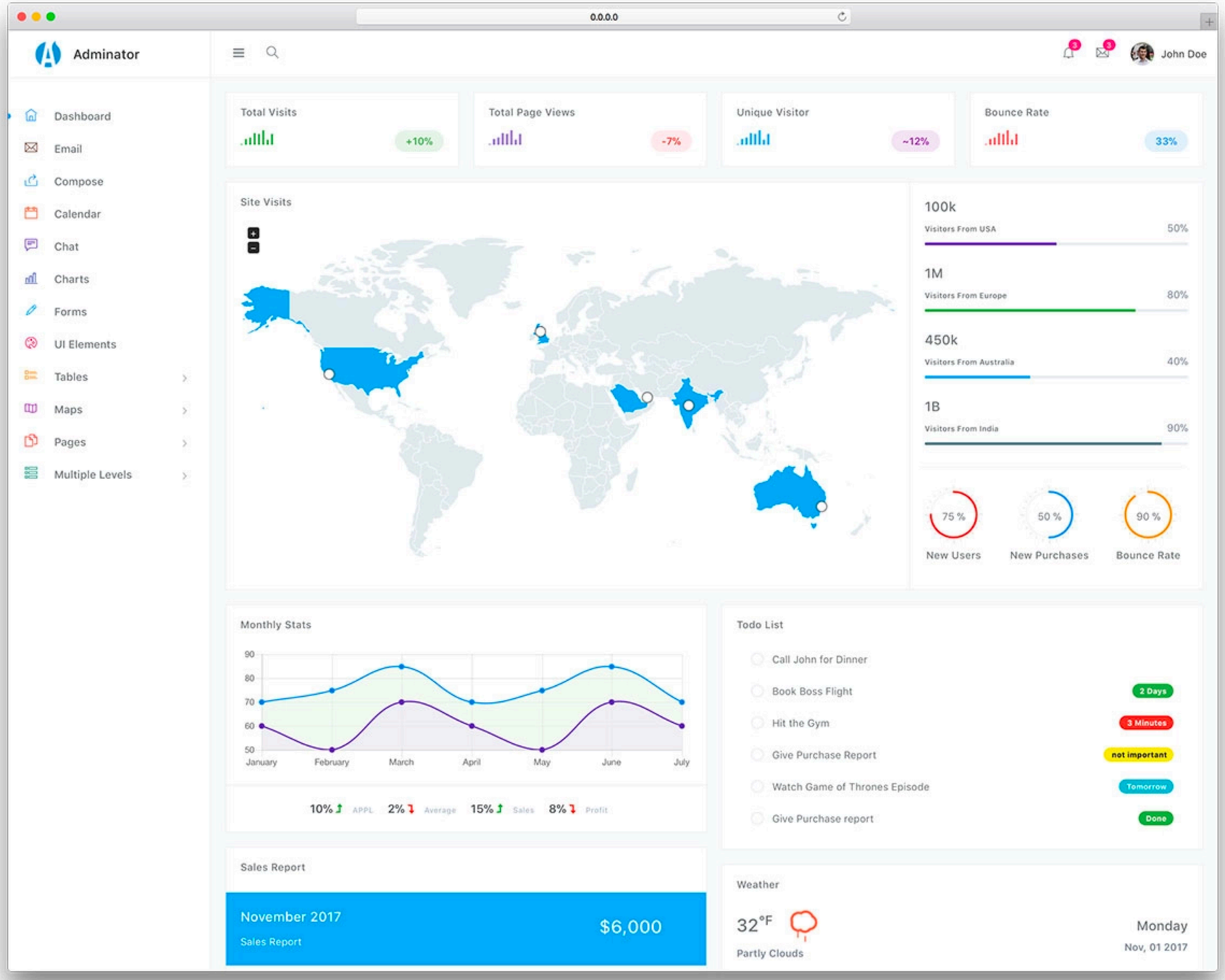
EVOLUINDO SUA APLICAÇÃO UTILIZANDO

MICRO FRONTENDS

SAMUEL MARTINS

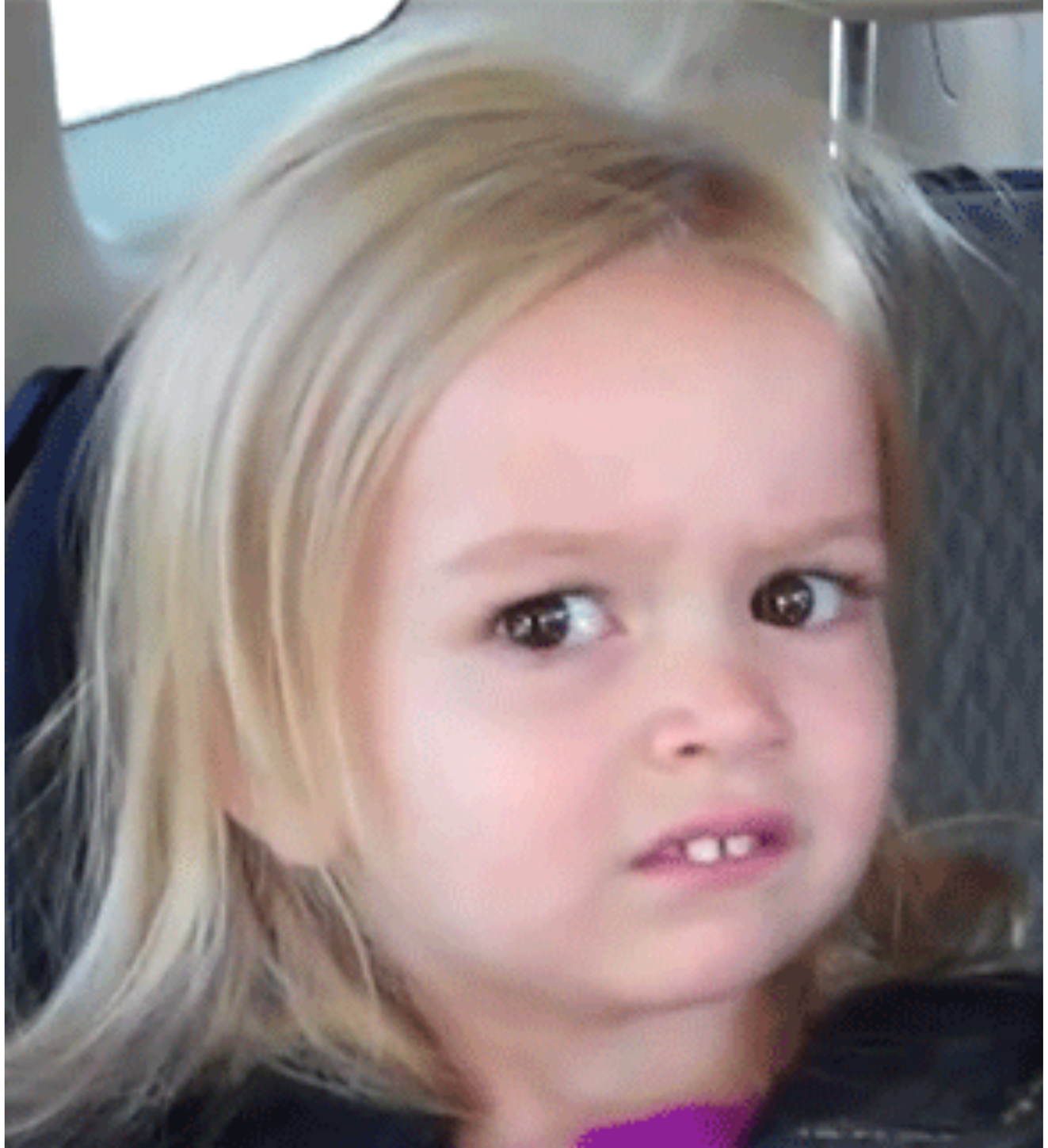
- Analista de Sistemas na Take;
- Formado em Sistemas de Informação pela PUC Minas;
- Professor de pós-graduação na PUC Minas nos cursos de Node.js e React + Angular;

CENÁRIO HIPOTÉTICO


0.0.0.0
Adminator
John Doe
Dashboard
Email
Compose
Calendar
Chat
Charts
Forms
UI Elements
Tables
Maps
Pages
Multiple Levels
Total Visits
+10%
Total Page Views
-7%
Unique Visitor
~12%
Bounce Rate
33%
Site Visits
100k
Visitors From USA
50%
1M
Visitors From Europe
80%
450k
Visitors From Australia
40%
1B
Visitors From India
90%
75 %
New Users
50 %
New Purchases
90 %
Bounce Rate
Monthly Stats
90
80
70
60
50
January
February
March
April
May
June
July
10%
APPL
2%
Average
15%
Sales
8%
Profit
Todo List
Call John for Dinner
Book Boss Flight
2 Days
Hit the Gym
3 Minutes
Give Purchase Report
not important
Watch Game of Thrones Episode
Tomorrow
Give Purchase report
Done
Sales Report
November 2017
Sales Report
$6,000
Weather
32°F
Partly Clouds
Monday
Nov, 01 2017

STACK

- AngularJs;
- Arquitetura MVVM (Model, View e View Model);
- ~50k linhas de código;
- Aplicação totalmente monolítica, todo o código no mesmo repositório;
- ~3 times com ~3 front-enders cada.



