



THE DEVELOPER'S CONFERENCE

{API-First Design}: Padrões de design para
APIs REST com foco em **#DeveloperExperience**

Guilherme de Cleva Farto
P&D e Arquitetura TOTVS



Agenda

- Contextualização e importância das APIs
- Modelo tradicional de APIs
- Desafios em *API Design*
- Modelo *API-First Design*
- Três regras para *API-First*
- O que esperar de *API-First*?
- *Developer Experience* (DX)
- Estratégias para DX e APIs
- *API steps to greatness*

Contextualização



THE
DEVELOPER'S
CONFERENCE

“APIs are perhaps the most critical technology in digital business design”

Forrester Research, 2015

“APIs are how we are going to build software in the future. We are just going to glue it together”

John Musser, founder of Programmable Web

A importância de APIs



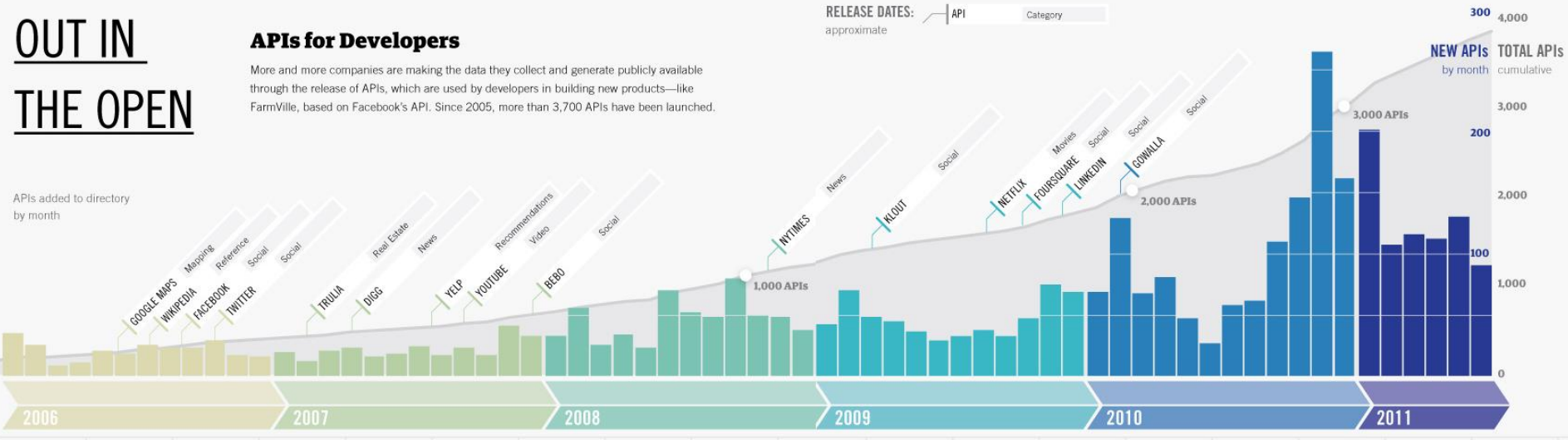
THE
DEVELOPER'S
CONFERENCE

OUT IN THE OPEN

APIs for Developers

More and more companies are making the data they collect and generate publicly available through the release of APIs, which are used by developers in building new products—like FarmVille, based on Facebook's API. Since 2005, more than 3,700 APIs have been launched.

RELEASE DATES: API Category
approximate



Fonte: <http://visual.ly/open-data-movement>

Por que APIs?



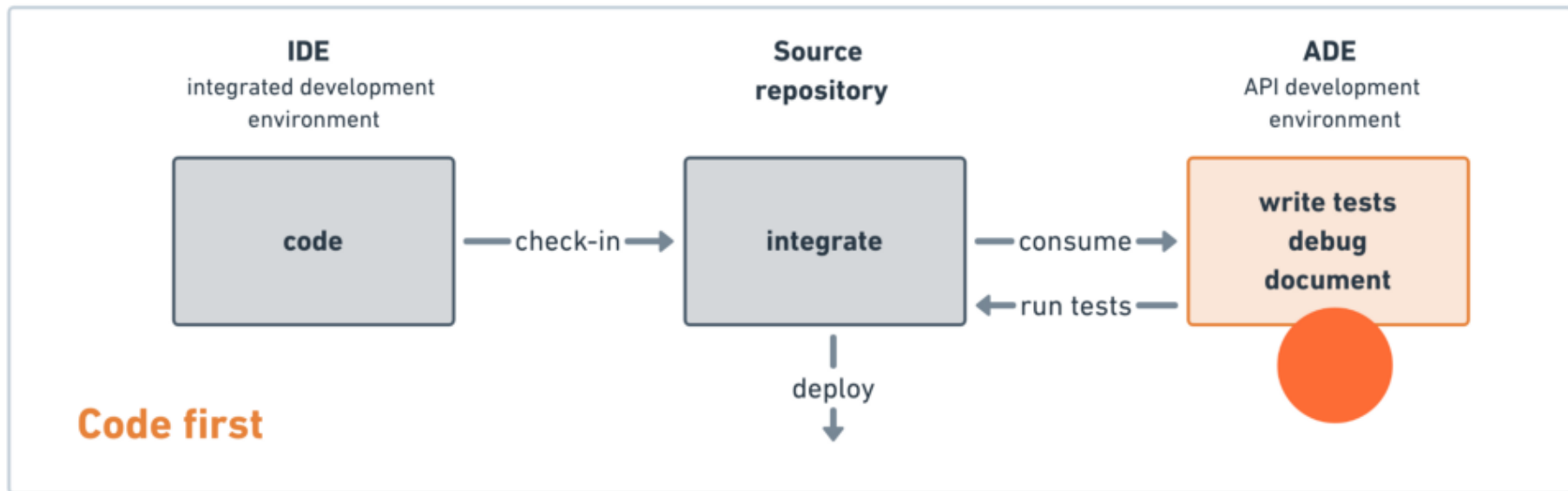
- Conecta dados, pessoas, processos e dispositivos digitais
- Possibilita lidar com o aumento exponencial de volume, escala e volatilidade de aplicativos e dispositivos
- Apoia a transformação digital e a experiência do usuário
 - Integrar melhor soluções internas e de parceiros
 - Abrir e compartilhar dados para capturar novas oportunidades

