



Universidade do Minho  
Escola de Engenharia  
Departamento de Informática

# Geração de Dados Sintéticos a partir de Schemas JSON

---

Hugo André Coelho Cardoso  
José Carlos Leite Ramalho

# Motivação

## JSON Schema

```
{
  "$id": "https://exemplo.com/schemas/cliente",
  "$schema": "https://json-schema.org/draft/2020-12/schema",

  "type": "object",
  "properties": {
    "nome": { "type": "string" },
    "apelido": { "type": "string" },
    "morada_envio": { "$ref": "/schemas/morada" },
    "morada_faturacao": { "$ref": "/schemas/morada" }
  },
  "required": ["nome", "apelido", "morada_envio", "morada_faturacao"],

  "$defs": {
    "morada": {
      "$id": "/schemas/morada",
      "$schema": "https://json-schema.org/draft/2020-12/schema",

      "type": "object",
      "properties": {
        "rua": { "type": "string" },
        "cidade": { "type": "string" },
        "distrito": { "$ref": "#/$defs/distrito" }
      },
      "required": ["rua", "cidade", "distrito"],

      "$defs": {
        "distrito": { "enum": ["Aveiro", "Beja", "... etc ..."] }
      }
    }
  }
}
```

## Documentos JSON

```
{
  "nome": "Bruno Leonel",
  "apelido": "Garrizo",
  "morada_envio": {
    "rua": "Rua de Trás da Poça, nº20",
    "cidade": "Braga",
    "distrito": "Braga"
  },
  "morada_faturacao": {
    "rua": "Rua de Trás da Poça, nº20",
    "cidade": "Braga",
    "distrito": "Braga"
  }
}
```



```
{
  "nome": "Tomás",
  "apelido": "Goulão",
  "morada_envio": {
    "rua": "Rua dos Álamos, nº31",
    "cidade": "Almada",
    "distrito": "Setúbal"
  },
  "morada_faturacao": {
    "rua": "Rua da Alegria, nº40",
    "cidade": "Almada",
    "distrito": "Setúbal"
  }
}
```



```
{
  "nome": "Mafalda",
  "apelido": "Costa",
  "morada_faturacao": {
    "rua": "Rua António Pedro Pares, nº 2",
    "cidade": "Lisboa",
    "distrito": "Lisboa"
  }
}
```



```
{
  "nome": "Dayan Moisés",
  "apelido": "Rôla",
  "morada_envio": {
    "rua": "Rua Chetas Anta, nº52",
    "cidade": "Espinho",
    "distrito": "Aveiro"
  },
  "morada_faturacao": {
    "rua": 52,
    "cidade": "Espinho",
    "distrito": "Aveiro"
  }
}
```



```
{
  "nome": "Ignácio",
  "apelido": "Mengo",
  "morada_envio": null,
  "morada_faturacao": null
}
```



