



Ciência da **Computação**

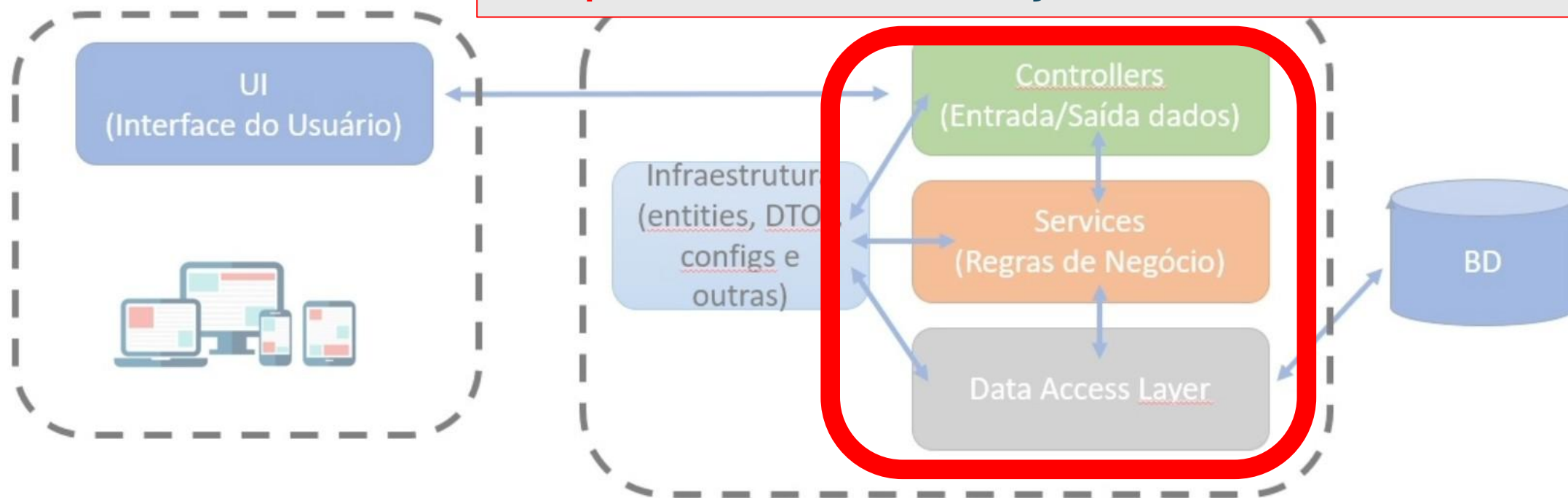
Programação Orientada a Objetos Avançado
Prof. Luciano Rodrigo Ferretto

E o nosso projeto??? Como vai ser???

- Arquitetura Cliente-Servidor
 - Somente o lado do Servidor
 - Seguindo os princípios KISS

Como será essa comunicação interna?

- **Como?** Através da invocação de métodos
- **Tipo de dados?** Objetos Java



E o nosso projeto??? Como vai ser???

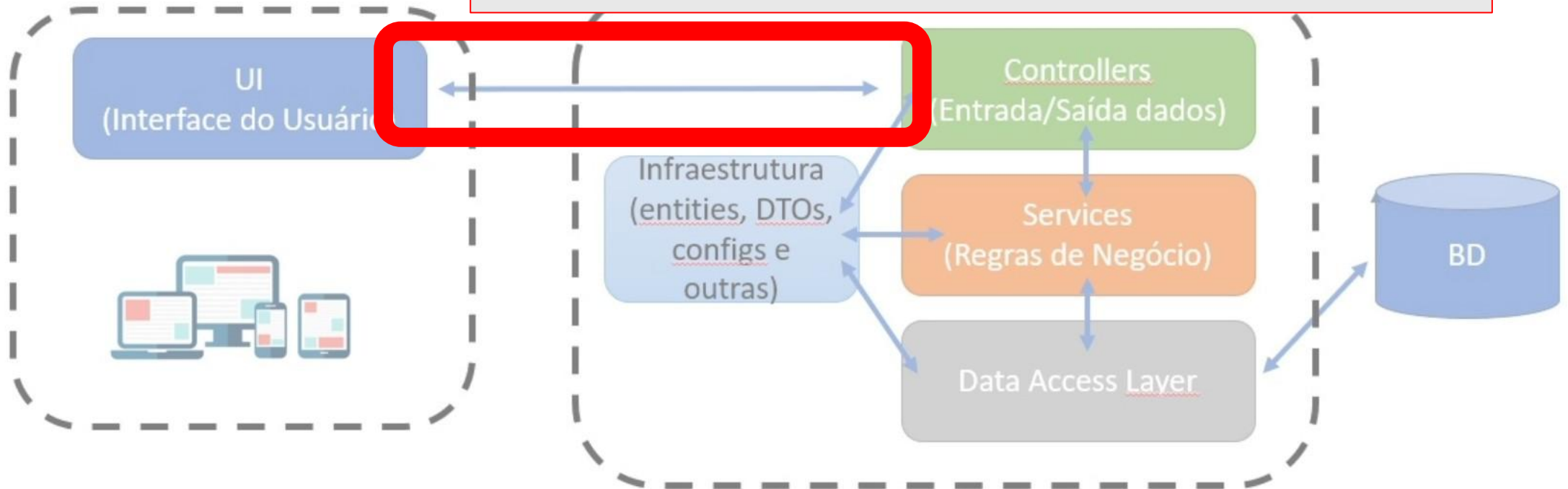
- Arquitetura Cliente-Servidor (VC)

