Munif Gebara Junior
munifgebara.github.io
munif@munif.com.br
(44) 98403-4939

Sobre o Munif

Pai

Primeiro curso de programação em 1984

Nerd aos 10 anos

Ciência da Computação (UEM) 1994-1997

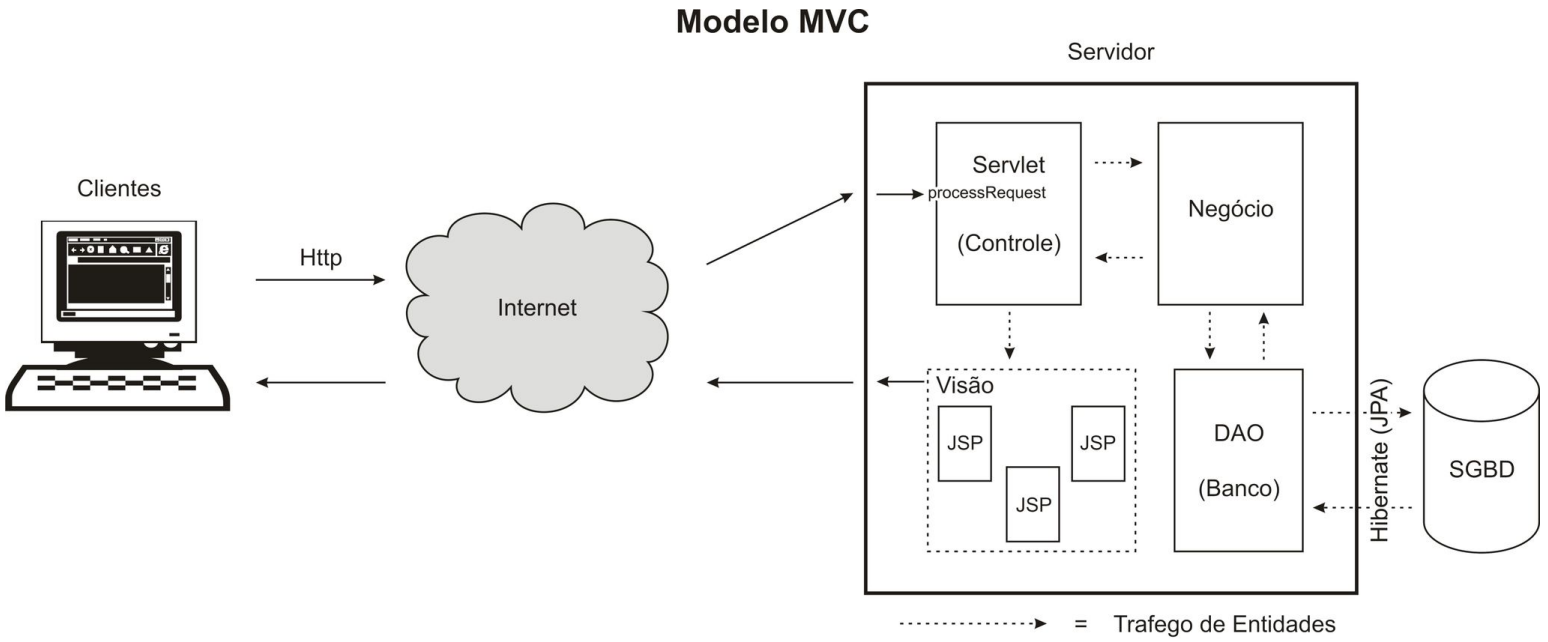
Mestrado Robótica (UTFPR) 1998-2000

Professor Nível Superior e Especialização 2000

Agenda

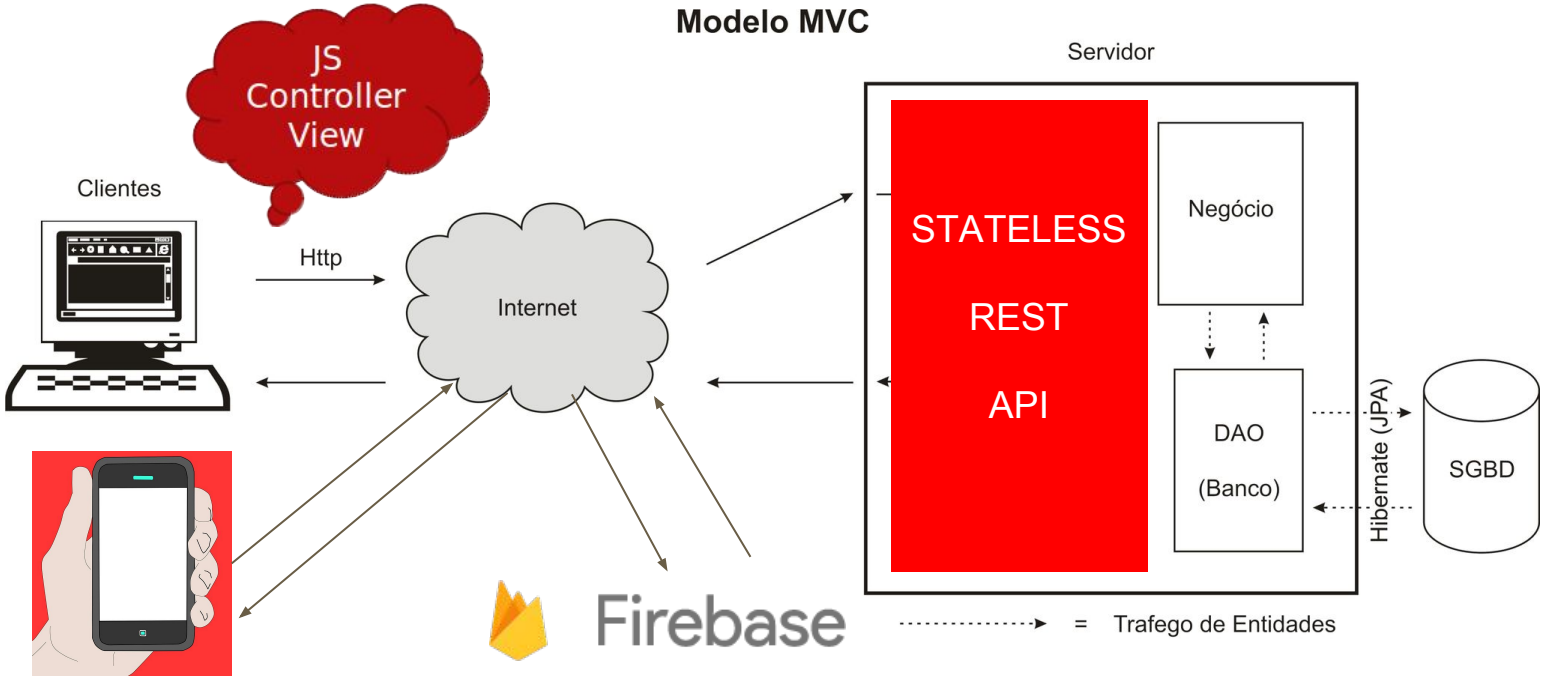
Conceitos

Motivação: Aplicações Desacopladas



Elaborado por Vinicius Sotti

Motivação: Aplicações Desacopladas



Tecnologias

Ferramentas

Require Api padrão

texto.txt

Arquivo de demonstração
munifgebara@gmail.com

Dados do usuario
User Name : munif
Home Folder: /home/munif

Api padrão https://nodejs.org/dist/latest-v8.x/docs

```
console.log(
const fs = require("fs");
const os = require("os");
let user = os.userInfo();
fs.appendFileSync("texto.txt",  usuario\n");
fs.appendFileSync("texto.txt", `User Name  : ${user.username}\n`);
fs.appendFileSync("texto.txt", `Home Folder: ${user.homedir}\n\n`);
console.log(user);
```

