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

Arquiteto da Gumga S/A e Sócio da ImpactIt

Agenda

Conceitos

ES5

Dependency Injection

Change Detection

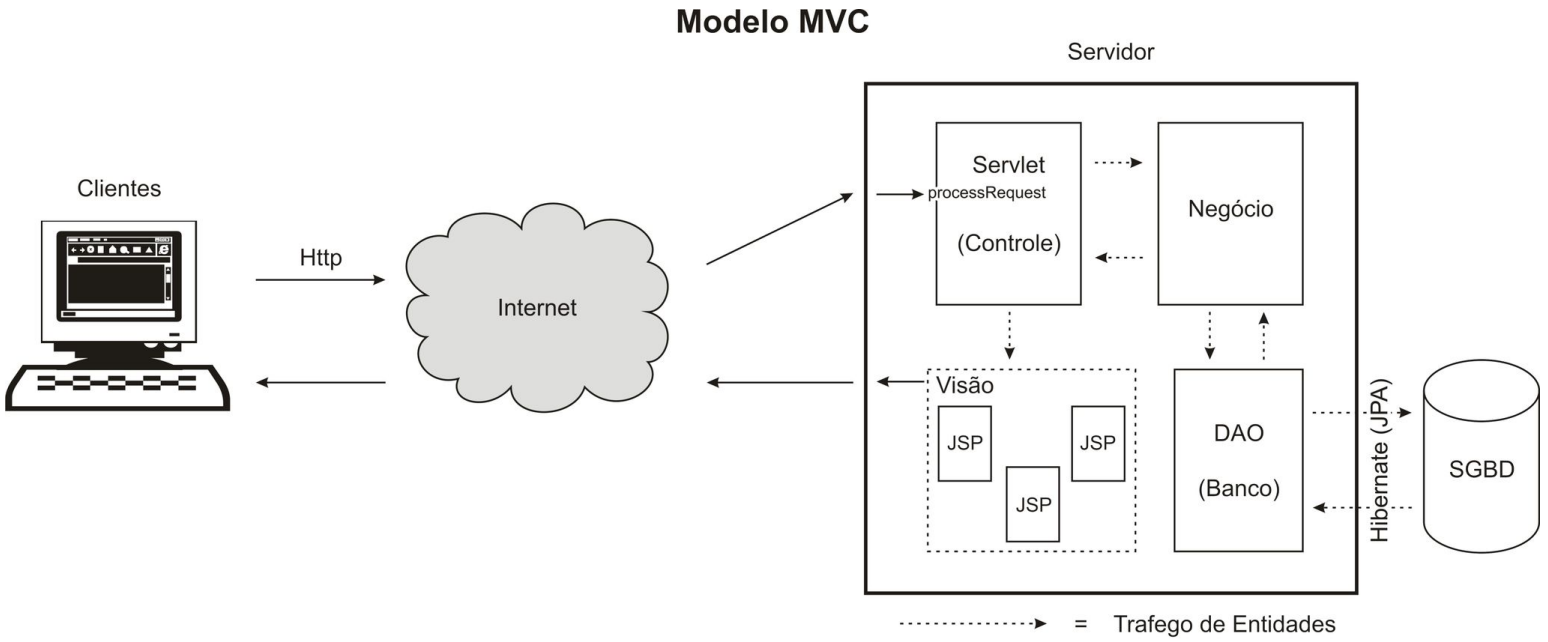
ES6

TypeScript

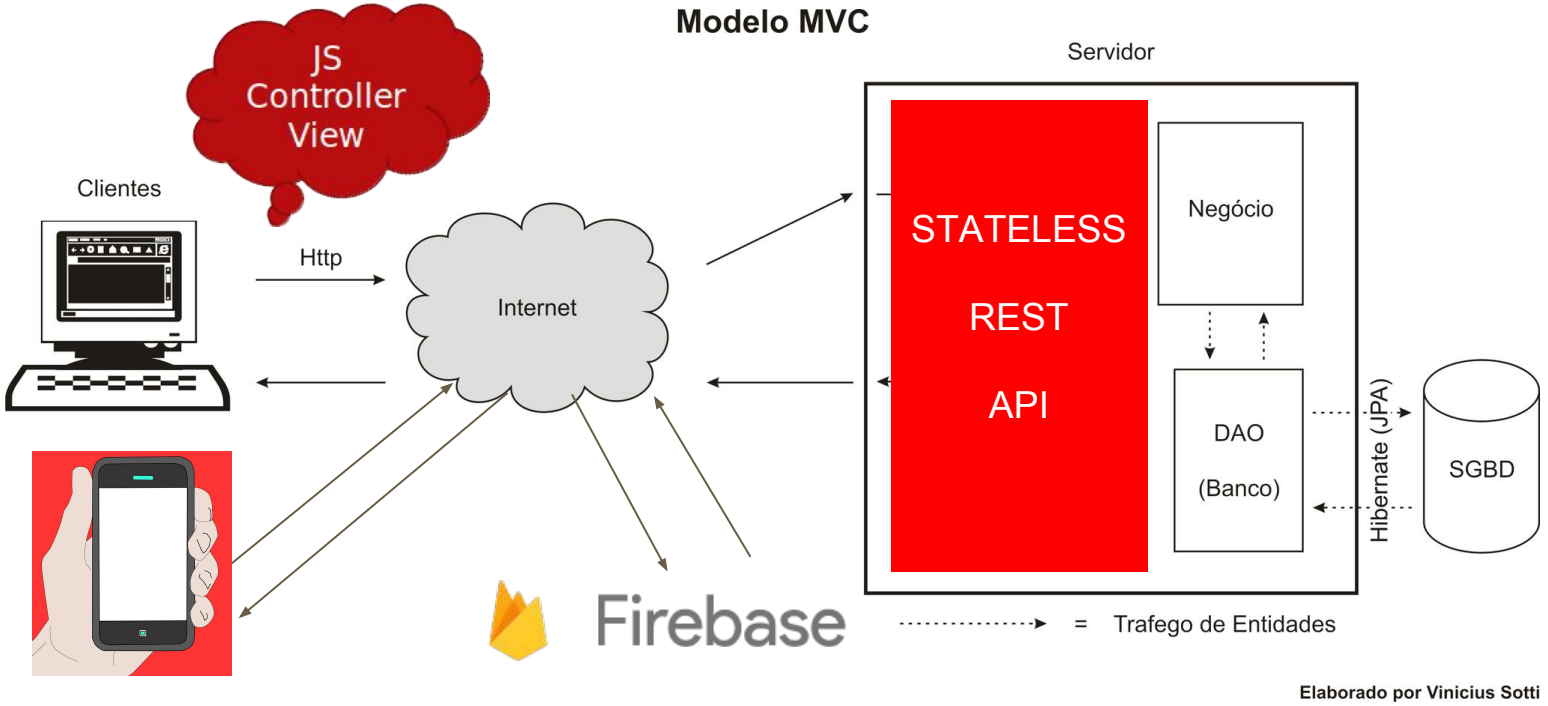
Biding

Pipes

Motivação: Aplicações Desacopladas



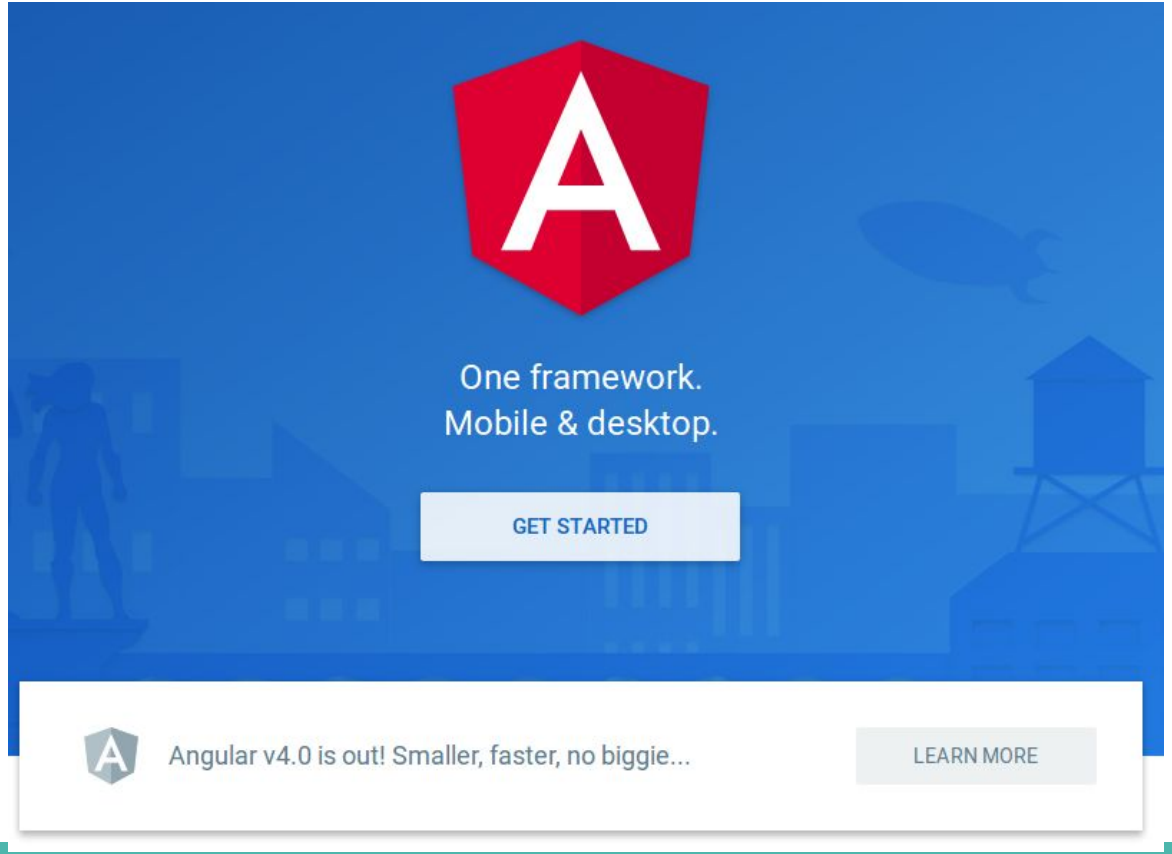
Motivação: Aplicações Desacopladas



Tecnologias

Ferramentas

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

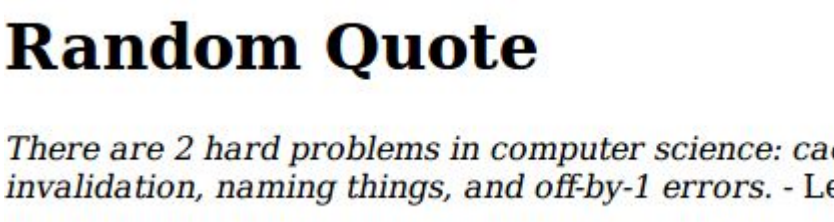
hello/app.js

```
(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

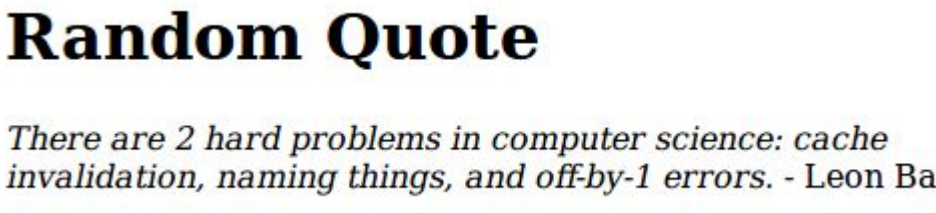



```
(function() {
  var Component = ng.core.Component;