Netflix APIs

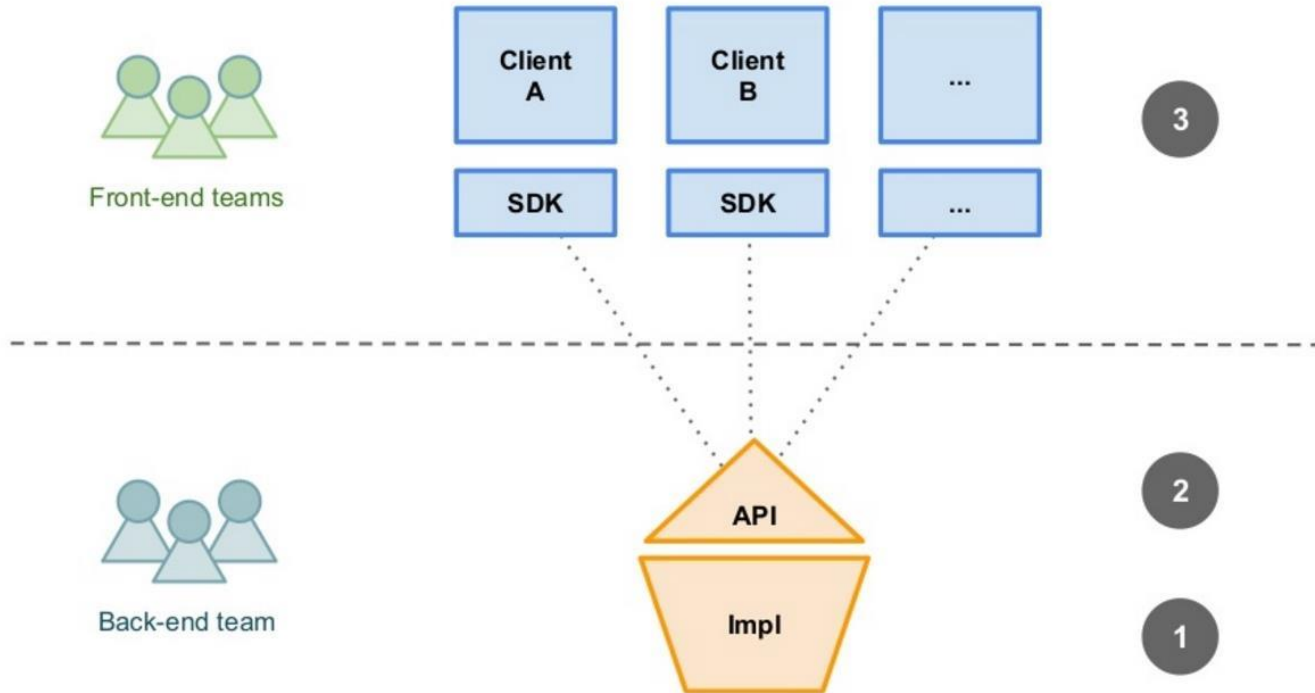


- Mais de 137 mi de usuários em mais de 100 países (Q3 2018)
- De 600 mi (2010) para ~42 bi (2012) de requisições
- Mais de 1k dispositivos compatíveis
 - Distintos sistemas operacionais
 - Distintas especificações técnicas
 - Distintas linguagens e plataformas de programação

Modelo tradicional de APIs



Modelo tradicional de APIs



Desafios em API Design



- Para quem é a API?
- Importância do *design* e de documentações?
- Qual é a experiência para o usuário da API?
- Completude e concisão das APIs?