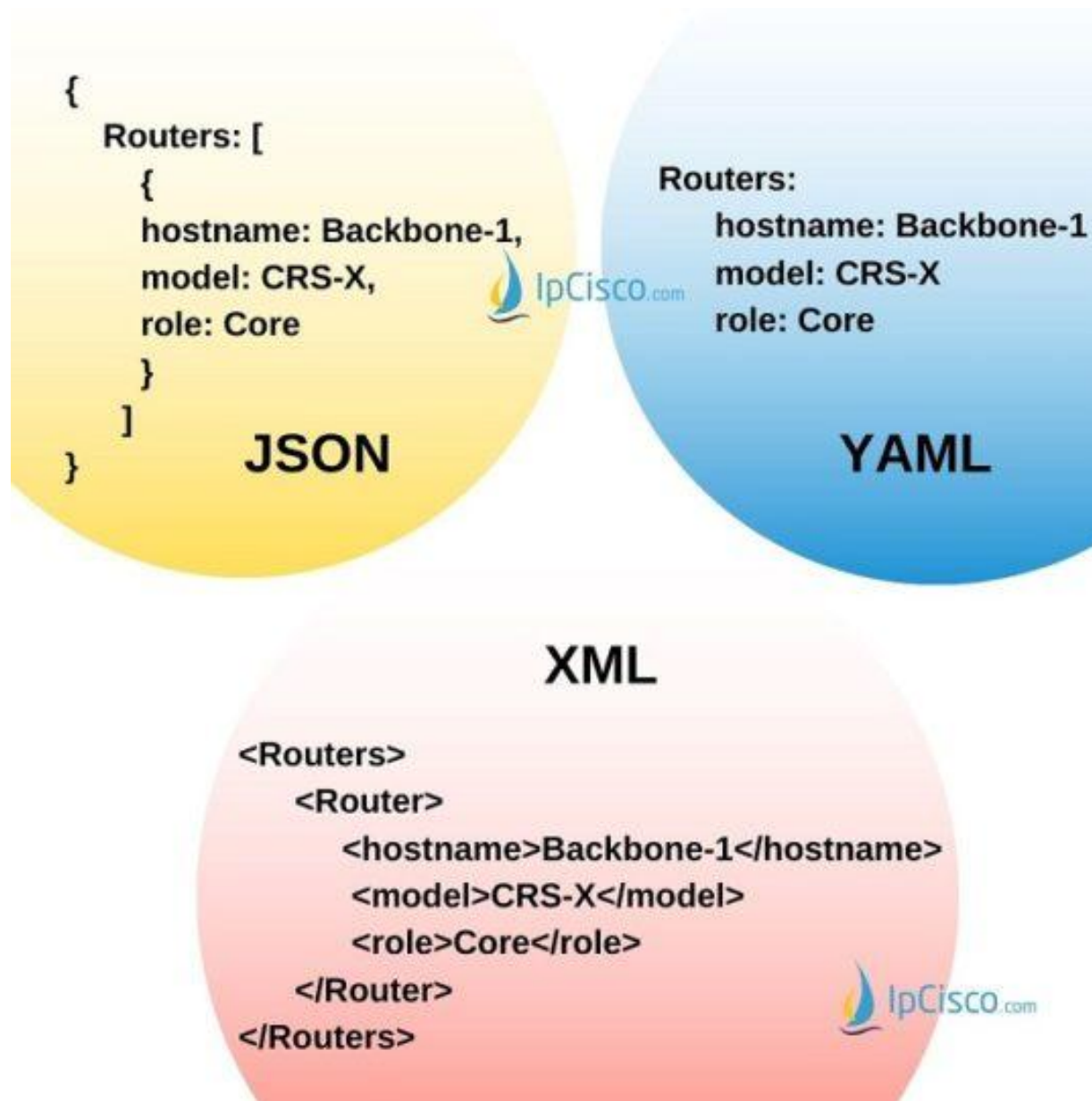
- Somente o lado do Servidor
- Seguindo os princípios KISS

Como será essa comunicação?

- Protocolo (soap, http, smtp, etc)?

- Tipo de dados (xml, json, yaml, etc)?





Por que JSON???

Quando usar o YAML versus JSON

Graças ao suporte abrangente e à integração com JavaScript, o JSON é um formato de serialização de dados mais popular para a maioria dos casos de uso do que o YAML. O JSON é usado extensivamente em comunicações de software distribuídas, aplicações Web, arquivos de configuração e APIs.