  var NgModule = ng.core.NgModule;
  var BrowserModule = ng.platformBrowser.BrowserModule;
  var platformBrowserDynamic =
ng.platformBrowserDynamic.platformBrowserDynamic;

  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

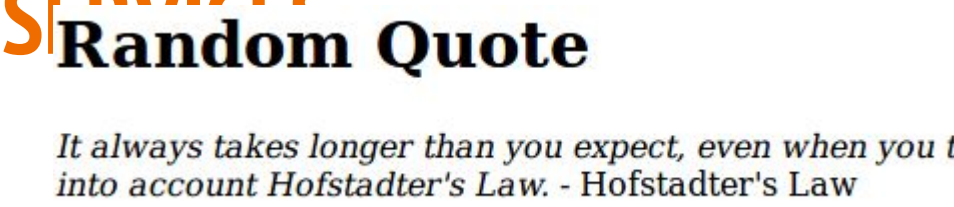
```
  platformBrowserDynamic().bootstrapModule(AppModule);

  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

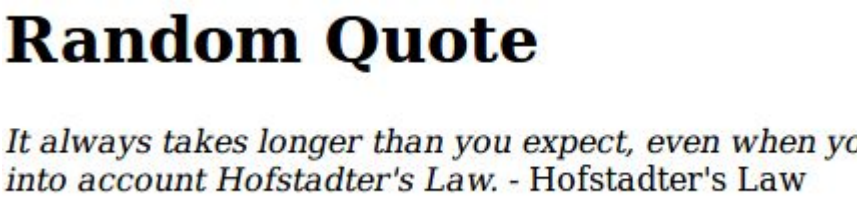
random-quote2/app.js SERVICES




```
(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];
    }
  });
```

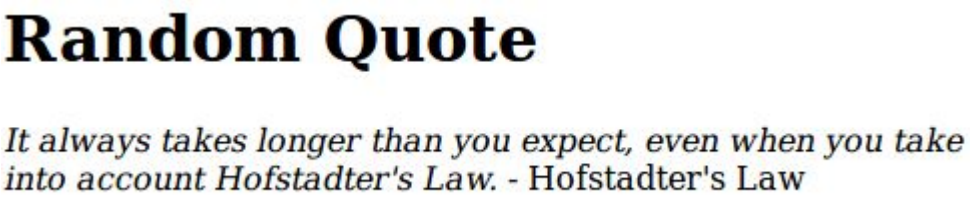
github.com/munifgebara/angular4 random-quote2/app.js SERVICES

```
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 = [ ...  ];