- Grandes corporações também têm se dedicado aos temas

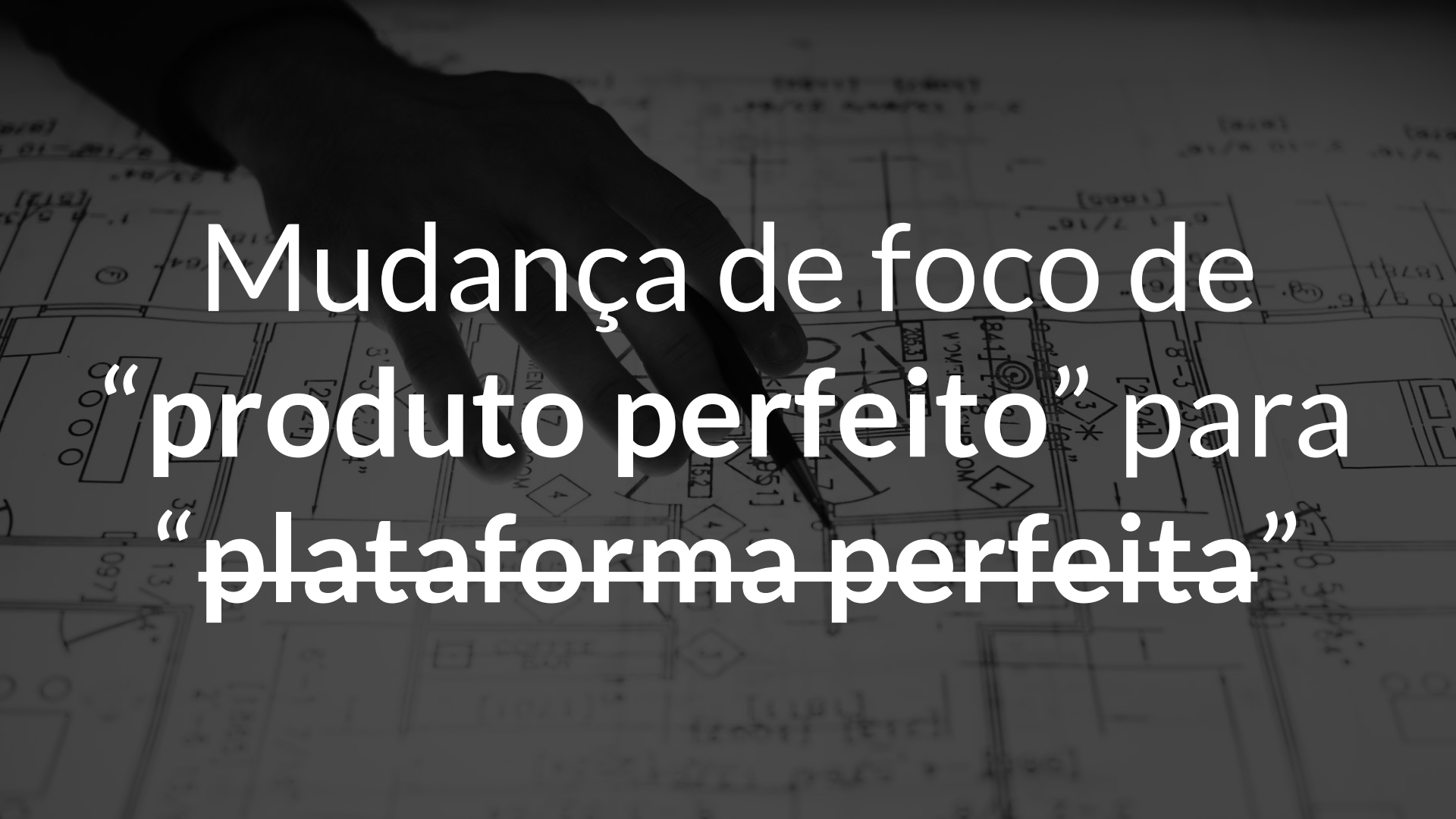
O mandato de Jeff Bezos



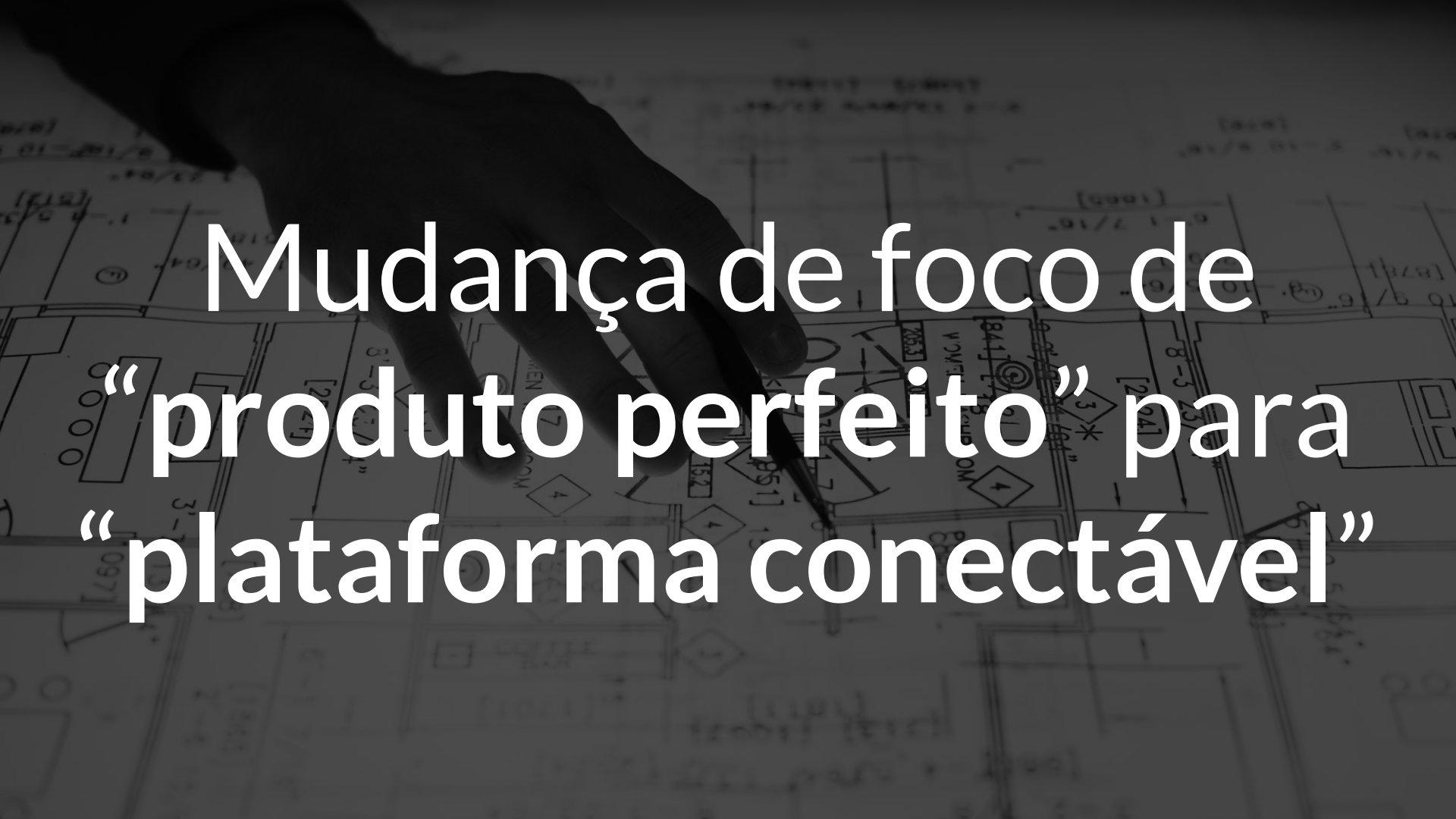
- *All teams will expose their data through **service interfaces***
- *Teams must **communicate through interfaces***
- ***No other form of interprocess communication allowed***
- *Interfaces must be **designed to be externalizable***
- *It **doesn't matter what technology you use***
- Descrito em 2002 como “Amazon’s API Mandate”

A hand holding a pen pointing to a technical drawing or blueprint. The drawing contains various lines, shapes, and text, including the word "MEN" and some numbers. The background is dark and the text is white.

Mudança de foco de
“produto perfeito” para
“plataforma perfeita”



Mudança de foco de
“produto perfeito” para
“plataforma perfeita”

A hand is shown pointing at a set of architectural blueprints. The blueprints contain various technical drawings, lines, and text. Overlaid on the image is a large, white, sans-serif text block. The text reads: "Mudança de foco de 'produto perfeito' para 'plataforma conectável'".

Mudança de foco de
“produto perfeito” para
“plataforma conectável”

Tecnologia e Design



*Responsive
Web Design*

Mobile First

?