Não é “ nem ' é `

Permite isso

Require outros arquivos

app2.js

```
const notes = require("./note
console.log("app2.js");
console.log(`Sua idade é ${no
notes.adiciona("Bom dia");
notes.adiciona("Boa tarde");
notes.adiciona(
notes.lista();

console.log(notes.notas);
```

Não funciona

notes.js

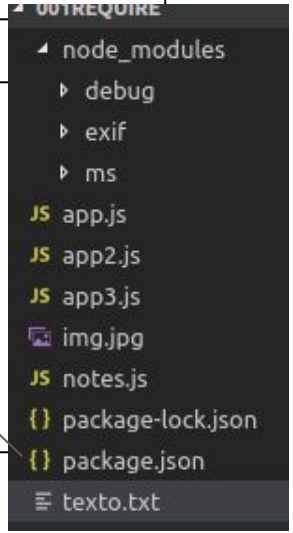
```
console.log('Arquivo notes.js');
let nota
module.e                .idade = 33;
module.exports.adiciona = (nota) => {
    notas.push(nota);
}
module.exports.lista = () => {
    notas.forEach(v => {
        console.log(v);
    });
}
```

Exporta

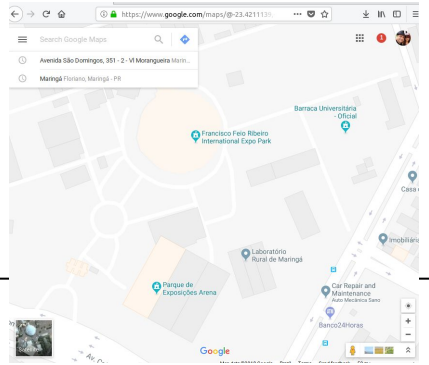
Require bibliotecas de terceiros com NPM

```
mkdir dadosfotos
cd dadosfotos
npm init
npm install --save exif
```

```
{
  "name": "001require",
  "version": "1.0.0",
...
  "dependencies": {
     "exif": "^0.6.0"
  }
}
```




```
const ExifImage = require('exif').ExifImage;

function converteCoordenada(p, ref) {...}

new ExifImage({ image: 'img.jpg' },
     function (error, exifData) {
     if (error) {
          console.log('Error: ' + error.message);
     } else {
          console.log(`Aparelho ${exifData.image.Make
${exifData.image.Model}`);
     }
...
     });
```

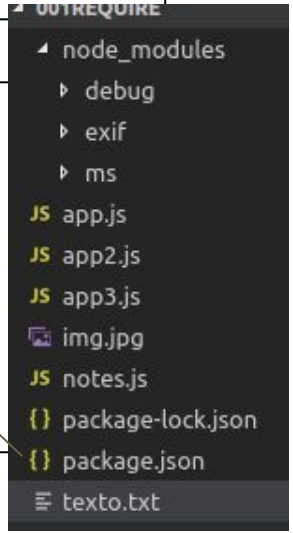


Yargs para ler entradas de linha de comando

```
mkdir dadosfotos
cd dadosfotos
npm init
npm install --save yargs
```

```
const argv = require('yargs').argv;

console.log(argv);

//node app4.js teste  --min 10 --max 20 -a -b -c
```

```
{
  "name": "001require",
  "version": "1.0.0",
...
  "dependencies": {
    "yargs": "^11.0.0"
  }
}
```

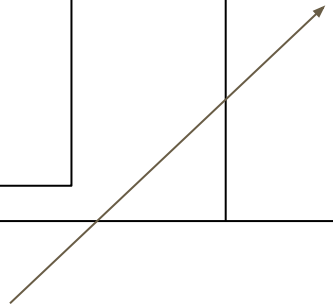



```
{ _: [ 'teste' ],
  help: false,
  version: false,
  min: 10,
  max: 20,
  a: true,
  b: true,
  c: true,
  '$0': 'app4.js' }
```

JSON

```
const notes =
require("./notes-disk");
notes.adiciona("Boa tarde");
notes.adiciona("Boa noite");
notes.lista();
```

```
["Bom dia","Boa tarde"]
```



```
const fs = require("fs");
let notas = [];
notas = JSON.parse(fs.readFileSync("notas.json"));
module.exports.idade = 33;
module.exports.adiciona = (nota) => {
     notas.push(nota);
     fs.writeFileSync("notas.json", JSON.stringify(notas));
}
```

Api - Produtos (service)

```
const fs = require("fs");
let lista = [];
const colecao = "produtos";
module.exports.colecao = colecao;
lista = JSON.parse(fs.readFileSync(colecao + ".json"));
module.exports.adiciona = (item) => {
     if (lista.length == 0) {
     item.id = 1;
     }
     else {
     item.id = lista[lista.length - 1].id + 1;
     }
     lista.push(item);
     fs.writeFileSync(colecao + ".json", JSON.stringify(lista));
     return item;
}
module.exports.lista = () => {
     return lista;
}

(continua)
```

Api - Produtos (service)

```
module.exports.porId = (id) => {
     let i = lista.find(item => item.id == id);
     if (i) {       return i; }
     return {};
}
module.exports.altera = (item) => {
     let r = {};
     lista.forEach(it => {
     if (it.id == item.id) {
          it.nome = item.nome;
          r = it;
          fs.writeFileSync(colecao + ".json", JSON.stringify(lista));    } })
     return r;
}
module.exports.remove = (id) => {
     let r = { id: id, nome: null };
     lista = lista.filter(it => it.id != id);
     fs.writeFileSync(colecao + ".json", JSON.stringify(lista));
     return r;
}
```