})();
```

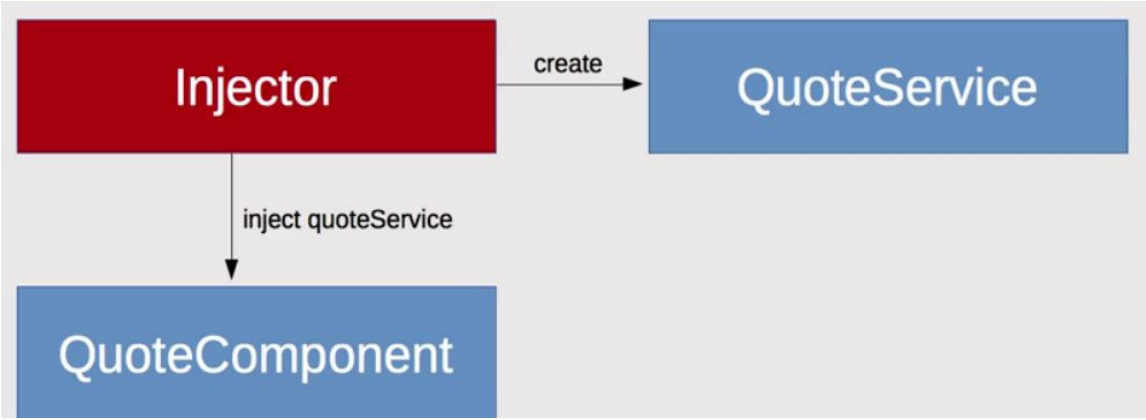


Dependency Injection

Reutilização

Manutenção

Teste

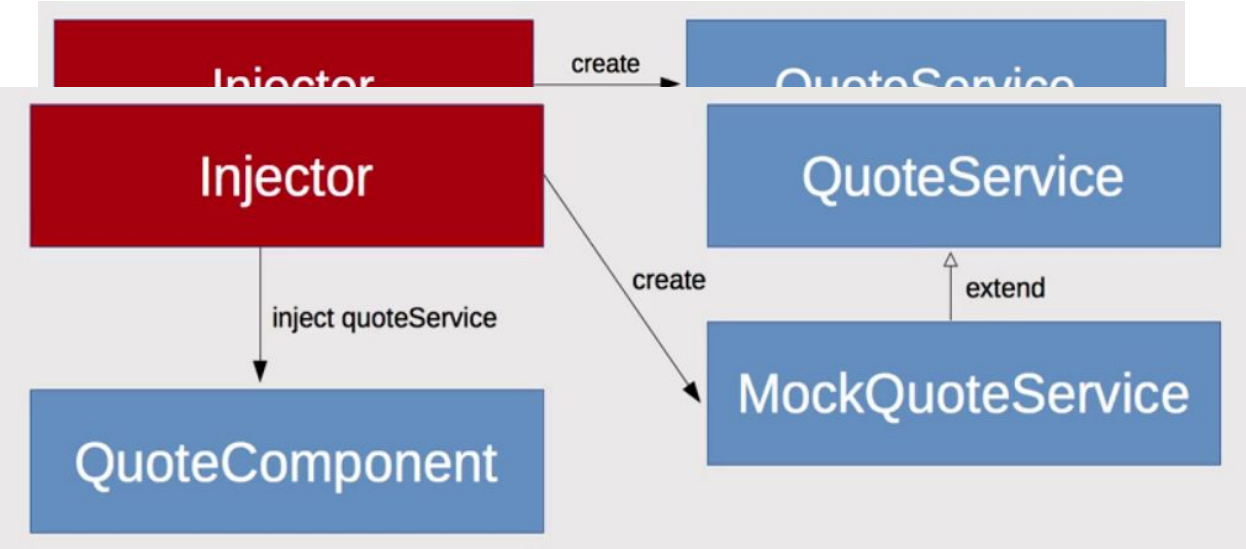


Dependency Injection

Reutilização

Manutenção

Teste

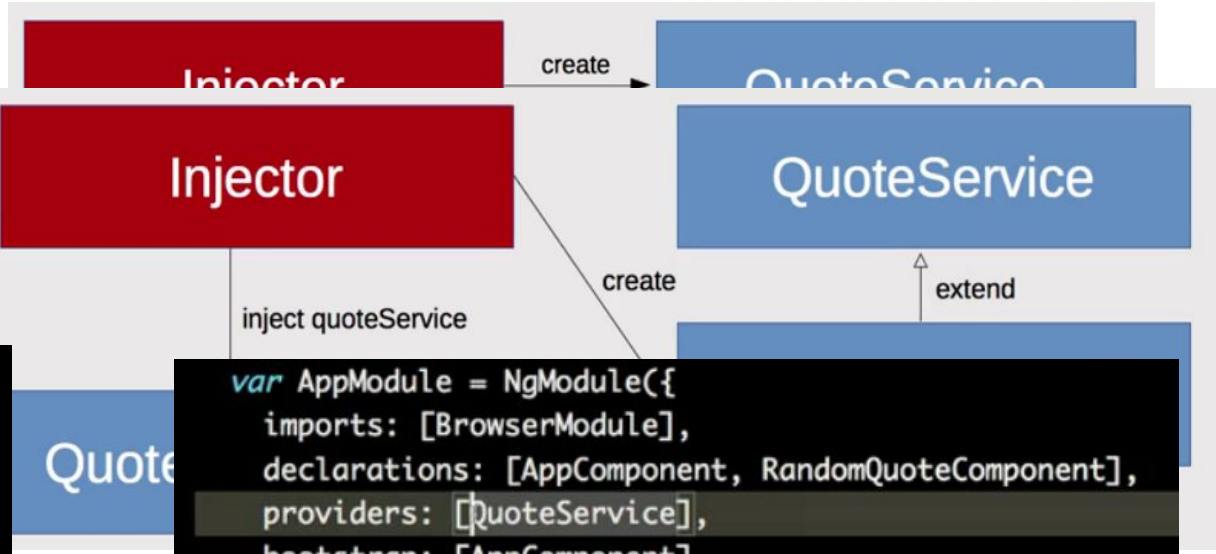


Dependency Injection

Reutilização

Manutenção

Teste




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

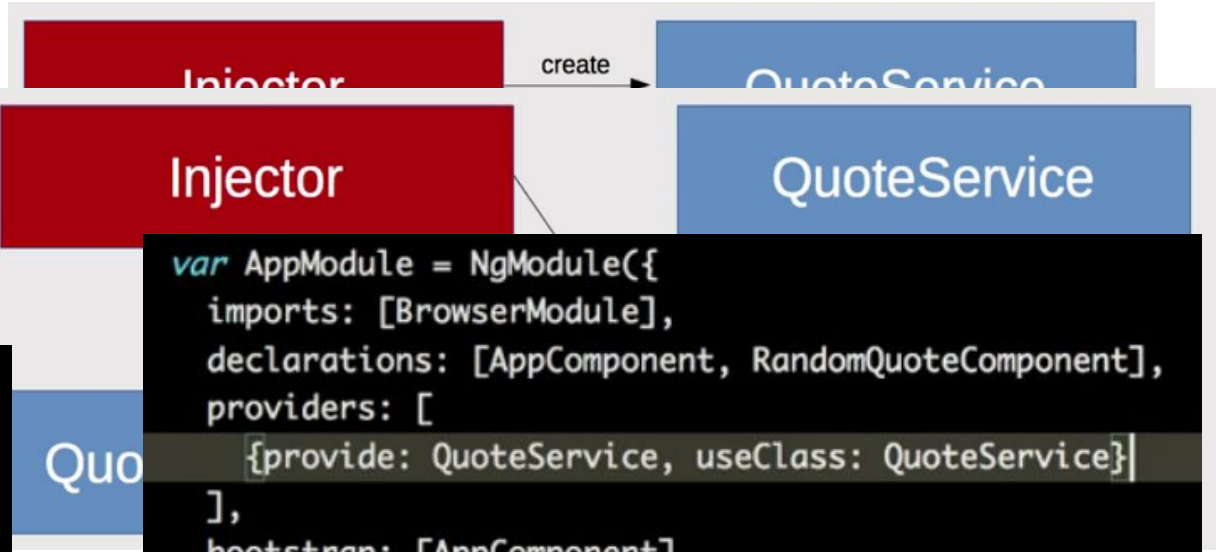
```
var AppModule = NgModule({
  imports: [BrowserModule],
  declarations: [AppComponent, RandomQuoteComponent],
  providers: [QuoteService],
  bootstrap: [AppComponent]
})
.Class({
  constructor: function() { }
});
```

Dependency Injection

Reutilização

Manutenção

Teste




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

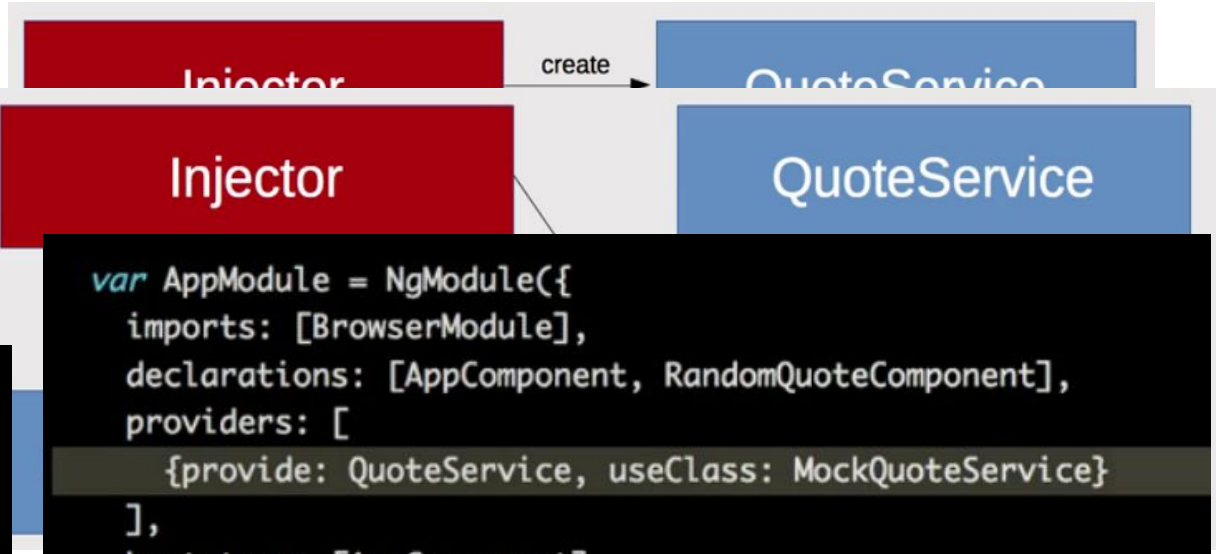
```
var AppModule = NgModule({
  imports: [BrowserModule],
  declarations: [AppComponent, RandomQuoteComponent],
  providers: [
    {provide: QuoteService, useClass: QuoteService}
  ],
  bootstrap: [AppComponent]
})
.Class({
  constructor: function() { }
});
```

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>
```