NLP First

Data First

AI First

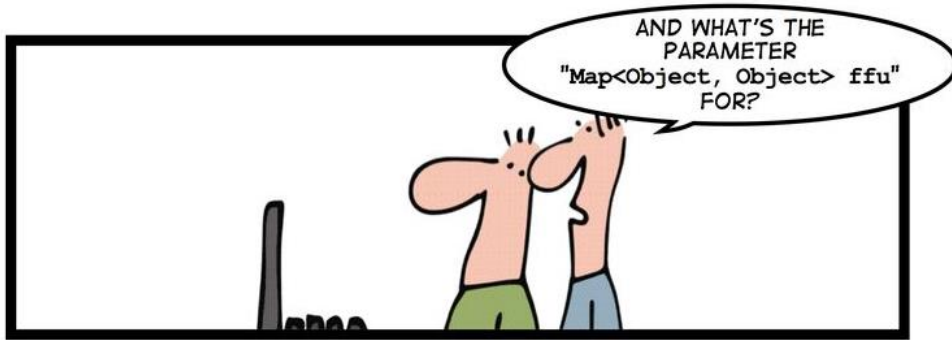
API-First Design



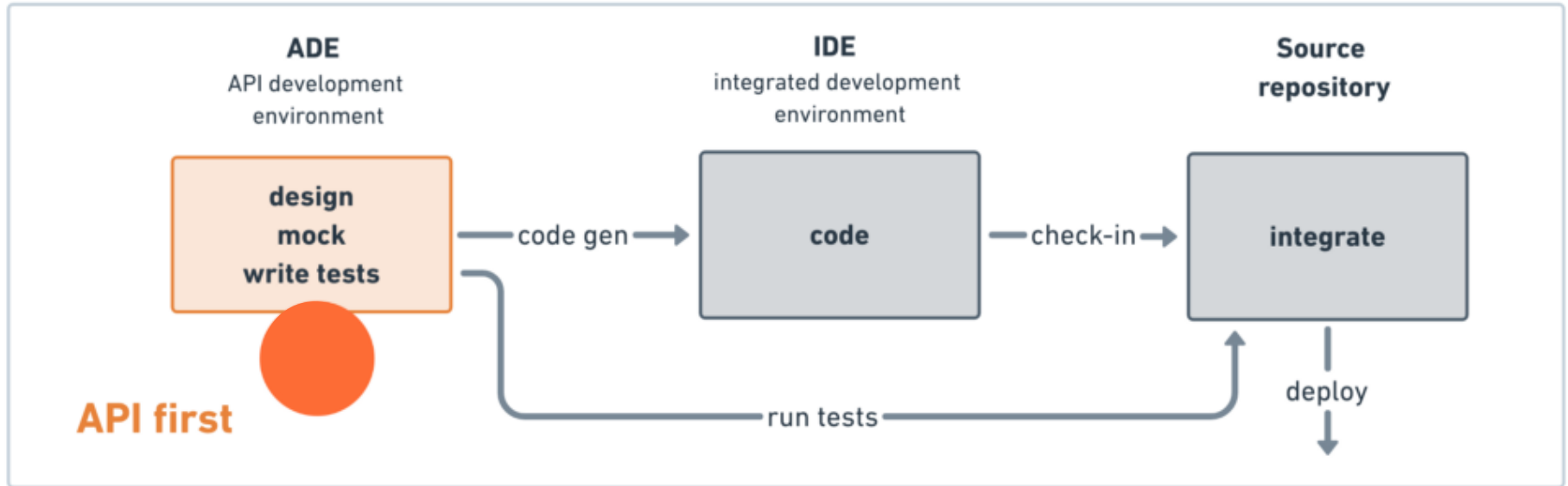


...