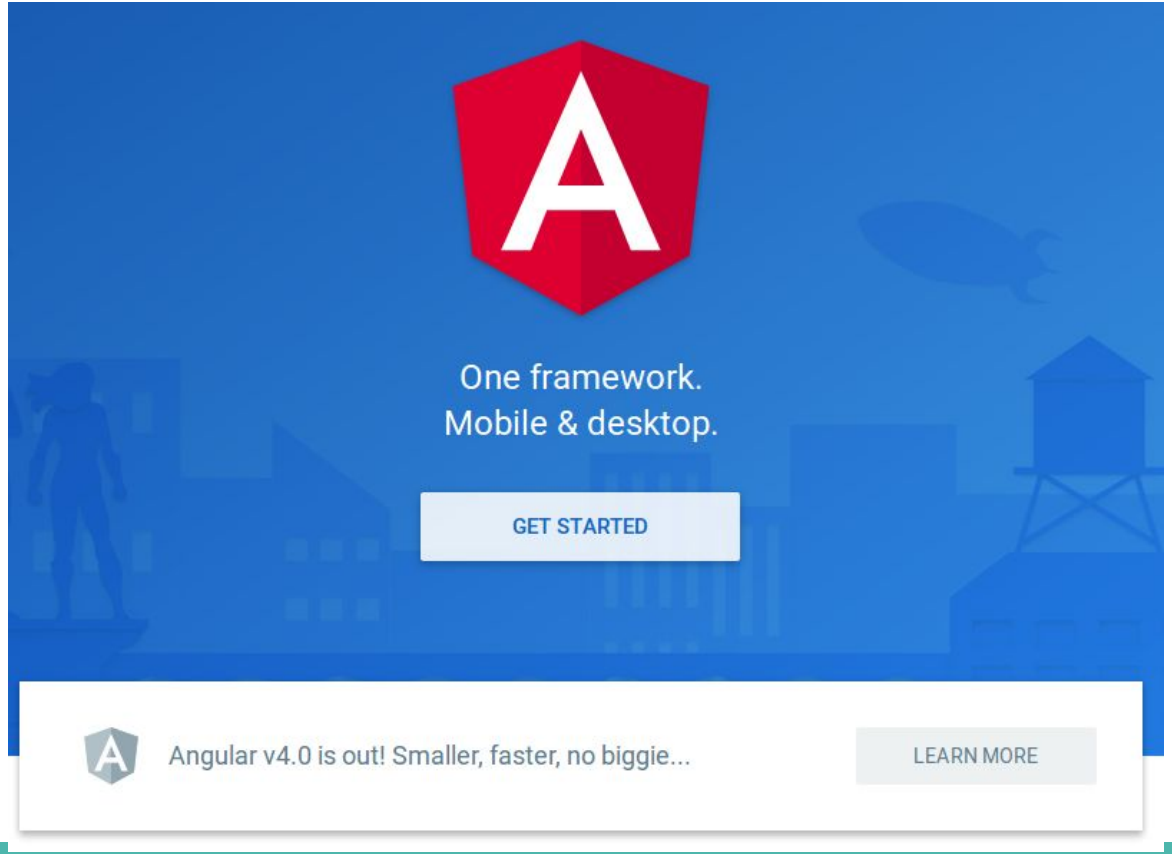
Api - Produtos (controller)

```
const express = require('express');
const bodyParser = require('body-parser');
const produtos = require('./produtos');
var app = express();
app.use(bodyParser.json());
app.get('/', function (req, res) {
     res.send('Hello World')
})
app.get(`/${produtos.colecao}`, (req, res) => {
     res.send(produtos.lista());
});
app.get(`/${produtos.colecao}/:id`, (req, res) => {
     res.send(produtos.porId(req.params.id));
});
app.post(`/${produtos.colecao}`, (req, res) => {
     res.send(produtos.adiciona({ id: null, nome: req.body.nome }));
});
```

Api - Produtos (controller)

```
app.put(`/${produtos.colecao}/:id`, (req, res) => {
     res.send(produtos.altera({ id: req.params.id, nome: req.body.nome }));
});

app.put(`/${produtos.colecao}`, (req, res) => {
     res.send(produtos.altera({ id: req.body.id, nome: req.body.nome }));
});


app.delete(`/${produtos.colecao}/:id`, (req, res) => {
     res.send(produtos.remove(req.params.id));
});


app.listen(3000)
```

Angular


One framework.
Mobile & desktop.
GET STARTED
Angular v4.0 is out! Smaller, faster, no biggie...
LEARN MORE

Conceitos - Componente

- **Encapsula:**
  - Template
  - Metadata
  - Data Binding
  - Comportamento da View

Conceitos - Diretiva

Responsável por manipular elementos DOM e seu comportamento

Conceitos - Serviços

Comunicação com Backend

Reutilização

Métodos de Negócio (front) ????

Espiando

github.com/munifgebara/angular4

hello/index.html

```
...
<script
src="https://unpkg.com/@angular/platform-browser-dynamic@2.
0.1/bundles/platform-browser-dynamic.umd.js"></script>
 </head>
<body>
<my-app>Loading…</my-app>
...
<script src="app.js"></script>
...
```

github.com/munifgebara/angular4

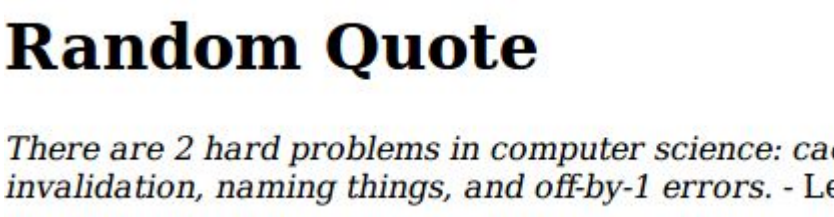
hello/app.js

```
(function() {
 var AppComponent = ng.core.Component({    selector: 'my-app',
    template: '<h1>Hello World!</h1>'  })
.Class({     constructor: function() { }  });

 var
AppModule=ng.core.NgModule({imports:[ng.platformBrowser.BrowserModule],
    declarations:[AppComponent],  bootstrap: [AppComponent]
   })
   .Class({  constructor:function() { }   });

   ng.platformBrowserDynamic.platformBrowserDynamic()
    .bootstrapModule(AppModule);
})();
```

Hello World!

github.com/munifgebara/angular4

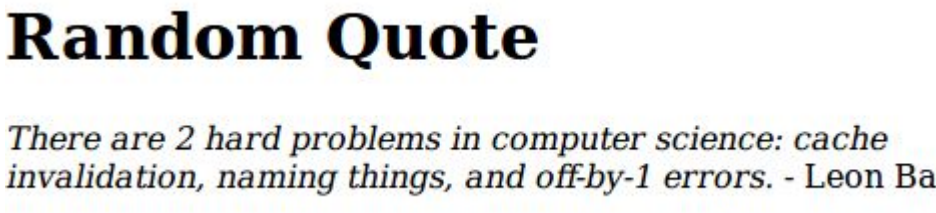
random-quote1/app.js




```
(function() {
  var Component = ng.core.Component;
  var NgModule = ng.core.NgModule;
  var BrowserModule = ng.platformBrowser.BrowserModule;
  var platformBrowserDynamic =
ng.platformBrowserDynamic.platformBrowserDynamic;

  var RandomQuoteComponent = Component({
    selector: 'random-quote',
    template: '<p><em>{{quote.line}}</em> - {{quote.author}}</p>'
  })
  .Class({
    constructor: function() {
    var randomIndex = Math.floor(Math.random() * quotes.length);
    this.quote = quotes[randomIndex];
    }
  });
...
```

github.com/munifgebara/angular4 random-quote1/app.js

```
var AppComponent = Component({
  selector: 'my-app',
  template:
  '<h1>Random Quote</h1>' +
  '<random-quote></random-quote>'
})
.Class({
  constructor: function() { }
});
var AppModule = NgModule({
  imports: [BrowserModule],
  declarations: [AppComponent, RandomQuoteComponent],
  bootstrap: [AppComponent]
})
.Class({   constructor: function() { }  });
platformBrowserDynamic().bootstrapModule(AppModule);
...
```



github.com/munifgebara/angular4 random-quote1/app.js

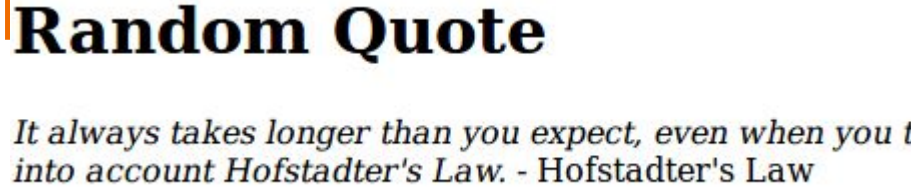
```
  platformBrowserDynamic().bootstrapModule(AppModule);

  var quotes = [
    {"line": "Debugging is twice ...",     "author": "Brian W. Kernighan"
    },
    {"line": "Walking on water ...",   "author": "Edward V Berard"},
   …];

})();
```