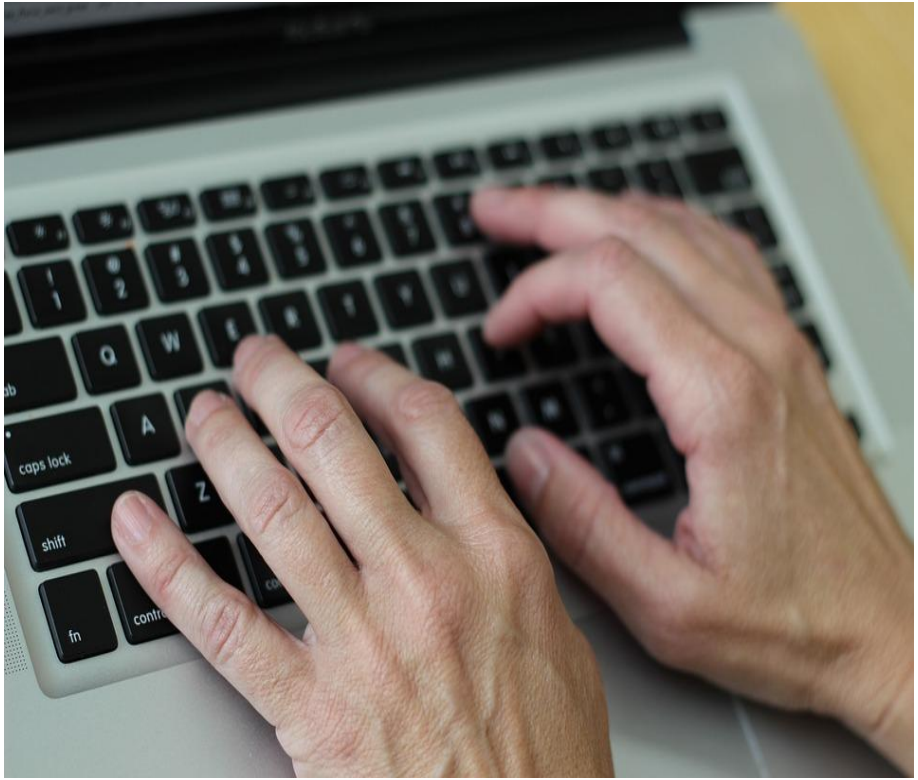
Mãos à Obra!!

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%	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);
  }
```

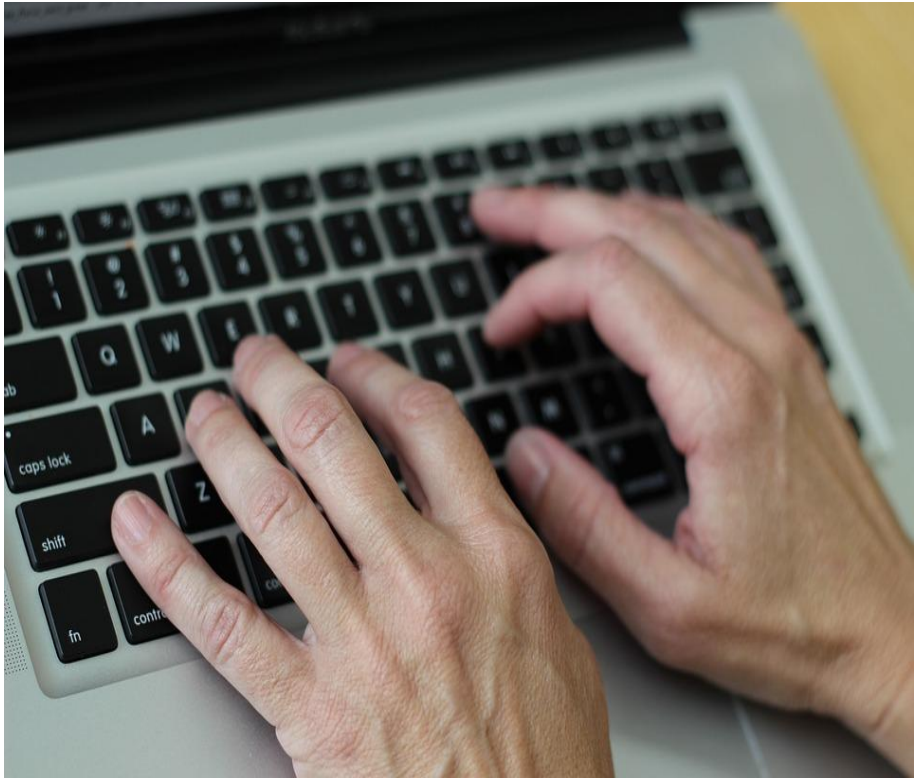
TypeScript