# Contextualização

DataGen

Gerar JSON XML CSV

Nome: dataset

Download Gerar API Download API

```
1 <?LANGUAGE pt>
2 {
3   perfil: [
4     'repeat(3)': {
5       nome: '{{fullName()}}',
6       idade: '{{integer(15,60)}}',
7       or() {
8         BI: '{{integerOfSize(8)}}-{{integer(0,9)}}',
9         CC: '{{integerOfSize(8)}}-{{integer(0,9)}}-{{letter("uppercase")}}{{letter("uppercase")}}{{integer(0,9)}}'
10      },
11      having(70) {
12        descrição: '{{lorem("sentences", 1)}}'
13      },
14      nr_livros: '{{integer(1,10)}}',
15      livros: [ 'repeat(this.nr_livros)': {
16        titulo: '{{lorem("words",1)}}',
17        suporte: '{{random("Físico","Digital")}}',
18        rating: '{{integer(1,5)}}'
19      } ],
20      livro_favorito(gen) {
21        var titulos = this.livros.map(x => x.titulo)
22        return gen.random(...titulos)
23      }
24    ]
25  }
26 }
```

```
1 {
2   "perfil": [
3     {
4       "nome": "Haiden Garrim",
5       "idade": 49,
6       "CC": "42204091-8-SG0",
7       "descrição": "Magna do esse id nulla deserunt sint",
8       "nr_livros": 3,
9       "livros": [
10        {
11          "titulo": "cillum",
12          "suporte": "Digital",
13          "rating": 1
14        },
15        {
16          "titulo": "eiusmod",
17          "suporte": "Digital",
18          "rating": 2
19        },
20        {
21          "titulo": "do",
22          "suporte": "Físico",
23          "rating": 5
24        }
25      ],
26      "livro_favorito": "cillum"
27    },
28    {
29      "nome": "Devlin Retre",
30      "idade": 54,
31      "CC": "94440867-4-JB6",
32      "nr_livros": 4,
33      "livros": [
34        {
35          "titulo": "non",
36          "suporte": "Físico",
37          "rating": 1
38        },
39        {
40          "titulo": "laborum",
41          "suporte": "Digital",
42          "rating": 5
43        }
44      ]
45    }
46  ]
47 }
```

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <perfil>
3   <elem_1>
4     <nome>
5       Ben Olas
6     </nome>
7     <idade>
8       31
9     </idade>
10    <CC>
11      23786290-7-QM5
12    </CC>
13    <nr_livros>
14      2
15    </nr_livros>
16    <livros>
17      <elem_1>
18        <titulo>
19          ullamco
20        </titulo>
21        <suporte>
22          Físico
23        </suporte>
24        <rating>
25          4
26        </rating>
27      </elem_1>
28      <elem_2>
29        <titulo>
30          id
31        </titulo>
32        <suporte>
33          Digital
34        </suporte>
35        <rating>
36          4
37        </rating>
38      </elem_2>
39    </livros>
40    <livro_favorito>
41      id
42    </livro_favorito>
43  </elem_1>
44  </perfil>
45</xml>
```

```
1 <!--LANGUAGE pt>
2 {
3   perfil: [
4     'repeat(3)': {
5       nome: '{{fullName()}}',
6       idade: '{{integer(15,60)}}',
7       or() {
8         BI: '{{integerOfSize(8)}}-{{integer(0,9)}}',
9         CC: '{{integerOfSize(8)}}-{{integer(0,9)}}-{{letter("uppercase")}}{{letter("uppercase")}}{{integer(0,9)}}'
10      },
11     having(70) {
12       descrição: '{{lorem("sentences", 1)}}'
13     },
14     nr_livros: '{{integer(1,10)}}',
15     livros: [ 'repeat(this.nr_livros)': {
16       titulo: '{{lorem("words",1)}}',
17       suporte: '{{random("Físico","Digital")}}',
18       rating: '{{integer(1,5)}}'
19     } ],
20     livro_favorito(gen) {
21       var titulos = this.livros.map(x => x.titulo)
22       return gen.random(...titulos)
23     }
24   ]
25 }
26 }
```

```
1 <!LANGUAGE pt>
2 {
3   perfil: [
4     'repeat(3)': {
5       nome: '{{fullName()}}',
6       idade: '{{integer(15,60)}}',
7       or() {
8         BI: '{{integerOfSize(8)}}-{{integer(0,9)}}',
9         CC: '{{integerOfSize(8)}}-{{integer(0,9)}}-{{letter("uppercase")}}{{letter("uppercase")}}{{integer(0,9)}}'
10      },
11     having(70) {
12       descrição: '{{lorem("sentences", 1)}}',
13     },
14     nr_livros: '{{integer(1,10)}}',
15     livros: [ 'repeat(this.nr_livros)': {
16       titulo: '{{lorem("words",1)}}',
17       suporte: '{{random("Físico","Digital")}}',
18       rating: '{{integer(1,5)}}'
19     } ],
20     livro_favorito(gen) {
21       var titulos = this.livros.map(x => x.titulo)
22       return gen.random(...titulos)
23     }
24   ]
25 }
26 }
```

object

integer

string

array