HOW TO CREATE A STABLE API



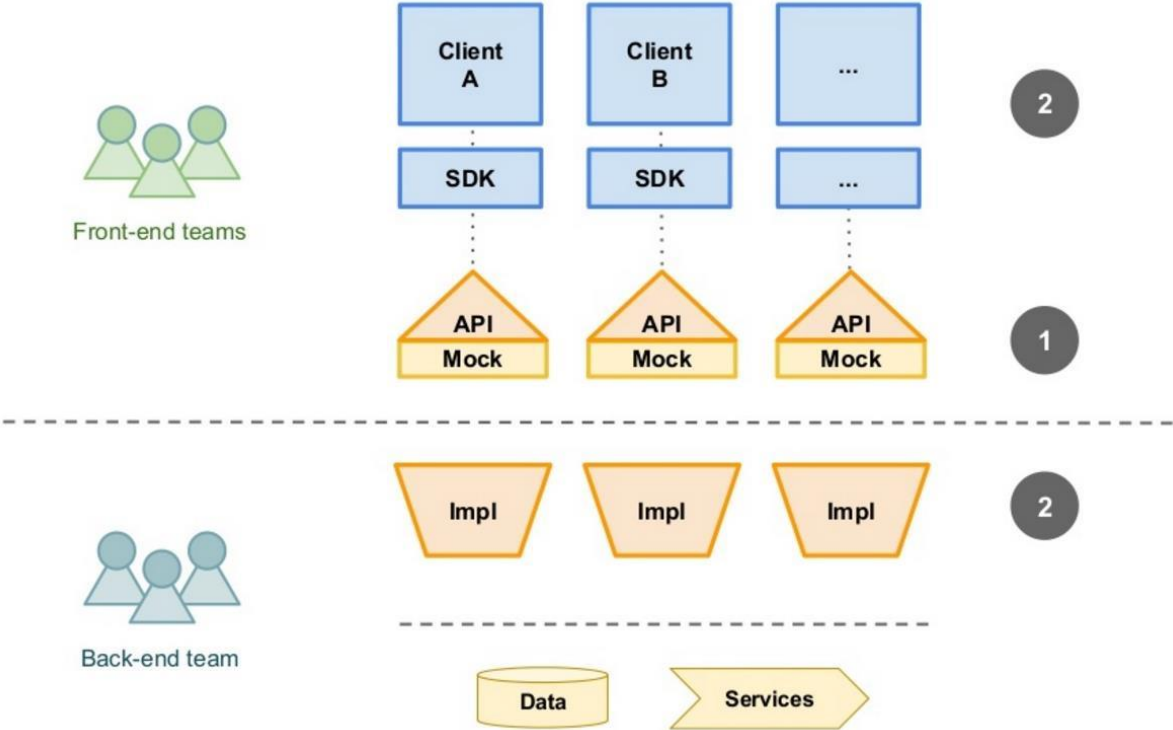
Modelo API-First Design



Modelo API-First Design



THE
DEVELOPER'S
CONFERENCE

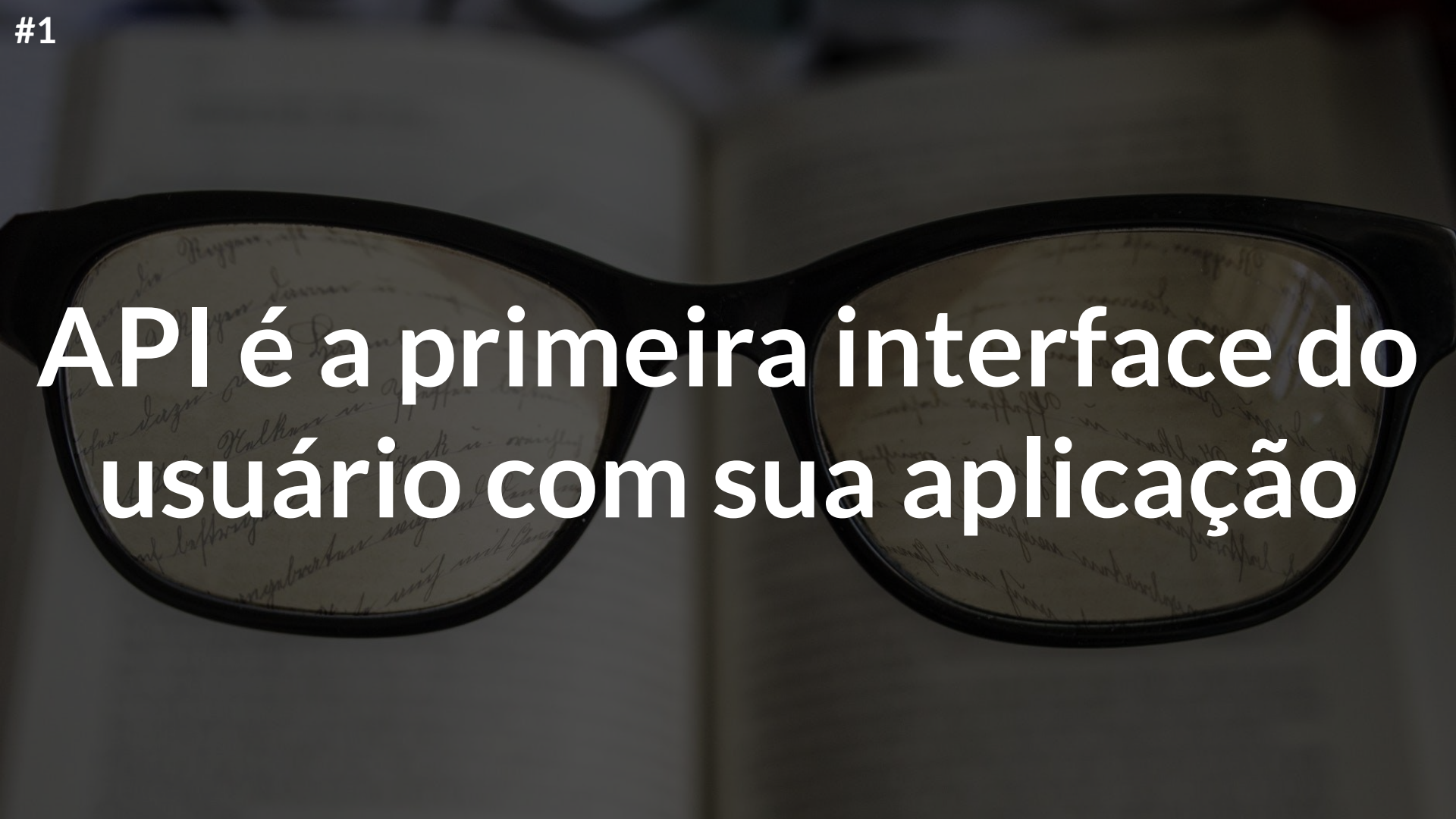


Três regras para API-First



THE
DEVELOPER'S
CONFERENCE

#1



API é a primeira interface do
usuário com sua aplicação

#2

Design da API vem primeiro;
depois a implementação

#3

A person is shown from the chest up, holding a silver metal can with both hands. They are drinking through a clear plastic straw that goes into the can. The person's mouth is open, and they appear to be in the middle of a drink. The background is a plain, light-colored wall. The overall image has a dark, semi-transparent overlay.