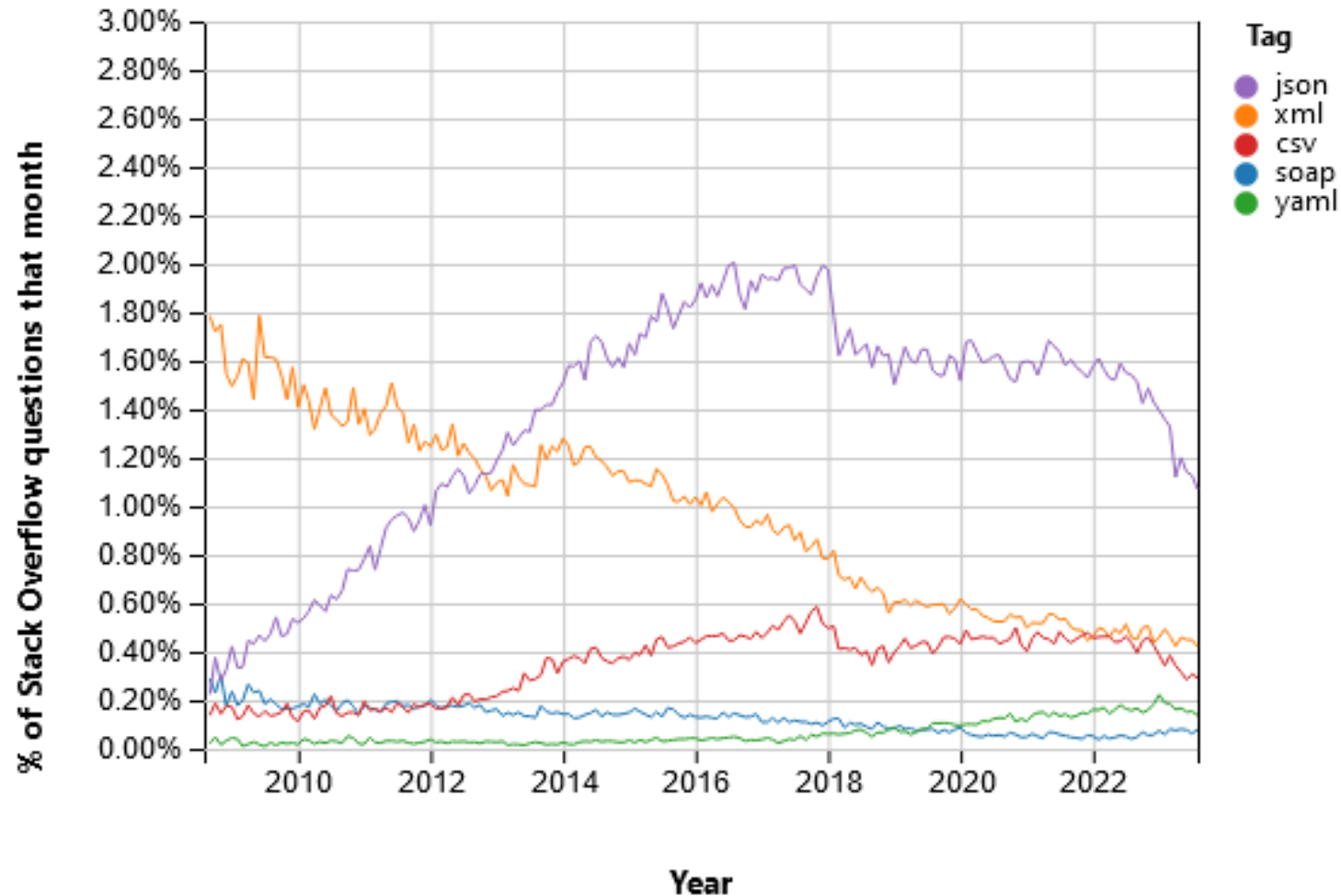


Resumo das diferenças: YAML versus JSON

	JSON	YAML
O que é isso?	Um formato de serialização de dados para trocar dados estruturados entre aplicações e serviços de software. Prioriza o uso de aplicações em relação ao uso humano.	Um formato de serialização de dados para trocar dados estruturados entre aplicações e serviços de software. Prioriza o uso humano em relação ao uso de aplicações.
Principais casos de uso	Difundido em plataformas, linguagens, comunicações de software distribuídas, aplicações Web, arquivos de configuração e APIs.	Arquivos de configuração em várias ferramentas e serviços de automação, DevOps e infraestrutura como código (IaC).
Legibilidade	Fácil.	A mais fácil.
Tipos de dados	Número, booleano, nulo, string, matriz e objeto.	Oferece suporte a todos os tipos de dados por meio da coleta de dados aninhados que incluem sequências, escalares e mapeamentos.
Suporta comentários	Não.	Sim.
Suporta objetos de dados como valores	Sim.	Não.

<https://aws.amazon.com/pt/compare/the-difference-between-yaml-and-json/>

Por que JSON???