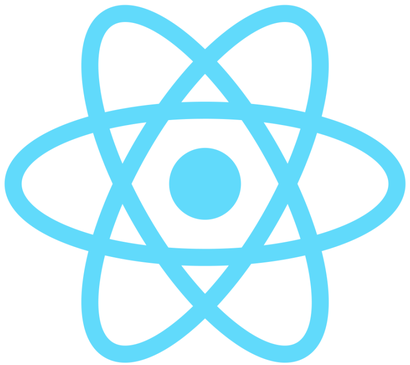
DIFICULDADES

- Comunidade **abandonou** o AngularJs;
- Componentes/diretivas **abandonadas**. Pull requests nem são olhados;
- Documentação do AngularJs **abandonada** e bem ruim;
- Componentes **difíceis** de testar;
- **Alta curva de aprendizado** para novos integrantes time;
- Services do angular **não favorecem o reaproveitamento** de código.

E MAIS...

- Falta de autonomia dos times;
- Deploys são mais sucetíveis a falhas;
- Deploy fica cada vez mais demorado;
- Aplicação totalmente amarrada a uma tecnologia;

ECOSSISTEMA FRONTEND EM CONSTANTE EVOLUÇÃO

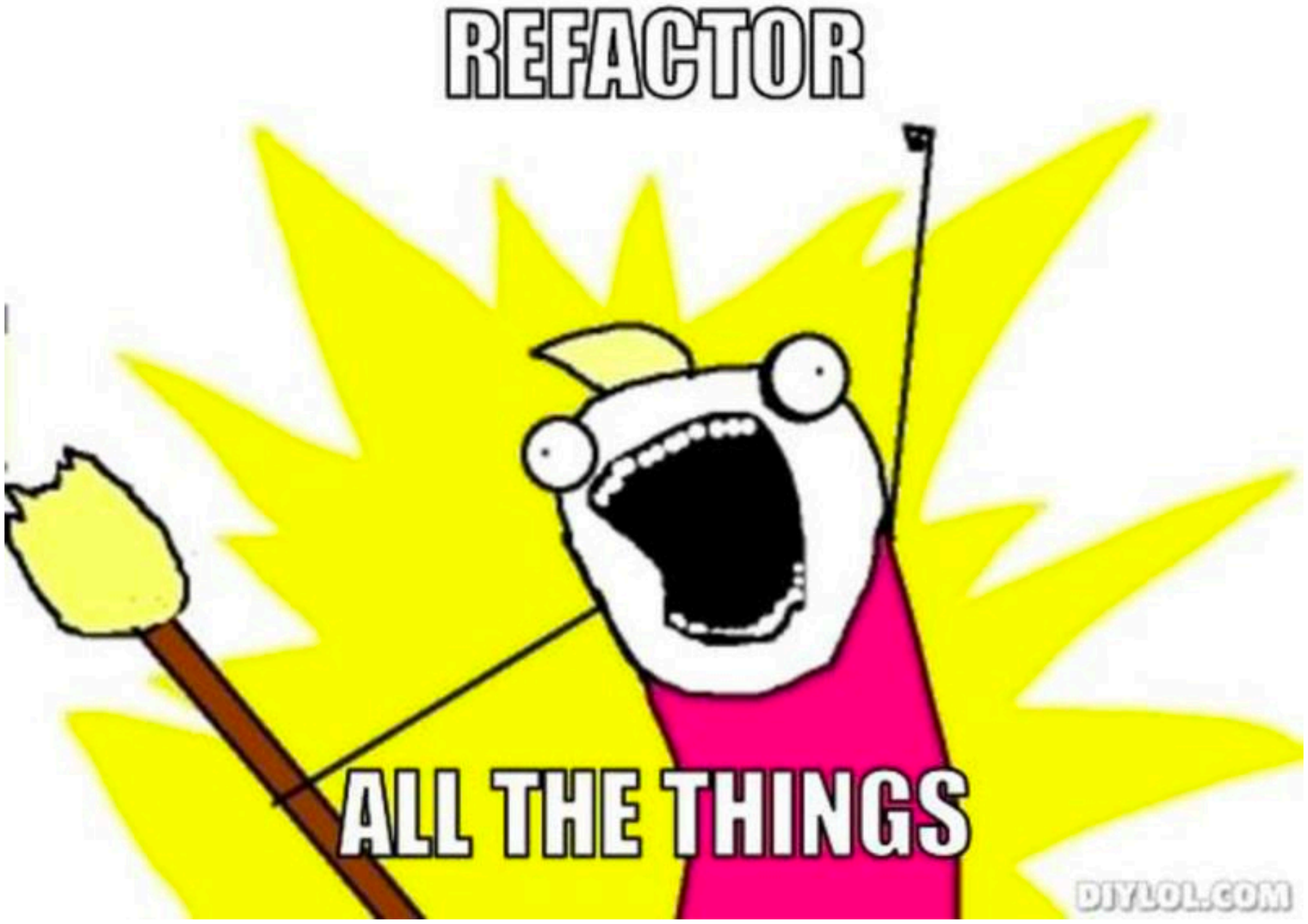


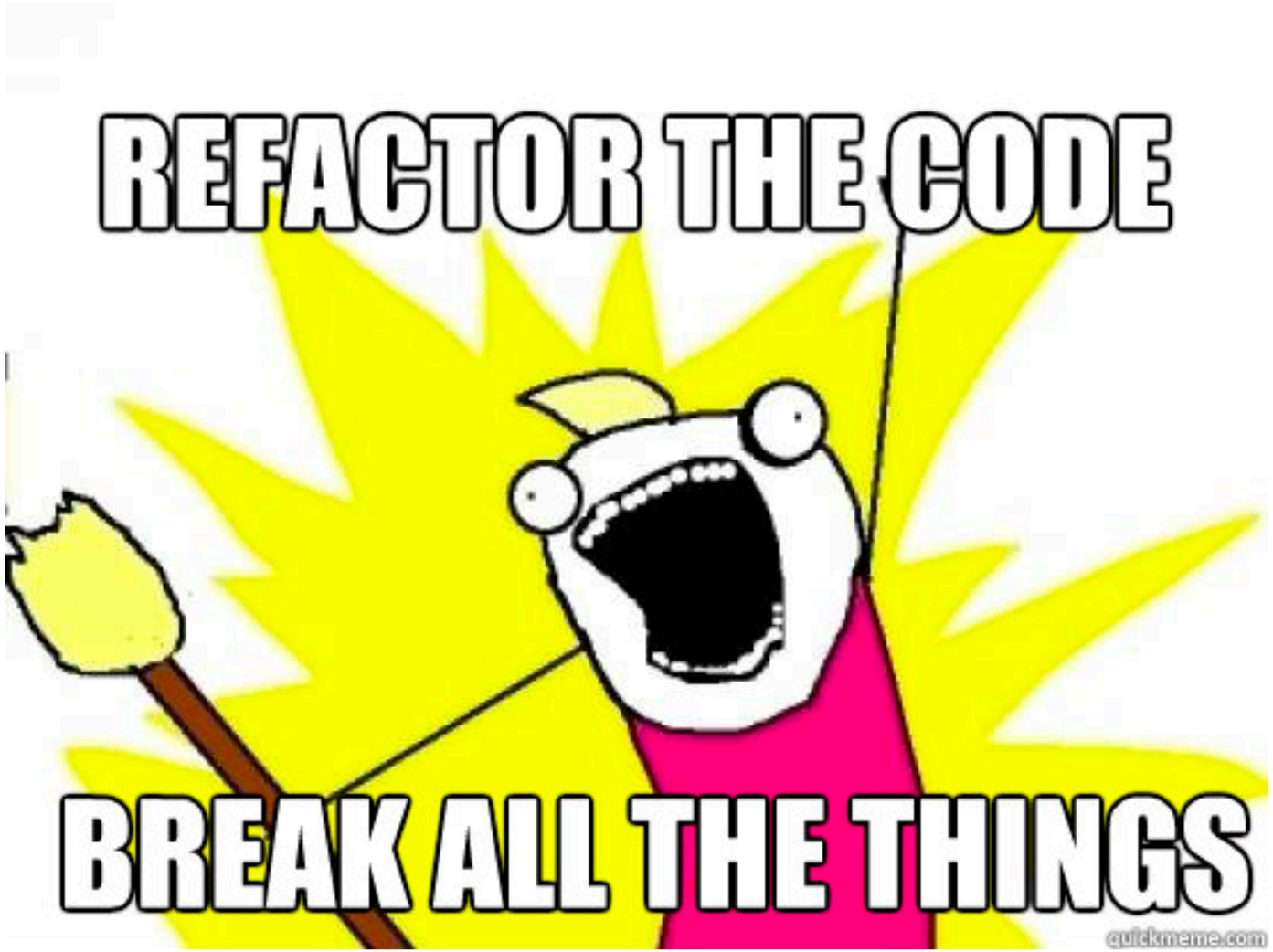
O QUE FAZER DIANTE DESSE CENÁRIO?