```
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



Two-way Binding (CurrencyC.)

```
@Component({
  selector: 'currency-converter',
  template: `
    Convert: <input type="number" [(ngModel)]="baseAmount"> USD
    <p><strong>{{baseAmount}}</strong> USD =
    <strong>{{targetAmount}}</strong> GBP</p>`,
  styles: [`...`]
})
export class AppComponent {
  exchangeRate = 0.70;
  baseAmount = 1;
  get targetAmount() {
    return this.baseAmount * this.exchangeRate;
  }
}
```

Class Binding

```
@Component({
  selector: 'currency-converter',
  template: `
    <input type="number" [(ngModel)]="baseAmount"
    [class.error]="isInvalid(baseAmount)"> USD
    = <strong>{{targetAmount}}</strong> GBP
  `,
  styles: [`
    input[type=number] {
    width: 10ex;
    text-align: right;
    }
    .error {
    background-color: #ff6666;
    }
  `]
})
```

```
isInvalid(value) {
  return !Number.isFinite(value);
}
```

Service

```
export class ExchangeService {
  supportedCurrencies = ['EUR', 'GBP', 'USD'];
  private exchangeRates = {
    "EUR/GBP": 0.8007,
    "EUR/USD": 1.1397,
    "GBP/EUR": 1.2478,
    "GBP/USD": 1.4225,
    "USD/EUR": 0.8778,
    "USD/GBP": 0.7029
  };
  getExchangeRate(baseCurrency: string, targetCurrency: string) {
    if (baseCurrency === targetCurrency) {
    return 1;
    }
    return this.exchangeRates[baseCurrency +'/'+ targetCurrency];
  }
}
```

Currency Select - template

```
import { Component, EventEmitter, Input, Output } from '@angular/core';
import { ExchangeService } from './exchange.service';

@Component({
  selector: 'currency-select',
  template: `
    <select [ngModel]="selected"
(ngModelChange)="onSelectedChange($event)">
    <option *ngFor="let currency of supportedCurrencies"
[value]="currency">
    {{currency}}
    </option>
    </select>
  `
})
```

Currency Select -class

```
export class CurrencySelectComponent {

  @Input() selected: string;
  @Output() selectedChange = new EventEmitter<string>();

  supportedCurrencies: string[];

  constructor(exchangeService: ExchangeService) {
    this.supportedCurrencies = exchangeService.supportedCurrencies;
  }

  onSelectedChange(selected: string) {
    this.selected = selected;
    this.selectedChange.emit(selected);
  }
}
```

Pipes

Formatam templates

Transformam uma entrada em uma saída

Exemplos de pipes fornecidos com Angular:

```
{{'Teste'| uppercase}}

{{now | date: 'short'}}
```

Pipes Customizados

```
import { Pipe, PipeTransform } from '@angular/core';

@Pipe({name: 'fixed'})
export class FixedPipe implements PipeTransform {

  transform(value: number, digits=2) {
    return value.toFixed(digits);
  }
}
```

Pipes Customizados

```
import { Pipe, PipeTransform } from '@angular/core';

@Pipe({name: 'fixed'})
export class FixedPipe implements PipeTransform {

  transform(value: number, digits=2) {
    return value.toFixed(digits);
  }
}
```

Pipes Customizados

```
import { Pipe, PipeTransform } from '@angular/core';

@Pipe({name: 'fixed'})
export class FixedPipe implements PipeTransform {

  transform(value: number, digits=2) {
    return value.toFixed(digits);
  }
}
```

AppComponet - template

```
@Component({
  selector: 'currency-converter',
  template: `
    <input type="number" [(ngModel)]="baseAmount"
    [class.error]="isInvalid(baseAmount)">
    <currency-select [(selected)]="baseCurrency"></currency-select>
    = <strong>{{targetAmount | fixed:2}}</strong>
    <currency-select [(selected)]="targetCurrency"></currency-select>
    <p *ngIf="isInvalid(baseAmount)">Please enter a valid amount</p>
  `,
  styles: [`
    input[type=number] {
    width: 10ex;
    text-align: right;    }
    .error {
    background-color: #ff6666;
    }
  `] })
```

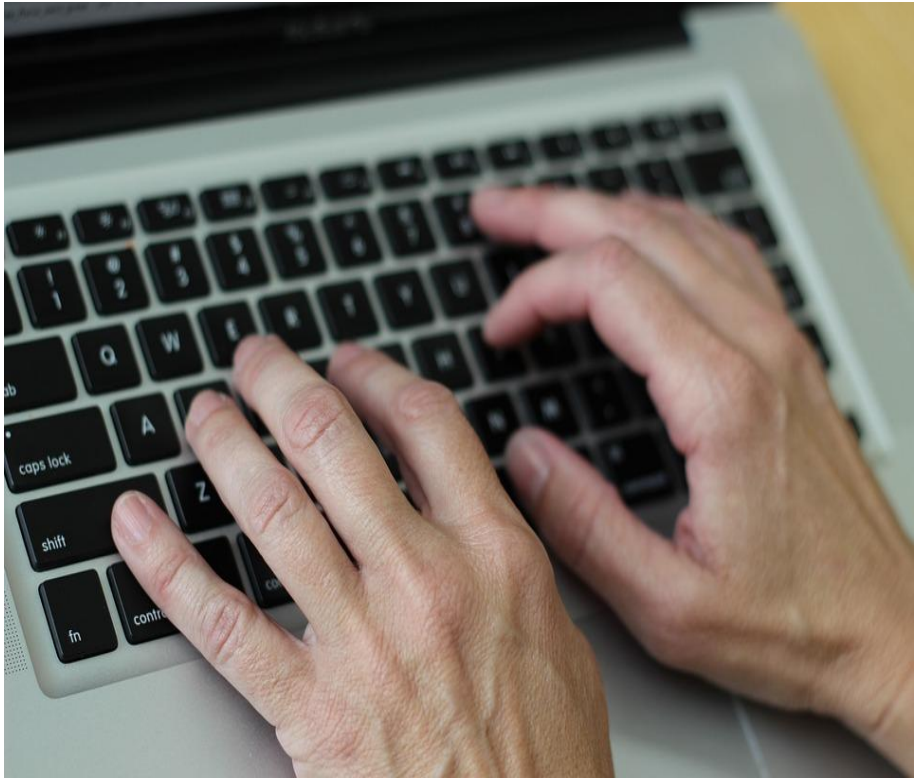
AppComponet - class

```
export class AppComponent {

  baseCurrency = 'USD';
  targetCurrency = 'GBP';
  baseAmount = 1;

  constructor(private exchangeService: ExchangeService) { }

  get targetAmount() {
    const exchangeRate = this.exchangeService
    .getExchangeRate(this.baseCurrency, this.targetCurrency);
    return this.baseAmount * exchangeRate;
  }

  isInvalid(value) {
    return !Number.isFinite(value);
  }
}
```

Mãos à Obra!!