<https://insights.stackoverflow.com/trends?tags=soap%2Cxml%2Cyaml%2Ccsv%2Cjson>

JSON - JavaScript Object Notation

JSON - JavaScript Object Notation

- **JSON** - Notação de Objeto JavaScript
- Formato de dados leve e fácil de ler que é amplamente utilizado para trocar informações entre sistemas, especialmente na web.
- Muito popular devido à sua simplicidade e facilidade de uso.
- O JSON foi “criado” como um formato de intercâmbio de dados por **Douglas Crockford**.

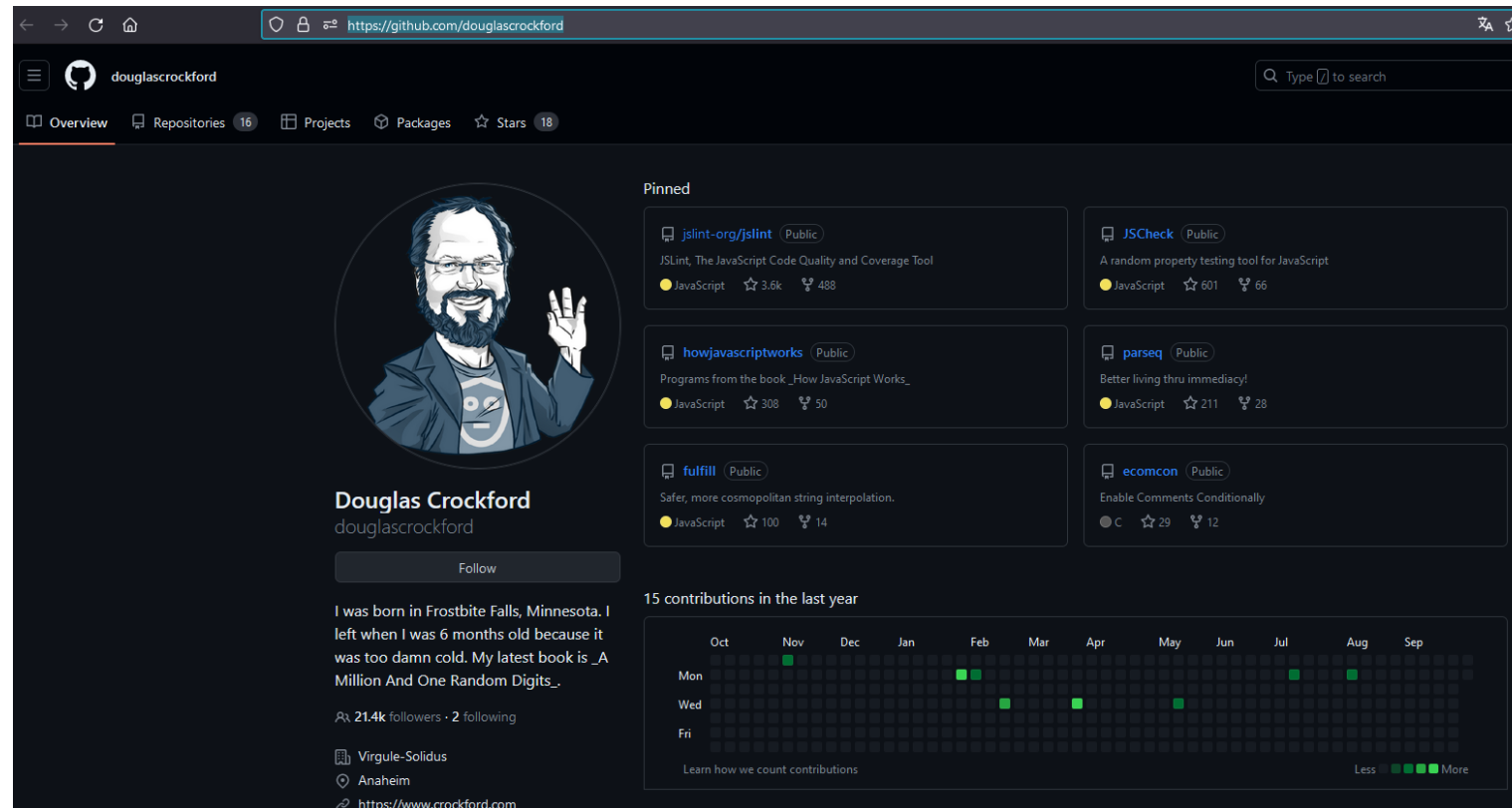
JSON – Resumo da história e evolução

- **Década de 1990:** Influenciado pelo padrão de notação de objeto literário JavaScript.
- **2001:** Douglas Crockford formalizou o formato JSON → "The JSON Saga".
- **RFC 4627 em 2006:** Padronizado pela IETF (Internet Engineering Task Force) no RFC 4627.
- **Amplamente Adotado:** A simplicidade do JSON, juntamente com a crescente popularidade da web e das tecnologias da web, o tornou uma escolha popular para trocar dados entre aplicativos, especialmente em serviços da web, APIs REST e AJAX.
- **Suporte em Linguagens de Programação:** A maioria das linguagens de programação modernas oferece suporte nativo ou bibliotecas para trabalhar com JSON.
- **JSON-LD e Outros Formatos:** Outras variantes do JSON surgiram para atender a necessidades específicas. Por exemplo, o JSON-LD (JSON Linked Data) é uma extensão que permite a representação de dados semânticos para a Web Semântica.
- **Aplicações Além da Web:** Além de ser amplamente utilizado na web, o JSON também é utilizado em muitos outros contextos, como armazenamento de configurações, troca de dados em sistemas embutidos, armazenamento de registros de log, entre outros.