1ª OPÇÃO

DEIXAR COMO ESTÁ E IR EMBORA

2ª OPÇÃO


REFACTOR
ALL THE THINGS
DIYLOL.COM


REFACTOR THE CODE
BREAK ALL THE THINGS
quickmeme.com

DIFICULDADES

- Refazer toda a lógica de negócio pode não ser viável;
- Todos os bugs resolvidos teriam que ser retestados;
- Falta de mão-de-obra;

- ~~Desistir e ir embora;~~
- ~~Refatorar tudo;~~

3ª OPÇÃO

REFATORAR GRADATIVAMENTE PARA UM NOVO FRAMEWORK

DIFICULDADES

- Manter dois frameworks numa mesma base de código;
- Aplicação amarrada a um novo framework;
- Compartilhar informações entre as partes pode ser complicado.

- ~~Desistir e ir embora;~~
- ~~Refatorar tudo;~~
- ~~Migrar gradativamente para um novo framework;~~

4ª OPÇÃO

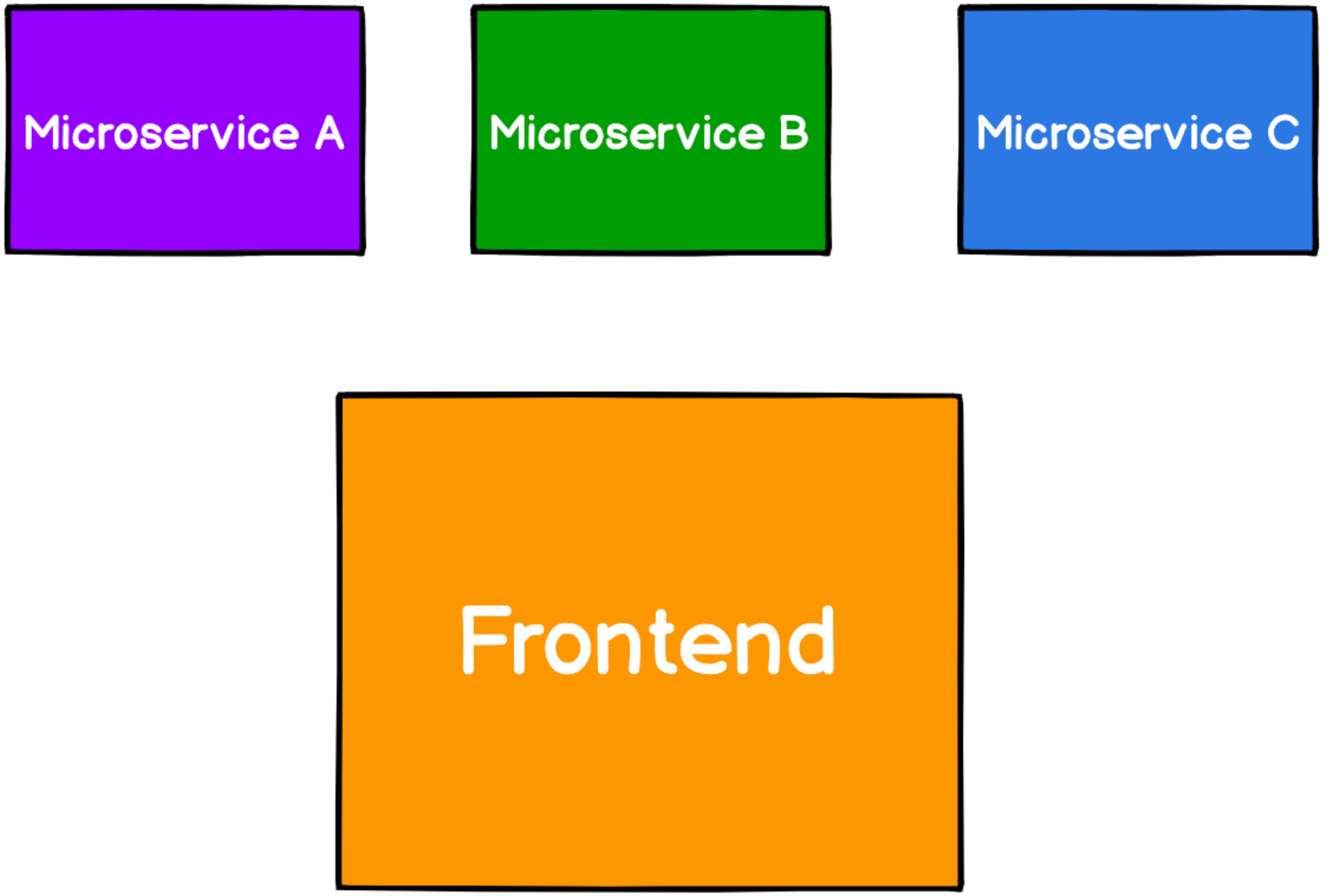
UTILIZAR
MICRO FRONTENDS

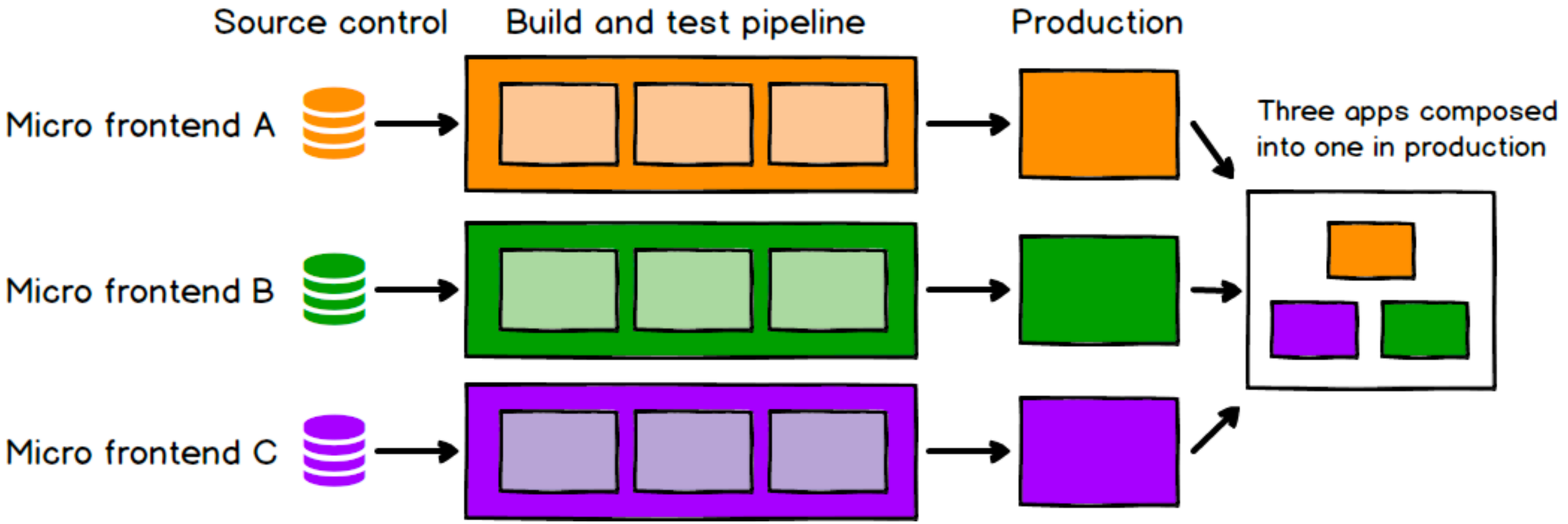
MAS AFINAL, O QUE É UM MICRO FRONTEND?