Exercício 3:
Crie um projeto com pelo menos 3 componentes, 1 serviço, colocando em prática os conceitos de hoje
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



FormValidation - signup-form

```
  template: `
    <form (ngSubmit)="onSubmit(form)"  novalidate #form="ngForm">
     ...
    <input type="email" class="form-control"  #emailField="ngModel"
        [(ngModel)]="email" name="email" required pattern=".+@.+">
    <p *ngIf="emailField.touched && emailField.invalid"
        class="alert alert-danger">Please enter a valid email</p>
     ...
    <button type="submit" class="btn btn-primary"
    [disabled]="form.invalid">Sign Up</button>
    </form>
  `,
  styles: [`
    input.ng-dirty.ng-invalid { border: solid red 2px; }
    input.ng-dirty.ng-valid { border: solid green 2px; }
  `]
})
```

Http - quote-of-the-day

main.ts

```
import 'rxjs/add/operator/toPromise';
import { platformBrowserDynamic } from
'@angular/platform-browser-dynamic';
import { AppModule } from './app.module';

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

Http - quote-of-the-day

app.module.ts

```
import { NgModule } from '@angular/core';
import { BrowserModule } from '@angular/platform-browser';
import { HttpModule } from '@angular/http';
import { QuoteService } from './quote.service';
import { AppComponent } from './app.component';

@NgModule({
  imports: [BrowserModule, HttpModule],
  declarations: [AppComponent],
  providers: [QuoteService],
  bootstrap: [AppComponent]
})
export class AppModule { }
```

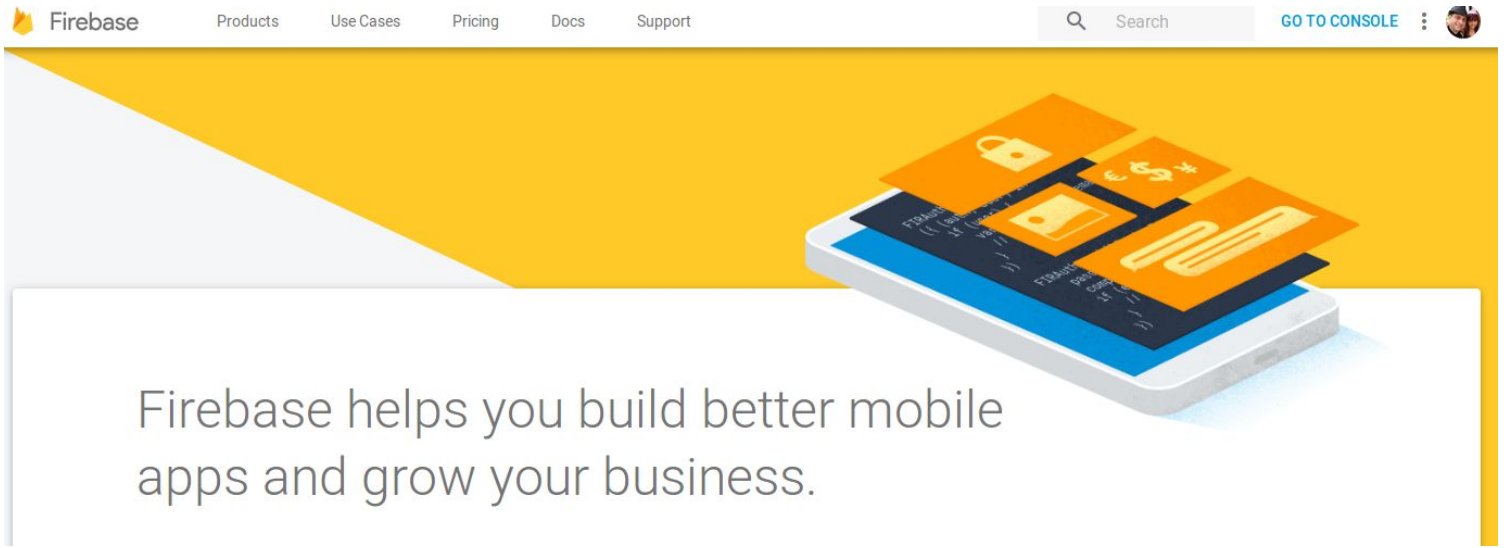
Http - quote-of-the-day

```
import { Injectable } from '@angular/core';
import { Http } from '@angular/http';
import { Quote } from './quote.model';