Douglas Crockford




- <https://www.crockford.com/>
- <https://github.com/douglascrockford>
- <https://www.json.org/>

A screenshot of Douglas Crockford's GitHub profile page. The page shows his profile picture, name, and bio. It also displays a list of pinned repositories, including jslint-org/jslint, JSCheck, howjavascriptworks, parseq, fulfill, and ecomcon. At the bottom, there is a contribution graph showing 15 contributions in the last year.

← → ↻ 🏠 <https://github.com/douglascrockford> 🔍 Type to search

📁 Overview 📁 Repositories 16 📁 Projects 📁 Packages ⭐ Stars 18

 **Douglas Crockford**
douglascrockford







Follow

I was born in Frostbite Falls, Minnesota. I left when I was 6 months old because it was too damn cold. My latest book is *_A Million And One Random Digits_*.

👤 21.4k followers · 2 following

📁 Virgule-Solidus
📍 Anaheim
<https://www.crockford.com>


Pinned

-  **jslint-org/jslint** (Public)
JSLint, The JavaScript Code Quality and Coverage Tool
🟡 JavaScript ⭐ 3.6k 🍷 488
-  **JSCheck** (Public)
A random property testing tool for JavaScript
🟡 JavaScript ⭐ 601 🍷 66
-  **howjavascriptworks** (Public)
Programs from the book *_How JavaScript Works_*
🟡 JavaScript ⭐ 308 🍷 50
-  **parseq** (Public)
Better living thru immediacy!
🟡 JavaScript ⭐ 211 🍷 28
-  **fulfill** (Public)
Safer, more cosmopolitan string interpolation.
🟡 JavaScript ⭐ 100 🍷 14
-  **ecomcon** (Public)
Enable Comments Conditionally
🟡 C ⭐ 29 🍷 12

15 contributions in the last year

	Oct	Nov	Dec	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep
Mon												
Wed												
Fri												

Learn how we count contributions

Less  More

I Discovered JSON

- I do not claim to have invented JSON.
It already existed in nature.
- I do not claim to have been the first to discover it.
- I gave it a specification and a little website.
- The rest happened by itself.