- Não é um framework;
- Não é uma biblioteca;
- Não é uma extensão;
- Não é magia;

É UM ESTILO ARQUITETURAL

- Baixo ou nenhum acoplamento;
- Alta coesão;
- Não deve assumir responsabilidades de outro micro frontend;
- Não deve interferir ou ser interferido por outro micro frontend;
- Base de código independente;
- Pipeline de build, test e deploys separados e indepentendes;

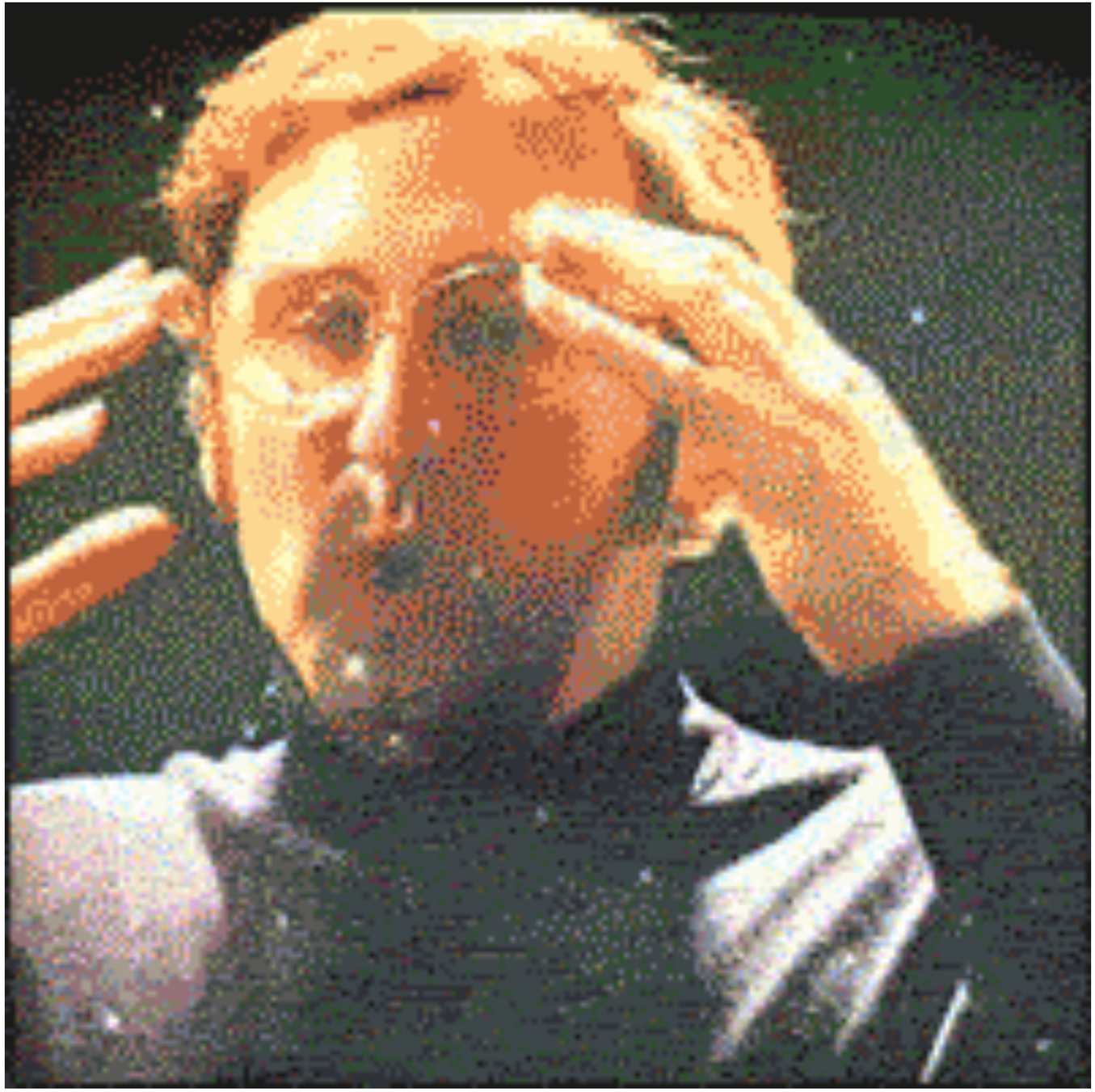

Microservice A
Microservice B
Microservice C
Frontend



Fonte: https://martinfowler.com/articles/micro-frontends.html

BENEFÍCIOS

- Modernização da aplicação sem a necessidade de jogar tudo fora;
- Aplicação totalmente agnóstica de novas tecnologias;
- Migração gradativa do código legado;
- Pipeline de build, test e deploys mais rápida;
- Maior tolerância a falhas;
- Separação de micro frontends por times.