**Facilmente entendida e
auto descritiva**

O que esperar de API-First?



THE
DEVELOPER'S
CONFERENCE

- Foco na qualidade de APIs (*design* e implementação)
- Reduz interdependências entre times
- Permite escalar a solução e suas integrações para *multiclients*
- Maior velocidade para novas soluções chegarem ao mercado
- Promove *open innovation*: IoT, Smart Env, Data Science, ...

Como fazer isto?



- Definir e planejar sua estratégia de API-First
 - Plano de evolução do modelo tradicional para API-First *design*
 - Compartilhe com os times os benefícios de API-First
 - Guidelines, padronizações e principais *features* das APIs
 - [API Stylebook](#) e [OData documentation](#)
 - Defina seu conjunto de ferramentas de apoio
 - Pense no ciclo de vida das APIs
 - Modelos de versionamento e evolução de suas APIs

Developer Experience (DX)



- Elabore documentação completa, porém simples e interativa
- Envolve desenvolvedores nas discussões de *design* de APIs
- Colete e avalie *feedbacks* de suas APIs
- Torne sua plataforma de APIs extensível

Estratégias para DX e APIs



➤ *Common Description Language*

➤ Swagger ([OpenAPI Initiative](#))

➤ Ecossistema Swagger: Editor, Codegen, UI e outros

➤ Fácil integração com distintas plataformas

➤ Alternativas: RAML, API Blueprint, WADL, Apiary e outros

swagger api
Search term

raml api
Search term

api blueprint
Search term

wadl api
Search term

apiary api
Search term

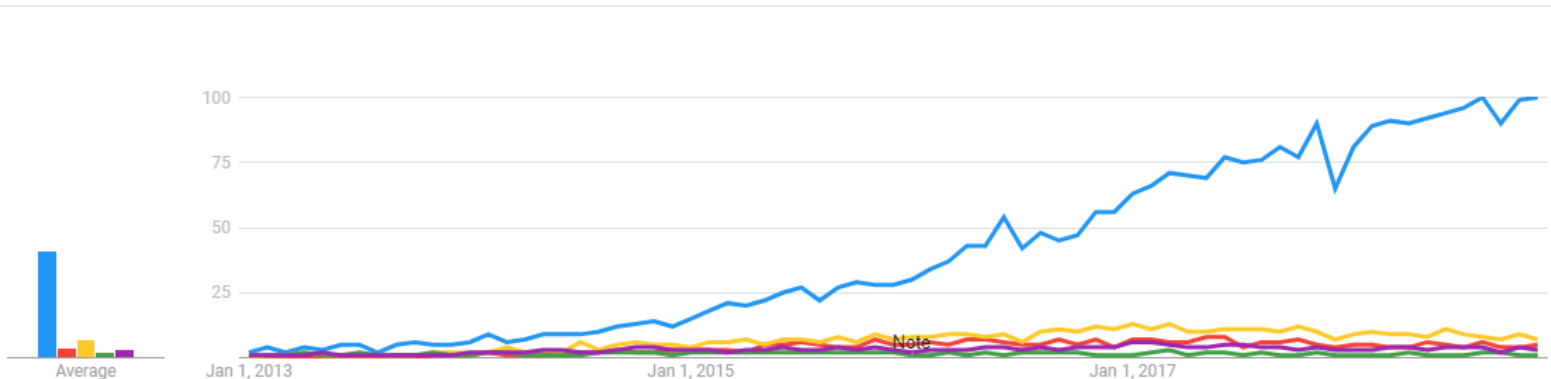
Worldwide

1/1/13 - 12/2/18

All categories

Web Search

Interest over time



TDC - Workshop APIs 1.0.0

[Base URL: localhost:9090/tdc]

<http://localhost:9090/tdc/v2/api-docs>

Workshop de APIs no TDC Porto Alegre 2018

[TDC Porto Alegre 2018 - Website](#)

[Send email to TDC Porto Alegre 2018](#)

hello-world-controller Hello World Controller

GET /api/hello hello

GET /api/hello/{name} getHelloPath

GET /api/hello/withHeader getHelloHeader

GET /api/hello/withParam getHelloParam

POST /api/users Inseere usuário



GET /api/users/{login} Recupera usuário pelo {login}



PUT /api/users/{login} Atualiza usuário pelo {login}



DELETE /api/users/{login} Remove usuário pelo {login}



DELETE /api/users/deleteAll Remove todos os usuários



GET /api/users/filter Filtra listagem de usuários



Models



```
User ▾ {
  description:      Representa uma entidade de usuário
  createdAt        string($date-time)
                   Data de criação do usuário
  login            string
                   example: user.test
                   x-database-metadata: OrderedMap { "column": "LOGIN" }
                   Login do usuário
  mail            string
                   example: user.test@gmail.com
                   x-database-metadata: OrderedMap { "column": "MAIL" }
                   E-mail do usuário
  name            string
                   example: User Test
                   x-database-metadata: OrderedMap { "column": "NAME" }
```