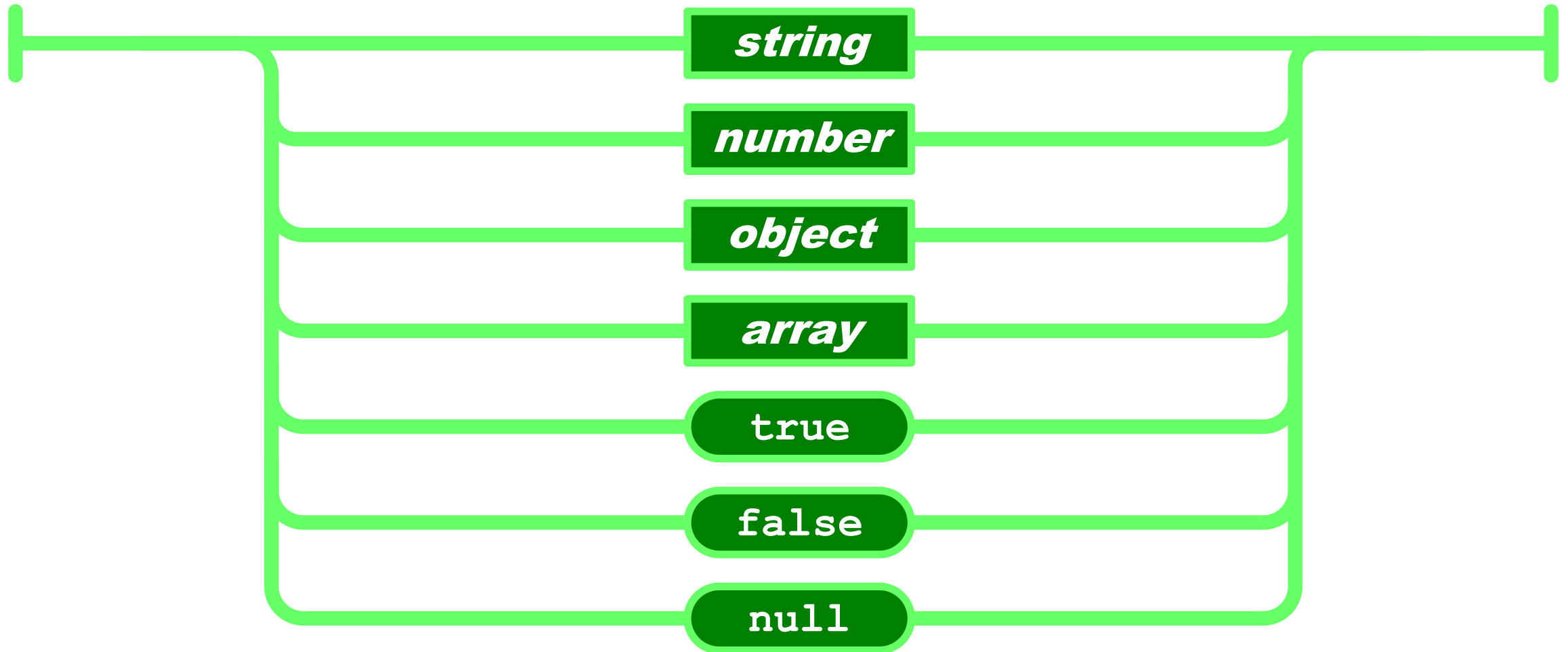
Sintaxe JSON

- Coleção de pares chave-valor.
- Os dados são armazenados em objetos JSON, que são delimitados por chaves {}.
- Cada par chave-valor é separado por vírgula e as **chaves** são strings que devem estar entre aspas duplas.

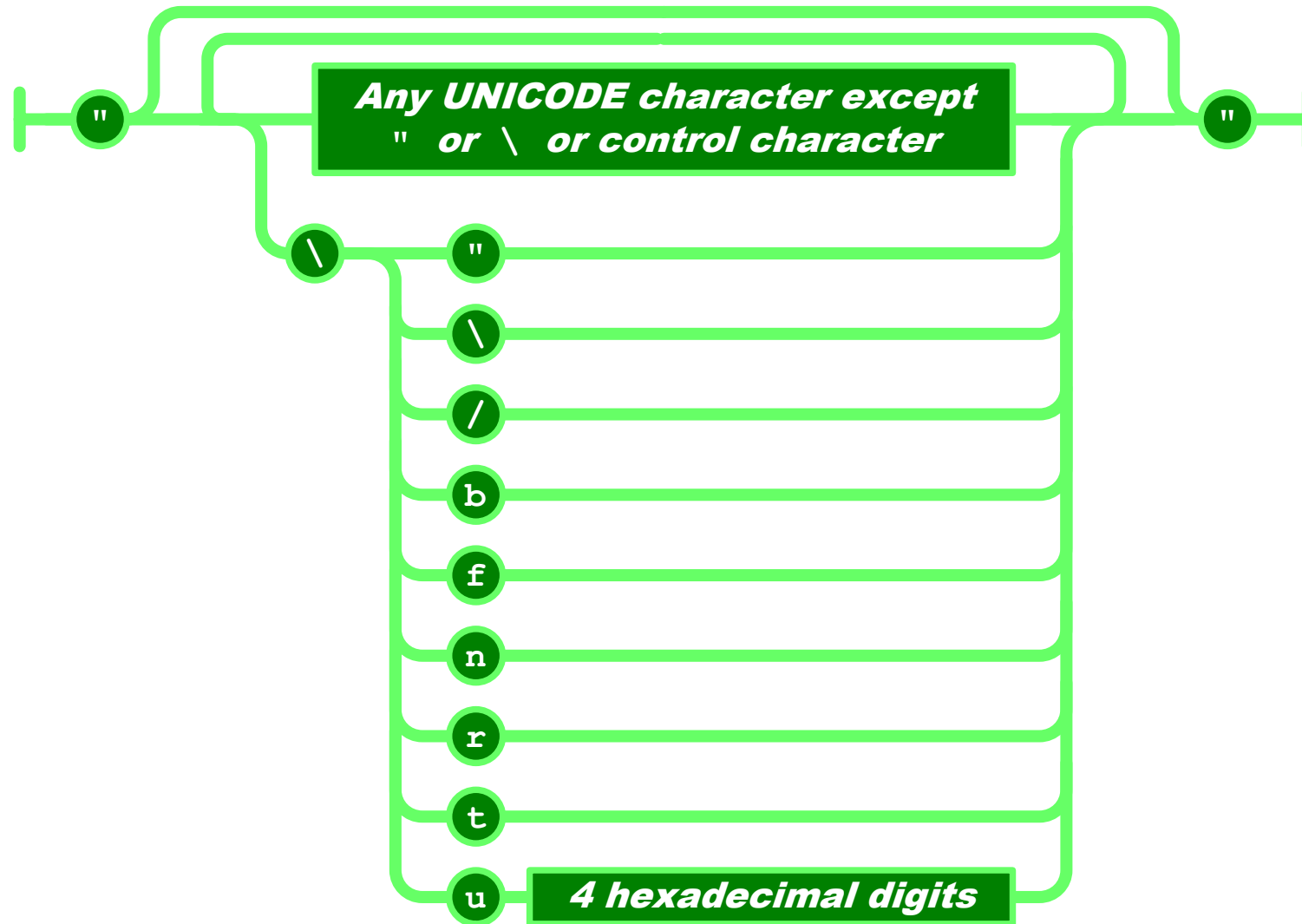
Exemplo Objeto JSON

```
1  {
2    "name": "Luke Skywalker",
3    "height": "172",
4    "mass": "77",
5    "hair_color": "blond",
6    "skin_color": "fair",
7    "eye_color": "blue",
8    "birth_year": "19BBY",
9    "gender": "male",
10   "homeworld": "https://swapi.dev/api/planets/1/",
11   "films": [
12     "https://swapi.dev/api/films/2/",
13     "https://swapi.dev/api/films/6/",
14     "https://swapi.dev/api/films/3/",
15     "https://swapi.dev/api/films/1/",
16     "https://swapi.dev/api/films/7/"
17   ],
18   "species": [
19     "https://swapi.dev/api/species/1/"
20   ],
21   "vehicles": [
22     "https://swapi.dev/api/vehicles/14/",
23     "https://swapi.dev/api/vehicles/30/"
24   ],
25   "starships": [
26     "https://swapi.dev/api/starships/12/",
27     "https://swapi.dev/api/starships/22/"
28   ],
29   "created": "2014-12-09T13:50:51.644000Z",
30   "edited": "2014-12-20T21:17:56.891000Z",
31   "url": "https://swapi.dev/api/people/1/"
32 }
```