```
1 missing(50) { prop1: 1, prop2: 2 },
2 having(80) { prop3: 3 }
```

```
1 at_least(2) {
2   prop1: 1,
3   prop2: 2,
4   prop3: 3
5 }
```

```
1 tipo: '{{random("A","B","C")}}',
2 if (this.tipo == "A") { A: "tipo A" }
3 else if (this.tipo == "B") { B: "tipo B" }
4 else { C: "tipo C" }
```

JS actors.js

JS animals.js

JS brands.js

JS buzzwords.js

JS capitals.js

JS cars.js

JS cities.js

JS continents.js

JS countries.js

JS cultural\_centers.js

JS gov\_entities.js

JS hackers.js

JS jobs.js

JS months.js

JS musicians.js

JS names.js

JS nationalities.js

JS political\_parties.js

JS pt\_businessmen.js

JS pt\_districts.js

JS pt\_entities.js

JS pt\_political\_parties.js

JS pt\_politicians.js

JS pt\_public\_figures.js

JS pt\_top100\_celebrities.js

JS religions.js

JS soccer\_clubs.js

JS soccer\_players.js

JS sports.js

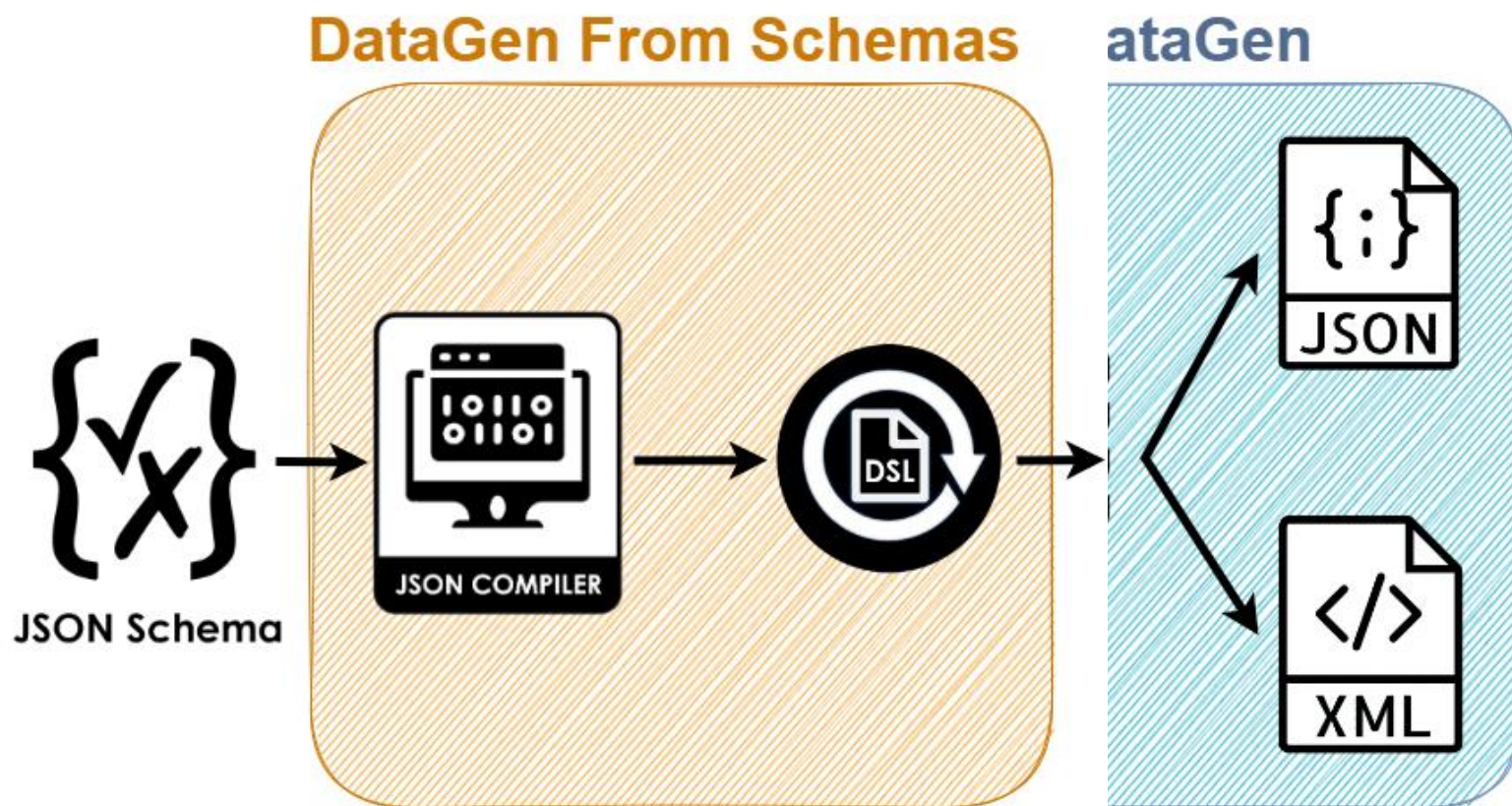
JS top100\_celebrities.js

JS weekdays.js

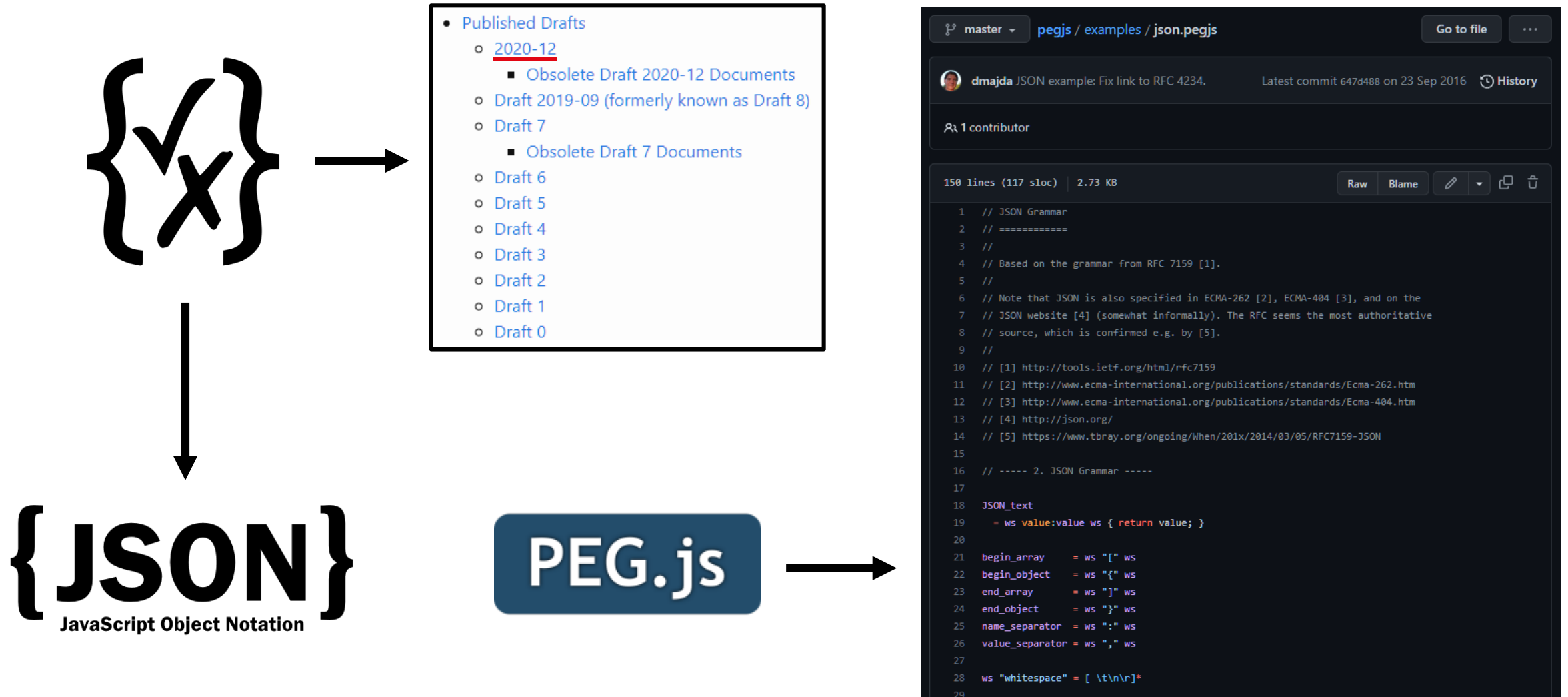
JS writers.js



## Fluxo de trabalho



# JSON Schema



# Gramática

- Restrição do valor de cada chave estritamente ao seu espaço léxico

## boolean

uniqueItems  
unevaluatedItems  
additionalProperties  
...

## array

enum  
anyOf  
prefixItems  
...

## schema

not  
items  
propertyNames  
...

...

- Rotina rigorosa de validação semântica da schema