Responses

Response content type

application/json

Curl

```
curl -X GET "http://localhost:9090/tdc/api/users" -H "accept: application/json" -H "x-api-key: tdc"
```

Request URL

```
http://localhost:9090/tdc/api/users
```

Server response

Code

Details

200

Response body

```
[
  {
    "login": "guilherme.farto",
    "name": "Guilherme Farto",
    "mail": "guilherme.farto@gmail.com",
    "createdAt": "2018-12-03T18:09:28.260+0000",
    "status": true,
    "userType": "admin"
  }
]
```

Response headers

```
content-type: application/json;charset=UTF-8
date: Mon, 03 Dec 2018 18:41:58 GMT
transfer-encoding: chunked
```

Request duration

27 ms

Estratégias para DX e APIs



THE
DEVELOPER'S
CONFERENCE

➤ SDKs, API Clients e Code Snippets

```
@ApiOperation("Filtra listagem de usuários")
@ApiResponses({
    @ApiResponse(code = 200, message = "Usuários filtrados com sucesso"),
    @ApiResponse(code = 401, message = "Falha na autenticação") })
@RequestMapping(value = "/filter",
    method = RequestMethod.GET,
    produces = MediaType.APPLICATION_JSON_VALUE)
@ResponseBody
public List<UserVO> filterUsers(
    @ApiParam(value = "Tipo de usuário", required = false)
    @RequestParam(name = "userType", required = false) UserTypeEnum userType) {
    return bo.filterUsers(userType);
}
```


Estratégias para DX e APIs



➤ *API Mocking*

- Permite “acelerar” desenvolvimentos em um API-First design
 - APIs com regras de negócio complexas
 - Elimina interdependências entre times
 - Desenvolvimento *offline*
- Mocky, Mockable, mockAPI, WireMock, Mockoon, MockServer, GetSandbox e outros

Showing 5 of 5 stubs

Text search

GET /my-things

200

Body content for stub 5

Delete everything!

DELETE /my-things

200

Body content for stub 4

Put a thing

PUT /thing/12345

200

Body content for stub 3

Post some JSON

POST /my-things

201

Body content for stub 2

Basic GET

GET /thing/1

Request

WireMock

Name

Text search

GET

/my-things

▼ Advanced ?

URL match type

Priority

Path

4

+ Query param

+ Header

+ Cookie

+ Body

q

contains

MockLab



(query)

► Scenario ?

Response

Direct

Fault

Proxy

Status

200

Enable templating ?



THE
DEVELOPER'S
CONFERENCE

Estratégias para DX e APIs



- **Portal para desenvolvedores**
 - Reúna as experiências em um único local
 - Simplifique o acesso às documentações e artefatos das APIs
 - Conteúdo interativo, dinâmico e colaborativo (+ *gamification*)
 - Tutoriais, artigos, comunidades e fóruns
 - *Developer playground e sandboxes*
 - Gelato, Stoplight, Hootr, OpenChannel, Pronovix, Box Platform e outros



Read

Design

Code

HTTP Debug

Environments

4



Todos Service API



TODOS

Get Todo

GET

Update Todo

PUT

Delete Todo

DELETE

List Todos

GET

Create Todo

POST

MODELS

todo-partial

todo-full

List Todos

GET /todos

This endpoint returns a list of Todo objects.

▸ 2 Query Parameters

Responses

200

▸ 2 Examples

▼ Schema

array[object]

name string

required

completed boolean or null

required

id integer

required

minimum: 0

maximum: 1000000

completed_at string or null

format: date-time

created_at string

format: date-time

updated_at string

format: date-time

Stoplight



PER'S
ENCE

rm

API steps to greatness

API Economy

*REST/JSON, Resources,
Operations, Error Handling,
Caching, and Testes*

Security

Metrics and Rate Limiting

Tracking

HATEOAS and API Linking

*Open Data Protocol
(OData)*

*Callbacks and
Event-Subscriber
(WebHooks e PubSub)*

GraphQL

Referências



- CIO. [Have you had your Bezos moment? What you can learn from Amazon](#), 2017
- Forrester Research, Inc., Brief: [Four Ways APIs Are Changing Your Business](#), 2015
- Geek and Poke. [How to create stable API?](#), 2012
- Netflix Technology Blog, [Embracing the Differences : Inside the Netflix API Redesign](#), 2012

Obrigado!



Guilherme de Cleva Farto



guilherme.farto@gmail.com



<https://www.linkedin.com/in/guilherme-farto/>



<https://github.com/guilhermefarto>



<https://twitter.com/gcleve>



THE DEVELOPER'S CONFERENCE