Random Quote

There are 2 hard problems in computer science: cache invalidation, naming things, and off-by-1 errors. - Leon Bambrick

github.com/munifgebara/angular4

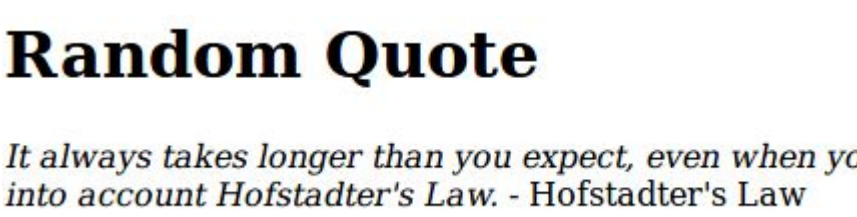
random-quote2/app.js SERVICES




```
(function() {
  var Class = ng.core.Class;
  var Component = ng.core.Component;
  var NgModule = ng.core.NgModule;
  var BrowserModule = ng.platformBrowser.BrowserModule;
  var platformBrowserDynamic =
ng.platformBrowserDynamic.platformBrowserDynamic;

  var QuoteService = Class({
    constructor: function QuoteService() {
    this.quotes = sampleQuotes;
    },
    getRandomQuote: function() {
    var randomIndex = Math.floor(Math.random() * this.quotes.length);
    return this.quotes[randomIndex];
    }
  });
```

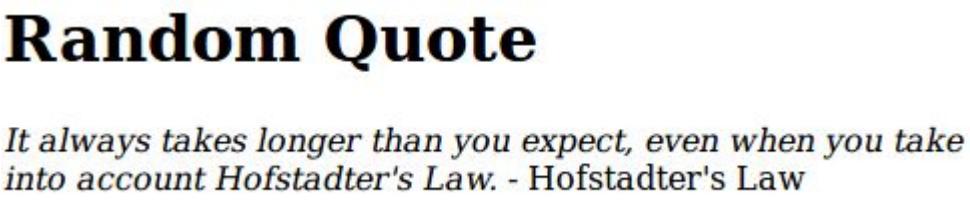
github.com/munifgebara/angular4 random-quote2/app.js SERVICES

```
var RandomQuoteComponent = Component({
    selector: 'random-quote',
    template: '<p><em>{{quote.line}}</em> - {{quote.author}}</p>'
  })


  .Class({
    constructor: [QuoteService, function
RandomQuoteComponent(quoteService) {
    this.quote = quoteService.getRandomQuote();
    }]
  });

  var AppComponent = Component({...  })
  .Class({
    constructor: function AppComponent() { }
  });
```



github.com/munifgebara/angular4 random-quote2/app.js SERVICES

```
  var AppModule = NgModule({
    imports: [BrowserModule],
    declarations: [AppComponent, RandomQuoteComponent],
    providers: [QuoteService],
    bootstrap: [AppComponent]
  })
  .Class({
    constructor: function() { }
  });

  platformBrowserDynamic().bootstrapModule(AppModule);

  var sampleQuotes = [ ...  ];

})();
```