Value



String

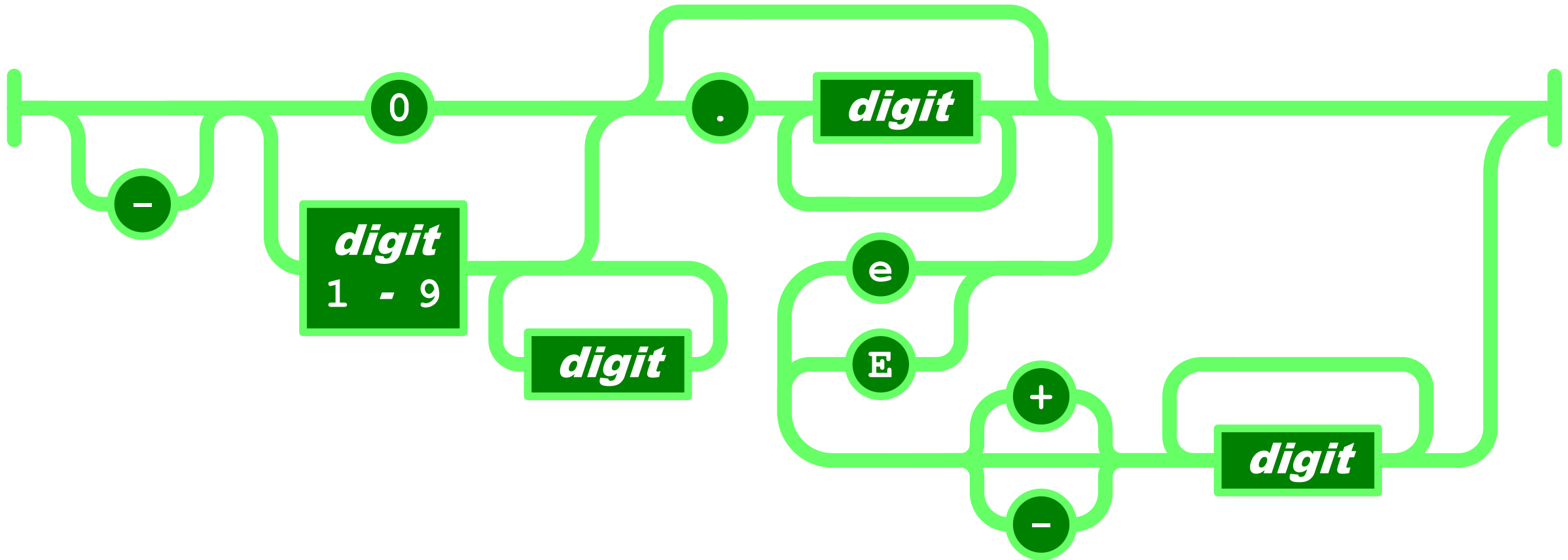


Tipos de Dados: String

- Uma sequência de caracteres entre aspas duplas.

```
"name": "Luke Skywalker",  
"height": "172",  
"mass": "77",  
"hair_color": "blond",  
"skin_color": "fair",  
"eye_color": "blue",  
"birth_year": "19BBY",
```

