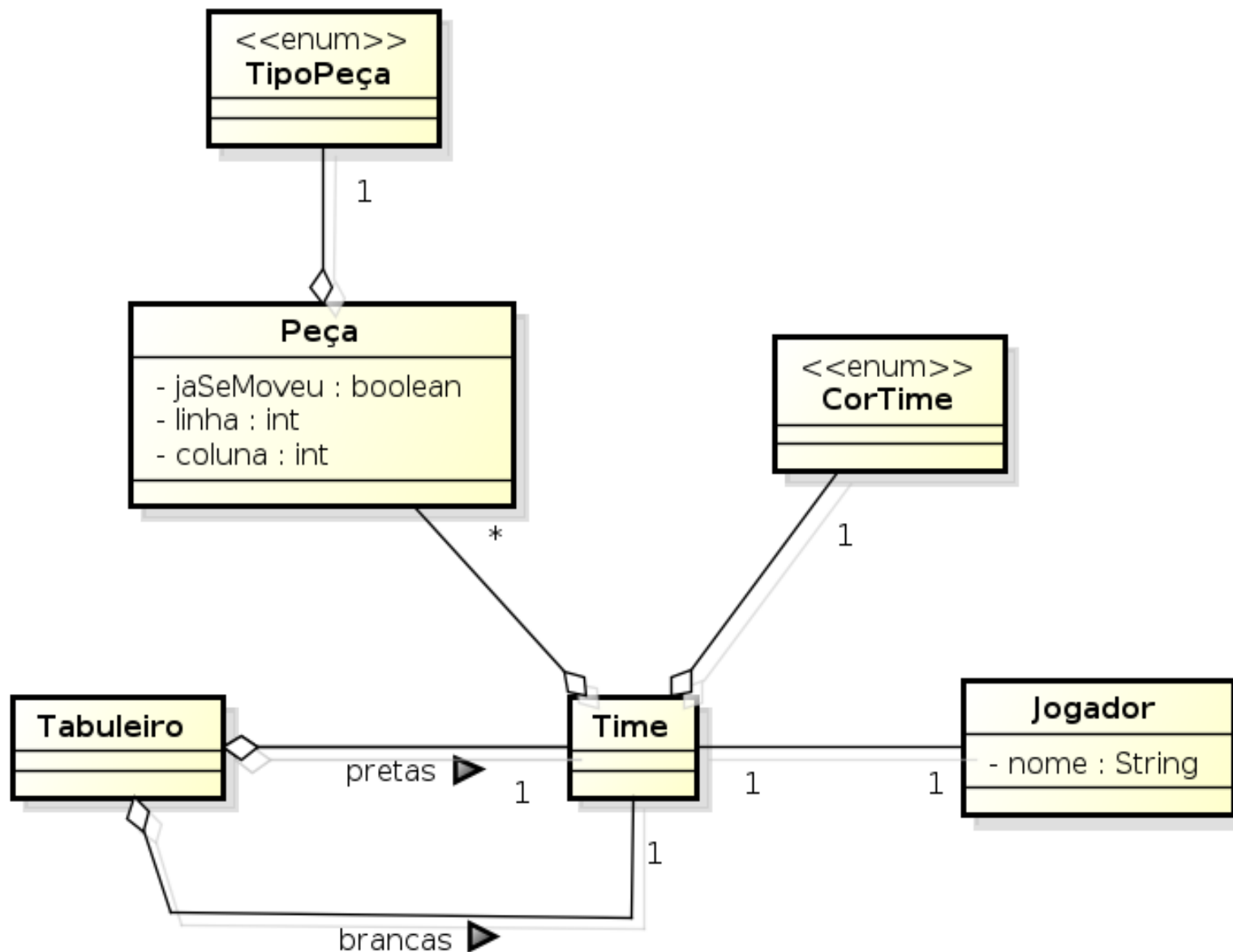
Modelagem:

Xadrez pode ser modelado de várias maneiras...



pkg





Modelagem:

- Perguntas:
 - Qual das três é mais adequada?
 - As classes estão boas assim?
 - Será útil ter a classe Jogador desde já?

Modelagem:

- São perguntas difíceis de se responder...

Modelagem:

- São perguntas difíceis de se responder...
- ...no início do projeto!

BDD

- http://en.wikipedia.org/wiki/Behavior-driven_development
- Especificação de comportamentos antes de implementar funcionalidades
- Requisitos guiando o *design* de código, de maneira incremental
- APIs sendo criadas antes de implementações

Jasmine

- <http://pivotal.github.io/jasmine/>
- *Framework* para BDD em JavaScript
- Roda tanto no navegador quanto no NodeJS
- Escrevem-se *specs* usando-se `describe` e `it`

Jasmine

```
describe('MeuObjeto', function() {  
  describe('no cenário XYZ', function() {  
    it('deve comportar-se da maneira W', function() {  
      expect(meuObjeto.getOQueEstaFazendo()).toBe('W');  
    });  
  
    it('deve acontecer tal outra coisa', function() {  
      expect(meuObjeto.getTalOutraCoisa()).toBe(true);  
    });  
  });  
});
```

Jasmine

- Na verdade, usa-se também um `beforeEach` para criar o objeto:

Jasmine

```
describe('MeuObjeto', function() {  
  
  var meuObjeto = null;  
  beforeEach(function() {  
    meuObjeto = new MeuObjeto();  
  });  
  
  describe('no cenário XYZ', function() {  
    it('deve comportar-se da maneira W', function() {  
      expect(meuObjeto.getOQueEstaFazendo()).toBe('W');  
    });  
  
    it('deve acontecer tal outra coisa', function() {  
      expect(meuObjeto.getTalOutraCoisa()).toBe(true);  
    });  
  });  
});
```

Exemplo padrão do Jasmine

```
git clone https://github.com/awvalenti/javascript-firme.git  
cd javascript-firme/exemplo-padrao-jasmine
```

(ou <https://github.com/downloads/pivotal/jasmine/jasmine-standalone-1.3.1.zip>)

xadrez-especificado

```
git clone https://github.com/awvalenti/javascript-firme.git  
cd javascript-firme/xadrez-especificado
```


xadrez-modular-e-especificado

```
git clone https://github.com/awvalenti/javascript-firme.git  
cd javascript-firme/xadrez-modular-e-especificado
```

Conclusões :

- Firme na base, flexível no topo
- Fazer sem estrutura...
 - ...funciona no começo, depois capenga
 - (experiência própria!)

Conclusões :

Não seja MacGyver, seja Profissional!

Conclusões :

MacGyvers têm mais dor de cabeça



Conclusões :

Profissionais têm mais sossego



Obrigado!

- André Valenti ("Fi")
- São Carlos-SP
- Professor no IFSP



- Contato:
 - E-mail: awvalenti@gmail.com
 - Twitter: [@awvFi](https://twitter.com/awvFi)
 - Blog: aosfi.blogspot.com
 - SlideShare: slideshare.net/AndrFi