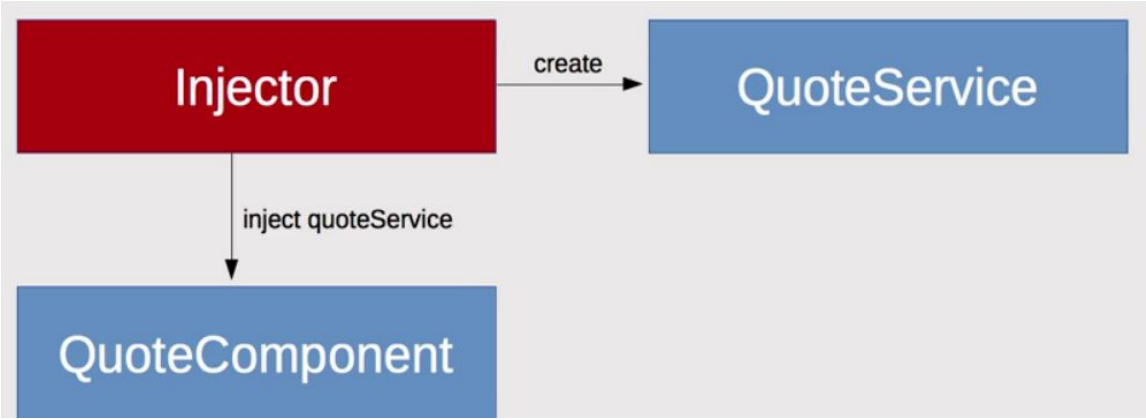


Dependency Injection

Reutilização

Manutenção

Teste

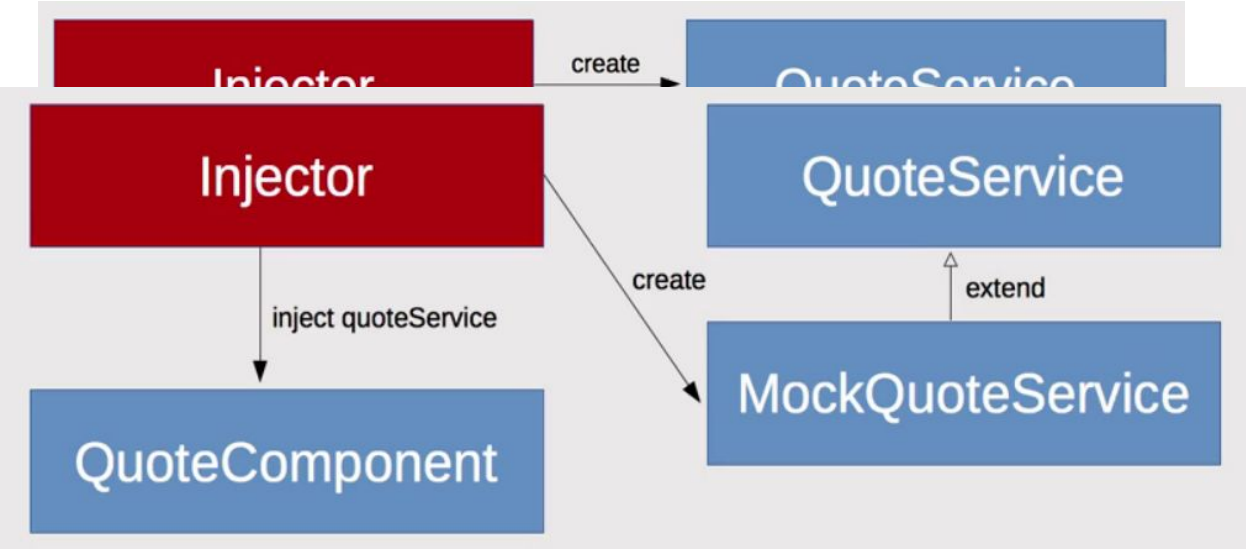


Dependency Injection

Reutilização

Manutenção

Teste

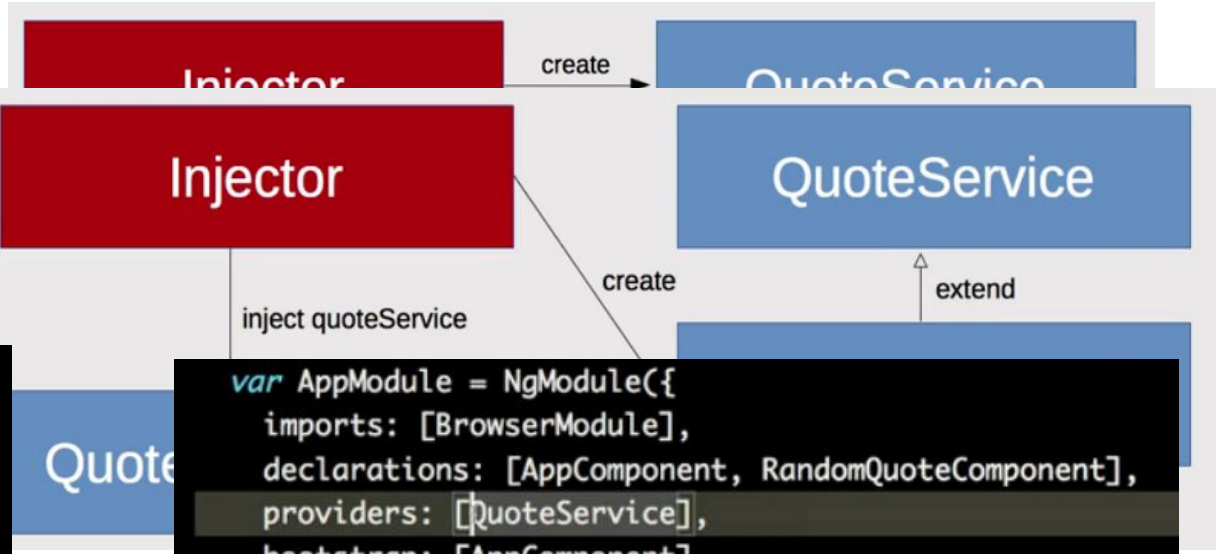


Dependency Injection

Reutilização

Manutenção

Teste




```
var MockQuoteService = Class({
  constructor: function() { },
  getRandomQuote: function() {
    return {
      line: 'A mock quote.',
      author: 'Mock Author'
    };
  }
});
```

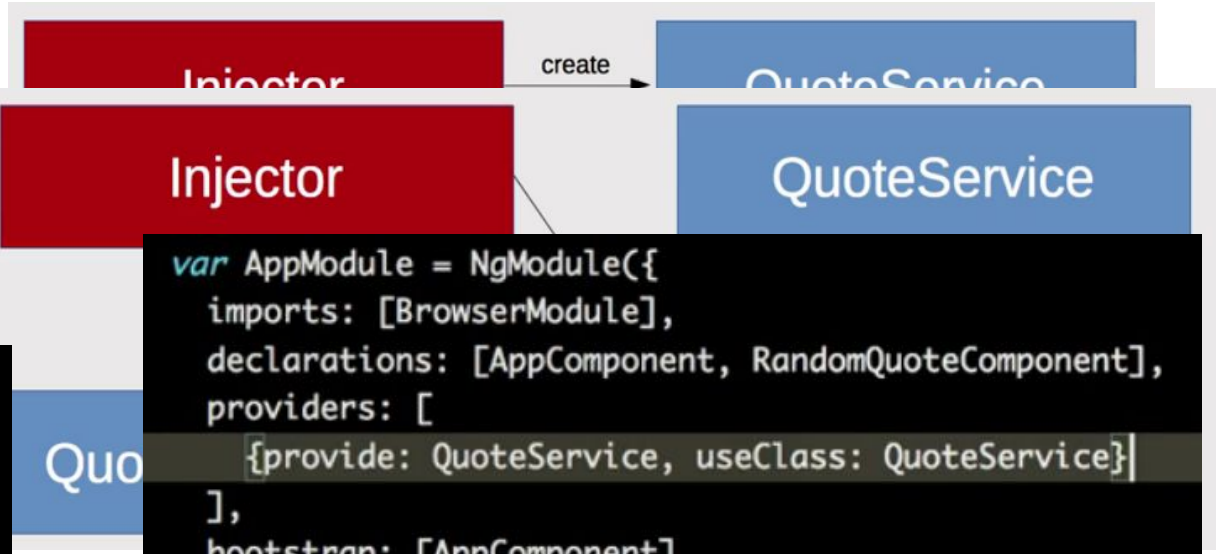
```
var AppModule = NgModule({
  imports: [BrowserModule],
  declarations: [AppComponent, RandomQuoteComponent],
  providers: [QuoteService],
  bootstrap: [AppComponent]
})
.Class({
  constructor: function() { }
});
```

Dependency Injection

Reutilização

Manutenção

Teste




```
var MockQuoteService = Class({
  constructor: function() { },
  getRandomQuote: function() {
    return {
      line: 'A mock quote.',
      author: 'Mock Author'
    };
  }
});
```

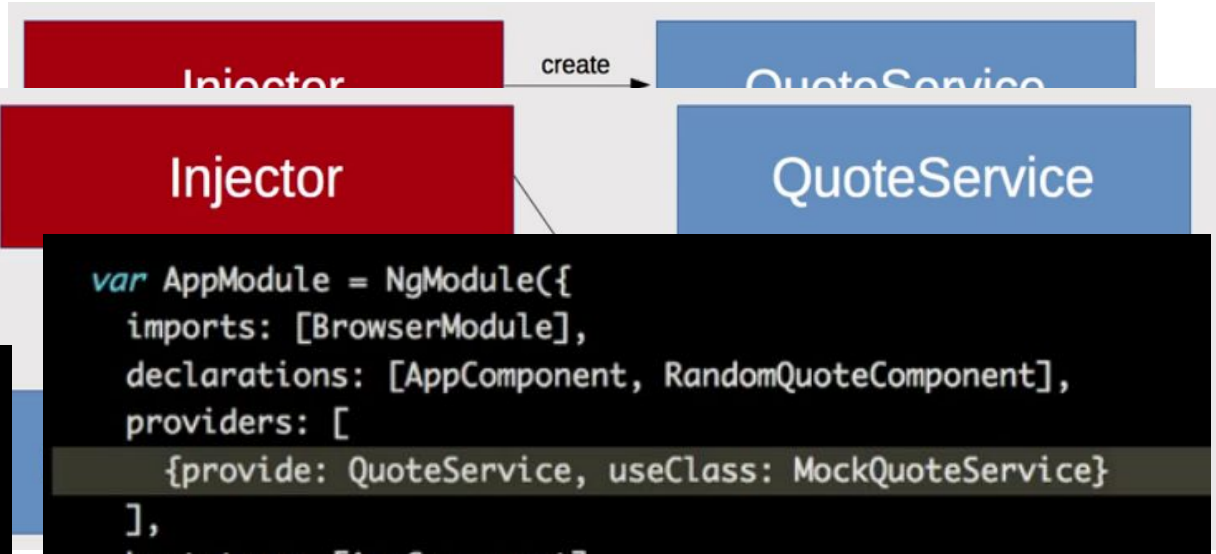
```
var AppModule = NgModule({
  imports: [BrowserModule],
  declarations: [AppComponent, RandomQuoteComponent],
  providers: [
    {provide: QuoteService, useClass: QuoteService}
  ],
  bootstrap: [AppComponent]
})
.Class({
  constructor: function() { }
});
```

Dependency Injection

Reutilização

Manutenção

Teste




```
var MockQuoteService = Class({
  constructor: function() { },
  getRandomQuote: function() {
    return {
      line: 'A mock quote.',
      author: 'Mock Author'
    };
  }
});
```

```
var AppModule = NgModule({
  imports: [BrowserModule],
  declarations: [AppComponent, RandomQuoteComponent],
  providers: [
    {provide: QuoteService, useClass: MockQuoteService}
  ],
  bootstrap: [AppComponent]
})
.Class({
  constructor: function() { }
});
```