Number

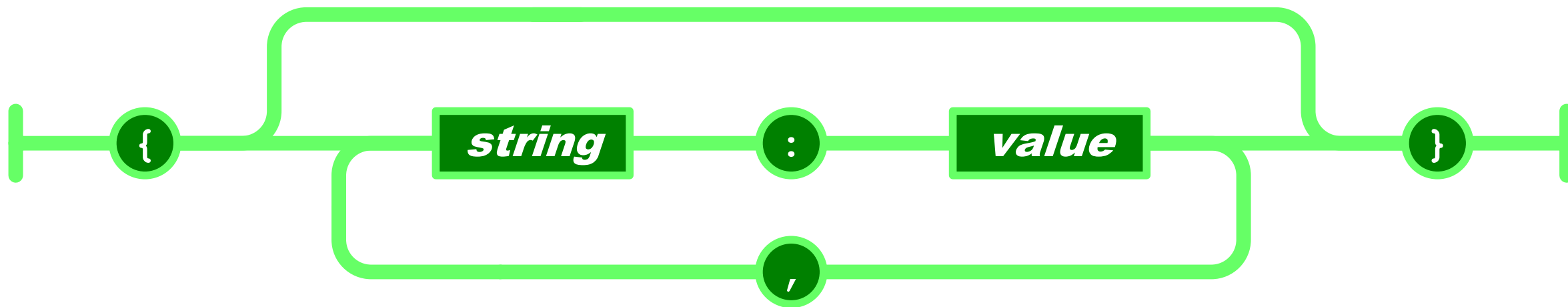


Tipos de Dados: Número

- Um número inteiro ou de ponto flutuante.

```
"height": 1.72,  
"mass": 77,
```

Object

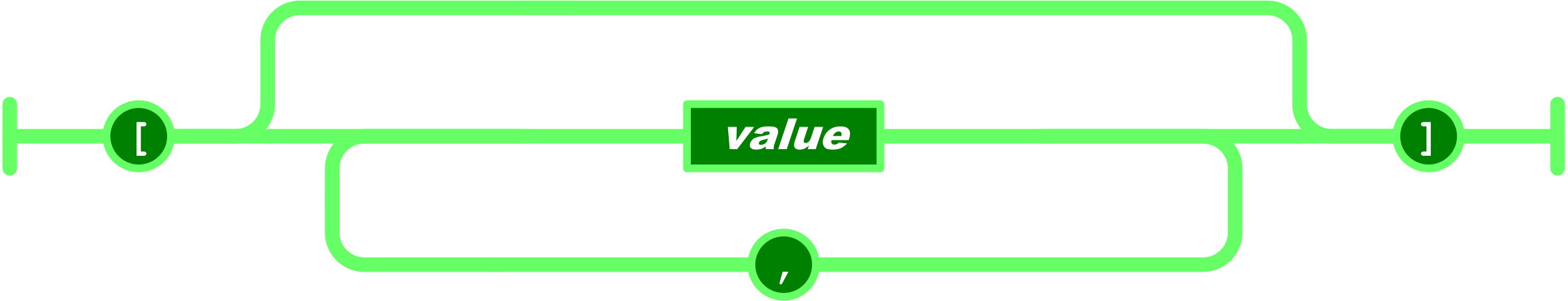


Tipos de Dados: Objeto

- Um conjunto não ordenado de pares chave-valor, separados por vírgulas e delimitados por chaves.

```
1  {  
2    "name": "Han Solo",  
3    "height": "180",  
4    "mass": "80",  
5    "hair_color": "brown",  
6    "skin_color": "fair",  
7    "eye_color": "brown",  
8    "birth_year": "29BBY",  
9    "gender": "male",  
10   "homeworld": "https://swapi.dev/api/planets/22/",  
11   "starship_main": {  
12     "name": "Millennium Falcon",  
13     "model": "YT-1300 light freighter",  
14     "manufacturer": "Corellian Engineering Corporation",  
15     "cost_in_credits": "100000",  
16     "length": "34.37",  
17     "max_atmosphering_speed": "1050"  
18   }  
19 }
```

Array



Tipos de Dados: Vetores

- Uma coleção ordenada de valores, separados por vírgulas e delimitados por colchetes

```
1 {  
2   "name": "Han Solo",  
3   "height": "180",  
4   "mass": "80",  
5   "starships": [  
6     {  
7       "name": "Millennium Falcon",  
8       "model": "YT-1300 light freighter",  
9       "manufacturer": "Corellian Engineering Corporation",  
10      "cost_in_credits": "100000",  
11      "length": "34.37",  
12      "max_atmosphering_speed": "1050"  
13    },  
14    {  
15      "name": "Imperial shuttle",  
16      "model": "Lambda-class T-4a shuttle",  
17      "manufacturer": "Sienar Fleet Systems",  
18      "cost_in_credits": "240000",  
19      "length": "20",  
20      "max_atmosphering_speed": "850"  
21    }  
22  ]  
23 }
```

Tipos de Dados: Boolean

- Pode ser **true** ou **false**
- Sem aspas (“)

```
1 {  
2   "name": "Luke Skywalker",  
3   "height": "172",  
4   "mass": "77",  
5   "single": true  
6 }
```


Tipos de Dados: Nulo

- Representado pela palavra-chave **null**
- Sem aspas (“)
- Usado para indicar que um valor está ausente ou indefinido.

```
1  ✓ {  
2    "name": "C-3PO",  
3    "height": "167",  
4    "mass": "75",  
5    "hair_color": null,  
6    "skin_color": "gold",  
7    "eye_color": "yellow",  
8    "birth_year": "112BBY",  
9    "gender": null  
10 }
```