@Injectable()
export class QuoteService {

  constructor(private http: Http) { }

  getQuoteOfTheDay(): Promise<Quote> {
    return this.http.get('/quote.json').toPromise()
    .then(response => response.json());
  }
}
```

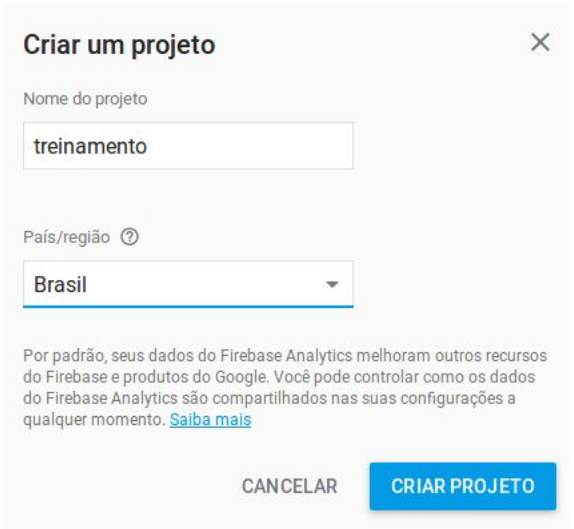
Http - quote-of-the-day

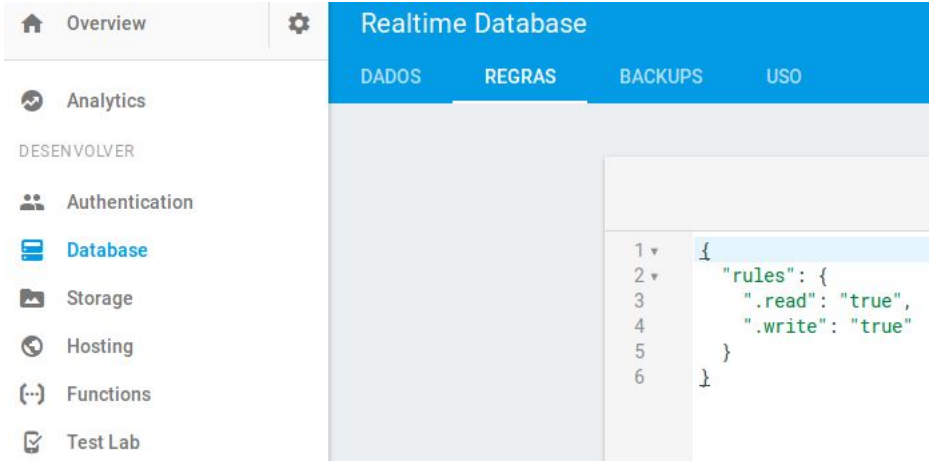
```
@Component({
  selector: 'my-app',
  template: `
    <h1>Quote Of The Day</h1>
    <p><em>{{quote?.line}}</em> - {{quote?.author}}</p>
  `
})
export class AppComponent {
  quote: Quote;
  constructor(quoteService: QuoteService) {
    quoteService.getQuoteOfTheDay()
    .then(quote => this.quote = quote);
  }
}
```

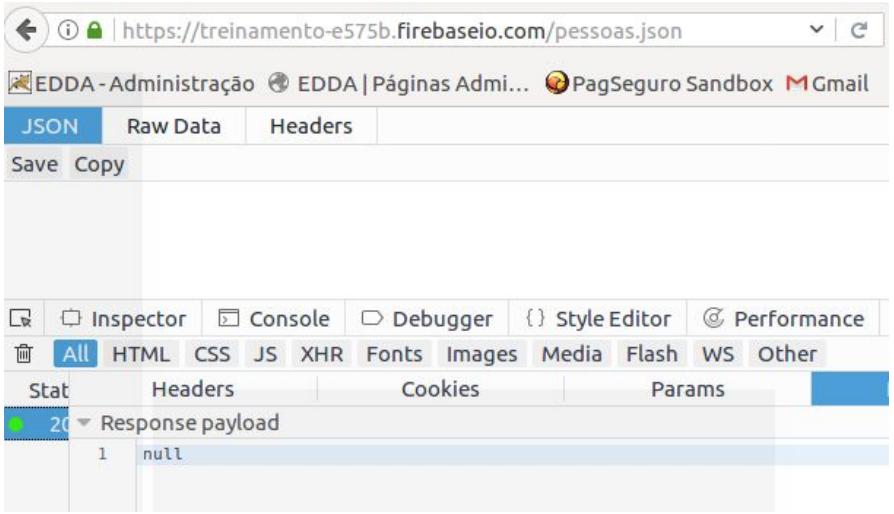
Firebase


Firebase
Products
Use Cases
Pricing
Docs
Support
Search
GO TO CONSOLE
Firebase helps you build better mobile apps and grow your business.

Firebase


Criar um projeto
Nome do projeto
treinamento
País/região
Brasil
Por padrão, seus dados do Firebase Analytics melhoram outros recursos do Firebase e produtos do Google. Você pode controlar como os dados do Firebase Analytics são compartilhados nas suas configurações a qualquer momento. Saiba mais
CANCELAR
CRIAR PROJETO


Overview
Analytics
DESENVOLVER
Authentication
Database
Storage
Hosting
Functions
Test Lab
Realtime Database
DADOS
REGRAS
BACKUPS
USO
{
"rules": {
".read": "true",
".write": "true"
}
}


https://treinamento-e575b.firebaseio.com/pessoas.json
EDDA - Administração
EDDA | Páginas Admi...
PagSeguro Sandbox
Gmail
JSON
Raw Data
Headers
Save
Copy
Inspector
Console
Debugger
Style Editor
Performance
All
HTML
CSS
JS
XHR
Fonts
Images
Media
Flash
WS
Other
Headers
Cookies
Params
Response payload
null

Firebase Testes

{"-KkcrIOh9pwnAAk6mozG":{"nome":"Munif","telefone":"123"}}

New Request

POST https://treinamento-e575b.firebaseio.com/pessoas.json

Request Headers:

```
Host: treinamento-e575b.firebaseio.com
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:53.0) Gecko/20100101 Firefox/53
Accept: application/json
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Content-Type: application/json
Content-Length: 42
Connection: keep-alive
Pragma: no-cache
```

Request Body:

```
{
  "nome":"Munif",
 "telefone":"123"

}
```

Crud - shared-bookmarks

main.ts

```
import 'rxjs/add/operator/toPromise';
import { platformBrowserDynamic } from
'@angular/platform-browser-dynamic';
import { AppModule } from './app.module';

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

Crud - shared-bookmarks

app.module.ts

```
import { NgModule } from '@angular/core';
import { BrowserModule } from '@angular/platform-browser';
import { FormsModule } from '@angular/forms';
import { HttpModule } from '@angular/http';
import { BookmarkService } from './bookmark.service';
import { AppComponent } from './app.component';
import { BookmarkEditComponent } from './bookmark-edit.component';
import { BookmarkListComponent } from './bookmark-list.component';
@NgModule({
  imports: [BrowserModule, FormsModule, HttpModule],
  declarations: [AppComponent, BookmarkEditComponent,
BookmarkListComponent],
  providers: [BookmarkService],
  bootstrap: [AppComponent]
})
export class AppModule { }
```

Crud - shared-bookmarks

```
@Injectable()
export class BookmarkService {
  errorHandler = error => console.error('BookmarkService error', error);
  private baseUrl = 'https://treinamento-e575b.firebaseio.com/';
  private collection = 'bookmarks';
  constructor(private http: Http) { }

  addBookmark(bookmark) {
    const json = JSON.stringify(bookmark);
    return this.http.post(`${this.baseUrl}/${this.collection}.json`, json)
    .toPromise()
    .catch(this.errorHandler);
  }
```

Crud - shared-bookmarks

```
  getBookmarks() {
    return this.http.get(`${this.baseUrl}/${this.collection}.json`)
    .toPromise()
    .then(response => this.convert(response.json()))
    .catch(this.errorHandler);
  }

  removeBookmark(bookmark) {
    return
this.http.delete(`${this.baseUrl}/${this.collection}/${bookmark.id}.json`)
    .toPromise()
    .catch(this.errorHandler);
  }
```

Crud - shared-bookmarks

```
  updateBookmark(bookmark) {
    const json = JSON.stringify({
    title: bookmark.title,
    url: bookmark.url
    });
    return
this.http.patch(`${this.baseUrl}/${this.collection}/${bookmark.id}.json`,
json)
    .toPromise()
    .catch(this.errorHandler);
  }
```

Crud - shared-bookmarks

```
  private convert(parsedResponse) {
     return Object.keys(parsedResponse)
     .map(id => ({
     id: id,
     title: parsedResponse[id].title,
     url: parsedResponse[id].url
     }))
     .sort((a, b) => a.title.localeCompare(b.title));
  }

}
```

Crud - shared-bookmarks

```
import { Component } from '@angular/core';
import { BookmarkService } from './bookmark.service';

@Component({
  selector: 'bookmark-app',
  template: `
    <bookmark-edit [bookmark]="editableBookmark"
    (save)="save($event)" (clear)="clear()"></bookmark-edit>
    <bookmark-list [bookmarks]="bookmarks"
    (edit)="edit($event)" (remove)="remove($event)"></bookmark-list>
  `,
})
```