Change Detection
github random-quote3/app.js

```
var QuoteService = Class({
    constructor: function QuoteService() {
    this.quotes = sampleQuotes;
    },
    getRandomQuote: function() {
    var randomIndex = Math.floor(Math.random() * this.quotes.length);
    return this.quotes[randomIndex];
    },
    generateRandomQuotes: function(delay, callback) {
    var self = this;
    callback(this.getRandomQuote());
    setInterval(function() {
    callback(self.getRandomQuote());
    }, delay);
    }
  });
```

Change Detection
github random-quote3/app.js

```
  var RandomQuoteComponent = Component({
    selector: 'random-quote',
    template: '<p><em>{{quote.line}}</em> - {{quote.author}}</p>'
  })
  .Class({
    constructor: [QuoteService, function
RandomQuoteComponent(quoteService) {
    var self = this;
    quoteService.generateRandomQuotes(2000, function(quote) {
    self.quote = quote;
    });
    }]
  });
```

File Modules
random-quote4/app/quote.service.js

```
 (function(app) {
  var Class = ng.core.Class;
  app.QuoteService = Class({
    constructor: function QuoteService() {     this.quotes =
sampleQuotes;},
    getRandomQuote: function() {
    var randomIndex = Math.floor(Math.random() * this.quotes.length);
    return this.quotes[randomIndex];},
    generateRandomQuotes: function(delay, callback) {
    var self = this;      callback(this.getRandomQuote());
    setInterval(function() {  callback(self.getRandomQuote());}, delay);
    }  });
  var sampleQuotes = [...];

})(window.app || (window.app = {}));
```

File Modules

...quote4/app/random-quote.component.js

```
(function(app) {
  var Component = ng.core.Component;
  var QuoteService = app.QuoteService;
  app.RandomQuoteComponent = Component({
    selector: 'random-quote',
    template: '<p><em>{{quote.line}}</em> - {{quote.author}}</p>'
  })
  .Class({constructor: [QuoteService, function
RandomQuoteComponent(quoteService) {       var self = this;
    quoteService.generateRandomQuotes(2000, function(quote) {
    self.quote = quote;
    });
    }]
  });

})(window.app || (window.app = {}));
```

File Modules

...quote4/app/app.component.js

```
(function(app) {
  var Component = ng.core.Component;

  app.AppComponent = Component({
    selector: 'my-app',
    template:
    '<h1>Random Quote</h1>' +
    '<random-quote></random-quote>'
  })
  .Class({
    constructor: function AppComponent() { }
  });

})(window.app || (window.app = {}));
```

File Modules

...quote4/app/app.module.js

```
(function(app) {
  var NgModule = ng.core.NgModule;
  var BrowserModule = ng.platformBrowser.BrowserModule;
  var QuoteService = app.QuoteService;
  var RandomQuoteComponent = app.RandomQuoteComponent;
  var AppComponent = app.AppComponent;
  app.AppModule = NgModule({  imports: [BrowserModule],
    declarations: [AppComponent, RandomQuoteComponent],
    providers: [QuoteService], bootstrap: [AppComponent]
  })
  .Class({
    constructor: function() { }
  });
})(window.app || (window.app = {}));
```

File Modules

...quote4/app/main.js

```
(function(app) {
  var platformBrowserDynamic =
ng.platformBrowserDynamic.platformBrowserDynamic;
  var AppModule = app.AppModule;

  platformBrowserDynamic().bootstrapModule(AppModule);

})(window.app || (window.app = {}));
```

File Modules
...quote4/index.html

```
  ....
  <body>
    <my-app>Loading...</my-app>
    <script src="app/quote.service.js"></script>
    <script src="app/random-quote.component.js"></script>
    <script src="app/app.component.js"></script>
    <script src="app/app.module.js"></script>
    <script src="app/main.js"></script>
  </body>
</html>
```

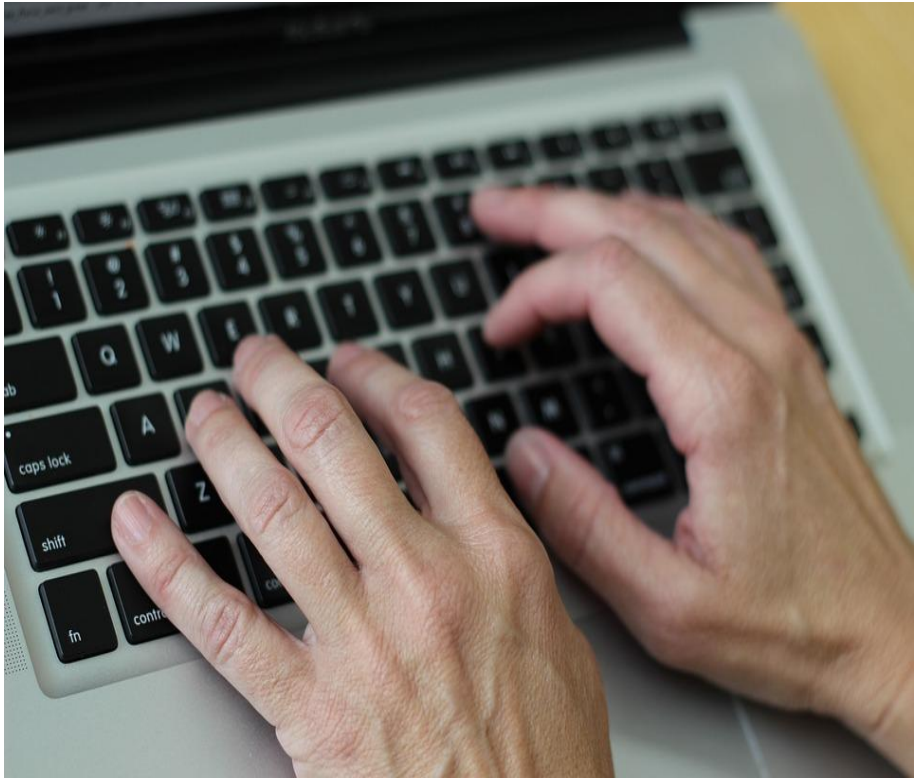
Mãos à Obra!!

Exercício 1:

Crie um componente Angular ES5 para exibir a hora corrente de 1 em 1 segundo.

Observações:

1) Deve ser colocado em um repositório do GitHub;
2) Pode ser Feito em Duplas mas o nome dos integrantes deve constar no index.html
3) Crie uma aplicação completa para demonstrar seu componente