BELEZA, MAS #COMOFAZ?

PACOTES NPM SEPARADOS

```
{
  "name": “@my-project/main”,
  "version": "1.0.0",
  "description": “My amazing application",
  "dependencies": {
    “@my-project/frontend-1“: “1.0.0”,
    “@my-project/frontend-2“: “1.0.0”,
    “@my-project/frontend-3“: “1.0.0”,
  }
}
```

- ✓ ~~Baixo ou nenhum acoplamento;~~
- ✓ ~~Alta coesão;~~
- ✓ ~~Não deve assumir responsabilidades de outro micro frontend;~~
- ✓ ~~Não deve interferir ou ser interferido por outro micro frontend;~~
- ✓ ~~Base de código independente;~~
- Pipeline de build, test e deploys separados independentes;

WEB COMPONENTS

```
// /about page
<div id="container">
  <about-micro-frontend></about-micro-frontend>
</div>

// /products page
<div id="container">
  <products-micro-frontend></products-micro-frontend>
</div>
```

WEB COMPONENTS

```
{
  "name": "@my-project/web-components",
  "version": "1.0.0",
  "description": "My amazing application",
  "dependencies": {
    "@my-project/about-micro-frontend": "1.0.0",
    "@my-project/products-micro-frontend": "1.0.0"
  }
}
```

WEB COMPONENTS

```
<script src="https://about.project.com/bundle.js"></script>

<script src="https://products.project.com/bundle.js"></script>


// /about page

<div id="container">

  <about-micro-frontend></about-micro-frontend>

</div>


// /products page

<div id="container">

  <products-micro-frontend></products-micro-frontend>

</div>
```

- ✓ ~~Baixo ou nenhum acoplamento;~~
- ✓ ~~Alta coesão;~~
- ✓ ~~Não deve assumir responsabilidades de outro micro frontend;~~
- ✓ ~~Não deve interferir ou ser interferido por outro micro frontend;~~
- ✓ ~~Base de código independente;~~
- ✓ ~~Pipeline de build, test e deploys separados independentes;~~
- Rotas “filhas" é um problema

IFRAMES

```
// /about page
<div id="container">
  <iframe src="https://about.project.com"></iframe>
</div>

// /products page
<div id="container">
  <iframe src="https://products.project.com"></iframe>
</div>
```

- ✓ ~~Baixo ou nenhum acoplamento;~~
- ✓ ~~Alta coesão;~~
- ✓ ~~Não deve assumir responsabilidades de outro micro frontend;~~
- ✓ ~~Não deve interferir ou ser interferido por outro micro frontend;~~
- ✓ ~~Base de código independente;~~
- ✓ ~~Pipeline de build, test e deploys separados independentes;~~
- ✓ ~~Rotas “filhas" é um problema.~~

DOCKER + KUBERNETES

```yaml
version: '2'
services:
  micro-frontend-1:
    container_name: micro-frontend-docs-page
    build: docs-page
    networks:
      - local-network
    environment:
      - NODE_ENV=production
      - PORT=5000

  micro-frontend-nginx:
    container_name: micro-frontend-nginx
    build: nginx
    volumes:
      - ./assets:/var/www
    ports:
      - "8888:80"
    networks:
      - local-network
    depends_on:
      - micro-frontend-1
```

DESAFIOS

- Comunicação entre partes da aplicação pode ser complicada;
- Aumento da complexidade da aplicação;
- Maior curva de aprendizado para novos integrantes do time.

```
/**
* Send message to parent iframe or to specified window
* @param payload
* @param element
*/
public sendMessage(payload: IMessagePayload): Promise<any> {
  const message = this.formatPayload(payload)
  const deferred = createDeferred()

  this.createPromiseCache(message.trackingProperties.id, deferred)
  this.targetWindow.postMessage(message, '*')

  return deferred.promise
}
```

```
init(): void {
  this.handleOnReceiveMessage = this.onReceiveMessage.bind(this)
  window.addEventListener('message', this.handleOnReceiveMessage)
}

onReceiveMessage(message: MessageEvent) {
  const action = message.data

  switch(action) {
    case 'a':
      doThis();
      break;
    case 'b':
      doThat();
      break;
    default:
      throw new Error('Unrecognized action')
  }
}
```

Fonte: https://github.com/takenet/iframe-message-proxy

"TUDO É SOBRE SAIR DA ZONA DE CONFORTO"

Samuel Martins

OBRIGADO!