Crud - shared-bookmarks

```
export class AppComponent {

  bookmarks = [];
  editableBookmark = {};

  constructor(private bookmarkService: BookmarkService) {
    bookmarkService.errorHandler = error =>
    window.alert('Oops! The server request failed.');
    this.reload();
  }

  clear() {
    this.editableBookmark = {};
  }
```

Crud - shared-bookmarks

```
edit(bookmark) {
  this.editableBookmark = Object.assign({}, bookmark);
}
remove(bookmark) {
  this.bookmarkService.removeBookmark(bookmark)
  .then(() => this.reload());
}

private reload() {
  return this.bookmarkService.getBookmarks()
  .then(bookmarks => this.bookmarks = bookmarks);
}
```

Crud - shared-bookmarks

```
  save(bookmark) {
    if (bookmark.id) {
    this.bookmarkService.updateBookmark(bookmark)
    .then(() => this.reload());
    } else {
    this.bookmarkService.addBookmark(bookmark)
    .then(() => this.reload());
    }
    this.clear();
  }

}
```

Crud - shared-bookmarks

```
@Component({
  selector: 'bookmark-list',
  template: `
    <div class="panel panel-default">
    <table class="table table-striped">
    <tr *ngFor="let bookmark of bookmarks">
    <td> <a
[href]="bookmark.url"target="_blank">{{bookmark.title}}</a></td>
    <td><button (click)="onEdit(bookmark)" class="btn btn-primary">
            <span class="glyphicon glyphicon-pencil"></span>
            <span class="hidden-xs">Edit</span></button>
        <button (click)="onRemove(bookmark)" class="btn btn-danger">
            <span class="glyphicon glyphicon-trash"></span>
            <span class="hidden-xs">Delete</span></button>
     </td></tr></table></div>`,})
```

Crud - shared-bookmarks

```
export class BookmarkListComponent {
  @Input() bookmarks = [];
  @Output() edit = new EventEmitter();
  @Output() remove = new EventEmitter();
  onEdit(bookmark) {
    this.edit.emit(bookmark);
  }
  onRemove(bookmark) {
    this.remove.emit(bookmark);
  }
}
```

Crud - shared-bookmarks

```
@Component({selector: 'bookmark-edit',
  template: `
    <div class="panel panel-primary">
    <div class="panel-body">
    <input type="text" [(ngModel)]="bookmark.title"
        placeholder="Title" style="width: 25%;">
    <input type="text" [(ngModel)]="bookmark.url"
        placeholder="URL" style="width: 50%;">
    <button (click)="onSave()" class="btn btn-primary">
        <span class="glyphicon glyphicon-ok"></span>
        <span class="hidden-xs">Save</span>
    </button>
    <button (click)="onClear()" class="btn btn-warning">
        <span class="glyphicon glyphicon-remove"></span>
        <span class="hidden-xs">Clear</span>
    </button></div></div>`,})
```

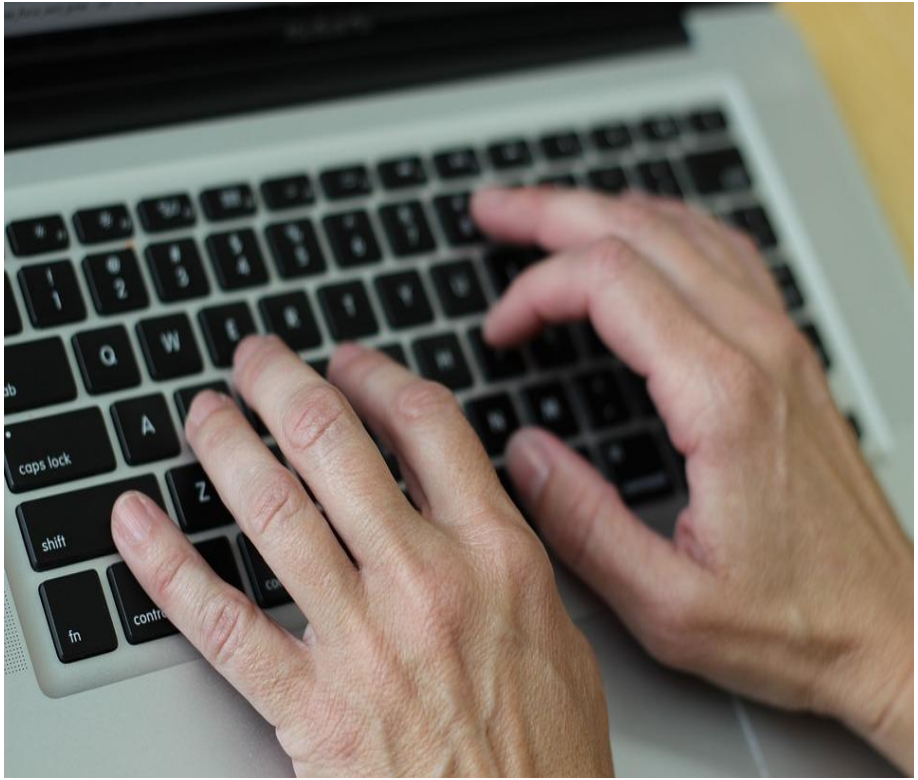
Crud - shared-bookmarks

```
export class BookmarkEditComponent {
  @Input() bookmark = {};
  @Output() clear = new EventEmitter();
  @Output() save = new EventEmitter();
  onClear() {
    this.clear.emit();
  }
  onSave() {
    this.save.emit(this.bookmark);
  }
}
```

Mãos à Obra!!

Exercício 4:
Crie um projeto com CRUD simples de contatos
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 exemplo-app
cd exemplo-app
ng serve
```

Angular Fire - EXEMPLO FIRE

https://github.com/angular/angularfire2/blob/master/docs/1-install-and-setup.md

npm uninstall -g angular-cli
npm uninstall -g @angular/cli
npm cache clean
npm install -g @angular/cli@latest

npm install -g typings
npm install -g typescrip

Angular - CLI

```
ng new compras
ng g module produtos
ng g module listas
ng generate component listas/crud
ng generate component produtos/crud

ng generate service produtos/produtos
ng generate component produtos/edita
ng generate component produtos/lista

ng generate service listas/listas
ng generate component listas/lista
ng generate component listas/edita
```

Angular - Submodules

```
import { ProdutosModule } from './produtos/produtos.module';
import { ListasModule } from './listas/listas.module';

@NgModule({
  declarations: [    AppComponent],
  imports: [BrowserModule,FormsModule,     HttpModule,
    ProdutosModule,  ListasModule ],
  providers: [],
  bootstrap: [AppComponent]
})
export class AppModule { }
```

Angular - Submodules

```
import { ListasService } from './listas.service';
import { EditaComponent } from './edita/edita.component';
import { ListaComponent } from './lista/lista.component';

@NgModule({
  imports: [CommonModule,FormsModule],
  declarations: [CrudComponent, EditaComponent, ListaComponent],
  exports: [CrudComponent],
  providers: [ListasService]
})
export class ListasModule { }
```

Angular - Submodules

```
import { Component, OnInit } from '@angular/core';
import { ListasService } from '../listas.service';

@Component({
  selector: 'app-crud-listas',
  templateUrl: './crud.component.html',
  styleUrls: ['./crud.component.css']
})
```