Observação - Ferramentas

ECMAScript 6
JS

ECMAScript 6

JS

		Compilers/polyfills					Desktop browsers																Servers/runtimes									Mobile					
	94%	56%	71%	48%	59%	18%	5%	11%	93%	96%	86%	94%	97%	97%	97%	97%	97%	97%	97%	99%	99%	99%	99%	4%	66%	96%	59%	52%	97%	97%	21%	23%	5%	10%	25%	54%	99%
Feature name	Current browser	Traceur	Babel + core-js[2]	Closure	Type-Script + core-js	es6-shim	KQ 4.14[3]	IE 11	Edge 14[4]	Edge 15[4]	FF 45 ESR	FF 52 ESR	FF 53	FF 54 Beta	FF 55 Nightly	CH 58, OP 45[1]	CH 59, OP 46[1]	CH 60, OP 47[1]	SF 10	SF 10.1	SF TP	WK	PJS	Echo JS	XS6	JXA	Node 4[5]	Node >=6.5 <7[5]	Node >=7.6 <8[5]	DUK 20	DUK 21	AN 4.4	AN 5.0	AN 5.1	iOS 9	iOS 10.0-10.3
imisation																																					
roper tail calls (tail call optimisation)	0/2	0/2	0/2	0/2	0/2	0/2	0/2	0/2	0/2	0/2	0/2	0/2	0/2	0/2	0/2	0/2	0/2	0/2	2/2	2/2	2/2	2/2	0/2	0/2	2/2	0/2	0/2	0/2	0/2	2/2	2/2	0/2	0/2	0/2	0/2	2/2
tax																																				
lefault function parameters	6/7	4/7	4/7	4/7	5/7	0/7	0/7	0/7	7/7	7/7	4/7	6/7	7/7	7/7	7/7	7/7	7/7	7/7	7/7	7/7	7/7	7/7	0/7	4/7	7/7	0/7	0/7	7/7	7/7	0/7	0/7	0/7	0/7	0/7	0/7	7/7
est parameters	5/5	4/5	3/5	2/5	4/5	0/5	0/5	0/5	5/5	5/5	5/5	5/5	5/5	5/5	5/5	5/5	5/5	5/5	5/5	5/5	5/5	5/5	0/5	3/5	5/5	0/5	0/5	5/5	5/5	0/5	0/5	0/5	0/5	0/5	0/5	5/5
pread (...) operator	15/15	15/15	13/15	12/15	4/15	0/15	0/15	0/15	15/15	15/15	15/15	15/15	15/15	15/15	15/15	15/15	15/15	15/15	15/15	15/15	15/15	15/15	0/15	10/15	15/15	11/15	0/15	15/15	15/15	0/15	0/15	0/15	0/15	0/15	9/15	15/15
bject literal extensions	6/6	6/6	6/6	4/6	6/6	0/6	0/6	0/6	6/6	6/6	6/6	6/6	6/6	6/6	6/6	6/6	6/6	6/6	6/6	6/6	6/6	6/6	0/6	5/6	6/6	5/6	6/6	6/6	6/6	4/6	4/6	0/6	0/6	0/6	5/6	6/6
or..of loops	7/9	9/9	9/9	6/9	3/9	0/9	0/9	0/9	7/9	9/9	7/9	7/9	9/9	9/9	9/9	9/9	9/9	9/9	9/9	9/9	9/9	9/9	0/9	7/9	9/9	8/9	7/9	9/9	9/9	0/9	0/9	0/9	0/9	7/9	8/9	9/9
ctal and binary literals	4/4	2/4	4/4	4/4	4/4	2/4	0/4	0/4	4/4	4/4	4/4	4/4	4/4	4/4	4/4	4/4	4/4	4/4	4/4	4/4	4/4	4/4	0/4	2/4	4/4	4/4	4/4	4/4	4/4	4/4	4/4	0/4	0/4	0/4	4/4	4/4
emplate literals	5/5	4/5	4/5	3/5	3/5	0/5	0/5	0/5	5/5	5/5	5/5	5/5	5/5	5/5	5/5	5/5	5/5	5/5	5/5	5/5	5/5	5/5	0/5	4/5	5/5	5/5	5/5	5/5	5/5	0/5	0/5	0/5	0/5	0/5	5/5	5/5
egExp "y" and "u" flags	5/5	3/5	3/5	0/5	0/5	0/5	0/5	0/5	5/5	5/5	2/5	5/5	5/5	5/5	5/5	5/5	5/5	5/5	5/5	5/5	5/5	5/5	0/5	2/5	2/5	0/5	0/5	5/5	5/5	0/5	0/5	0/5	0/5	0/5	0/5	5/5
lestructuring, declarations	21/22	20/22	21/22	19/22	15/22	0/22	0/22	0/22	21/22	22/22	19/22	21/22	22/22	22/22	22/22	22/22	22/22	22/22	22/22	22/22	22/22	22/22	0/22	12/22	21/22	19/22	0/22	22/22	22/22	0/22	0/22	0/22	0/22	0/22	19/22	22/22
lestructuring, assignment	23/24	23/24	24/24	17/24	19/24	0/24	0/24	0/24	23/24	24/24	21/24	23/24	24/24	24/24	24/24	24/24	24/24	24/24	24/24	24/24	24/24	24/24	0/24	14/24	24/24	21/24	0/24	24/24	24/24	0/24	0/24	0/24	0/24	0/24	21/24	24/24
lestructuring, parameters	20/23	19/23	20/23	18/23	15/23	0/23	0/23	0/23	22/23	23/23	18/23	20/23	23/23	23/23	23/23	23/23	23/23	23/23	23/23	23/23	23/23	23/23	0/23	12/23	23/23	18/23	0/23	23/23	23/23	0/23	0/23	0/23	0/23	0/23	18/23	23/23
Unicode code point escapes	1/2	1/2	1/2	1/2	1/2	0/2	0/2	0/2	2/2	2/2	1/2	1/2	2/2	2/2	2/2	2/2	2/2	2/2	2/2	2/2	2/2	2/2	0/2	2/2	2/2	2/2	2/2	2/2	2/2	2/2	2/2	0/2	0/2	0/2	2/2	2/2

ECMAScript 6

Babel e NPM

```
> npm install

> npm run build

> npm run serve
```

```
app.AppComponent = Component({
  selector: 'my-app',
  template: `
  <h1>Random Quote</h1>
  <random-quote></random-quote>
  `
})
```

```
  .Class({
    constructor: [QuoteService, function
RandomQuoteComponent(quoteService) {
    quoteService.generateRandomQuotes(2000,
                                quote => this.quote = quote);
    }]
  });


    generateRandomQuotes: function(delay, callback) {
    callback(this.getRandomQuote());
    setTimeout(
       () => callback(this.getRandomQuote()), delay);
    }
```

CONST e LET

Mudança no escopo

Igual a outras linguagens

Recomenda-se CONST no lugar do VAR ao menos que a variável seja alterada

Classes e Decorators

```
(function(app) {
  var Component = ng.core.Component;

  @Component({
    selector: 'my-app',
    template: `
    <h1>Random Quote</h1>
    <random-quote></random-quote>
    `
  })
  class AppComponent { }

  app.AppComponent = AppComponent;

})(window.app || (window.app = {}));
```

```
{
  "presets": ["es2015"],
  "plugins": [
      "angular2-annotations",
      "transform-decorators-legacy"
  ]
}
```

Modules

```
import { Component, Inject } from '@angular/core';
import { QuoteService } from './quote.service';

@Component({
  selector: 'random-quote',
  template: '<p><em>{{quote.line}}</em> - {{quote.author}}</p>'
})
export class RandomQuoteComponent {

  constructor(@Inject(QuoteService) quoteService) {
    quoteService.generateRandomQuotes(2000, quote => this.quote =
quote);
  }

}
```

SystemJS

```
<html>
  <head>
    <title>Angular 2 - Random Quote</title>
    <meta name="viewport" content="width=device-width, initial-scale=1">

    <script src="node_modules/core-js/client/shim.min.js"></script>
    <script src="node_modules/zone.js/dist/zone.js"></script>
    <script src="node_modules/systemjs/dist/system.src.js"></script>
    <script src="systemjs.config.js"></script>
    <script>
    System.import('app').catch(function(err) { console.error(err);  });
    </script>
  </head>
  <body>
    <my-app>Loading...</my-app>
  </body>
</html>
```

TypeScript

```
{
  "name": "random-quote",   "version": "1.1.1",
  "scripts": {
    "build": "tsc",
    "serve": "concurrently \"tsc -w\" lite-server"
  },
  "dependencies": {
     ...
  },
  "devDependencies": {
    "@types/core-js": "0.9.34",
    "concurrently": "3.0.0",
    "lite-server": "2.2.2",
    "typescript": "2.0.3"
  }
}
```

```
{
 "compilerOptions": {
      "rootDir": "src",
      "outDir": "app",
      "target": "es5",
      "module": "system",
      "moduleResolution": "node",
      "emitDecoratorMetadata": true,
      "experimentalDecorators": true,
      "types": [
      "core-js"
      ]
 },
 "exclude": [
      "node_modules"
 ],
 "compileOnSave": false
}
```

TypeScript

```
export class QuoteService {

  quotes = sampleQuotes;  // <- Declarar atributos

  getRandomQuote() {
    const randomIndex = Math.floor(Math.random() * this.quotes.length);
    return this.quotes[randomIndex];
  }

  generateRandomQuotes(delay, callback) {
    callback(this.getRandomQuote());
    setTimeout(() => callback(this.getRandomQuote()), delay);
  }

}
```

TypeScript

```
import { Component } from '@angular/core';
import { Quote } from './quote.model';
import { QuoteService } from './quote.service';

@Component({
  selector: 'random-quote',
  template: '<p><em>{{quote.line}}</em> - {{quote.author}}</p>'
})
export class RandomQuoteComponent {
  quote: Quote;
  constructor(quoteService: QuoteService) {
    quoteService.generateRandomQuotes(2000, quote => this.quote =
quote);
  }

}
```

TypeScript

```
export interface Quote {
  line: string;
  author: string;
}
```

TypeScript

```
import { Quote } from './quote.model';

export class QuoteService {

  quotes: Quote[] = sampleQuotes;

  getRandomQuote(): Quote {
    const randomIndex = Math.floor(Math.random() * this.quotes.length);
    return this.quotes[randomIndex];
  }

  generateRandomQuotes(delay: number, callback: (quote: Quote)
=> void) {
    callback(this.getRandomQuote());
    setTimeout(() => callback(this.getRandomQuote()), delay);
  }
```

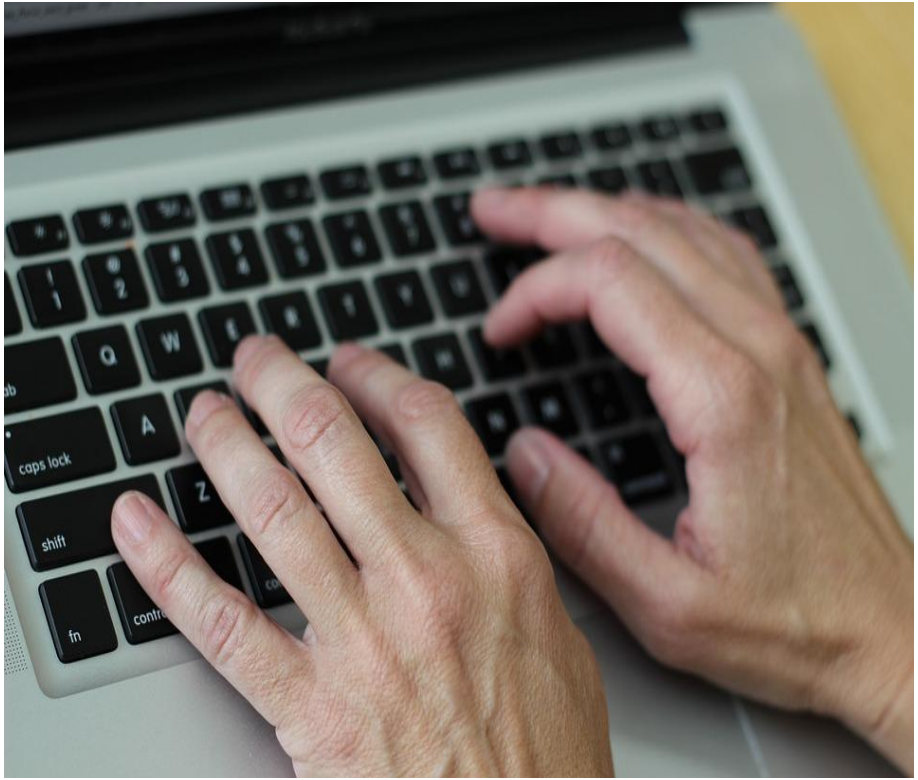
TypeScript

```
import { Quote } from './quote.model';
export class QuoteService {
  private quotes: Quote[] = sampleQuotes;
  private getRandomQuote(): Quote {
    const randomIndex = Math.floor(Math.random() *
this.quotes.length);
    return this.quotes[randomIndex];
  }
  generateRandomQuotes(delay: number, callback: (quote:
Quote) => void) {
    callback(this.getRandomQuote());
    setTimeout(() => callback(this.getRandomQuote()),
delay);  }
```

Mãos à Obra!!

Exercício 2:
Altere o projeto do Exercío 1 para TypeScript
Observações:

1) Deve ser colocado em um repositório do GitHub;
2) Pode ser Feito em Duplas mas o nome dos integrantes deve constar no index.html



Angular CLI

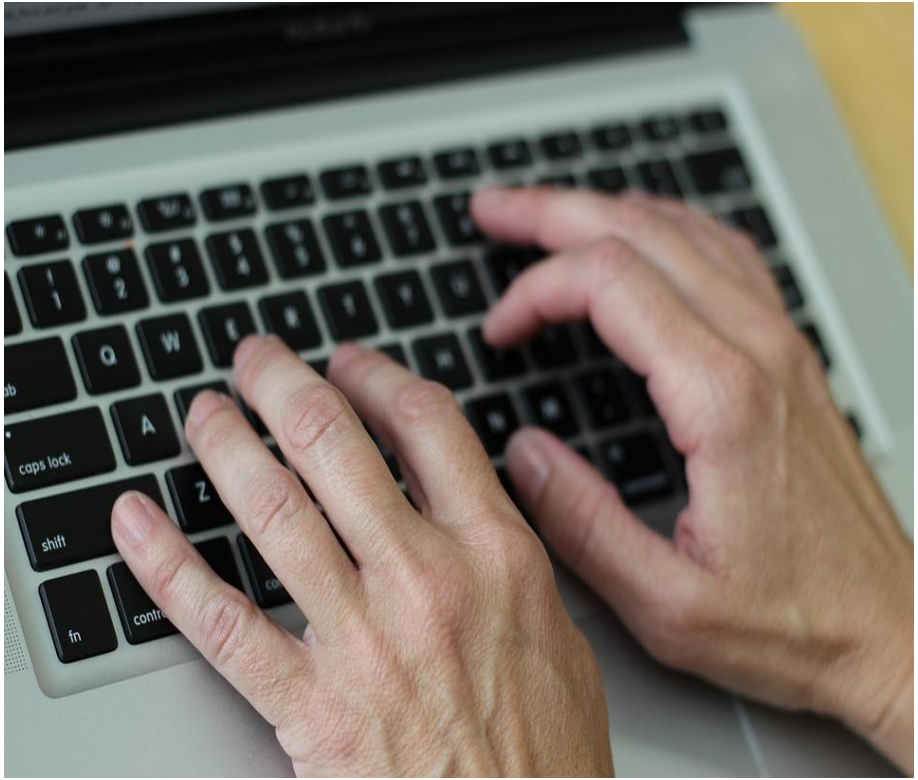
https://cli.angular.io/

```
npm install -g @angular/cli
ng new produto
cd produto
ng serve
```

Mãos à Obra!!

Exercício 3:

Crie um cadastro com pelo menos 3 atributos em Angular e uma API Node.

Observações:

1) Deve ser colocado em um repositório do GitHub;
2) Pode ser Feito em Duplas mas o nome dos integrantes deve constar no index.html