```
{  
  "type": "number",  
  "minimum": 50,  
  "maximum": 20  
}
```

15 ✖  
35 ✖

70 ✖

- Relatórios de erros

```
morada + ↗  
1 {  
2   "$id": "https://datagen.di.uminho.pt/json-schemas/morada",  
3  
4   "type": "map",  
5   "properties": {  
6     "rua": { "type": "string" },  
7     "cidade": { "type": "string" },  
8     "distrito": { "type": "string" }  
9   },  
10  "required": ["rua", "cidade", "distrito"]  
11 }
```



## Existe um erro no modelo!

Era esperado "array", "boolean", "integer", "null", "number", "object" ou "string" mas foi encontrado "m".

- **Início:** Linha 4, Coluna 12
- **Fim:** Linha 4, Coluna 13



# Estrutura Intermédia

- string
  - minLength / maxLength
  - pattern
  - format
- Numeric types (integer / number)
  - multipleOf
  - minimum / exclusiveMinimum
  - maximum / exclusiveMaximum
- object
  - properties
  - patternProperties
  - additionalProperties
  - unevaluatedProperties
  - required
  - propertyNames
  - minProperties / maxProperties
- array
  - items
  - prefixItems
  - unevaluatedItems
  - contains / minContains / maxContains
  - minItems / maxItems
  - uniqueItems
- boolean
- null

```
{
  "oneOf": [
    { "type": "number", "multipleOf": 5 },
    { "type": "string", "minLength": 10 },
    {
      "type": ["number", "string"],
      "multipleOf": 3,
      "maxLength": 20
    }
  ]
}
```

```
{
  "oneOf": [
    { "type": "number", "multipleOf": 5 },
    { "type": "number", "multipleOf": 3 },
    { "type": "string", "minLength": 10 },
    { "type": "string", "maxLength": 20 }
  ]
}
```

```
{
  "number": {
    "oneOf": [
      { "multipleOf": "5" },
      { "multipleOf": "3" }
    ]
  },
  "string": {
    "oneOf": [
      { "minLength": "10" },
      { "maxLength": "20" }
    ]
  }
}
```

```
{
  "type": ["string", "number"],
  "maxLength": 10,
  "maximum": 90,
  "allOf": [
    { "type": "number", "multipleOf": 3 },
    { "type": "number", "multipleOf": 5 }
  ]
}
```

```
{
  "number": {
    "maximum": 90,
    "allOf": [
      { "type": "number", "multipleOf": 3 },
      { "type": "number", "multipleOf": 5 }
    ]
  }
}
```

# DataGen From Schemas

DataGen From

XMLJSONSchemas

DATAGENAPIENTRAR

GERAR

MODELO INTERMÉDIO

XMLJSON

encomenda

DOWNLOAD

morada

cliente

produto

encomenda

+

```
1 {
2   "$id": "https://datagen.di.uminho.pt/schemas/encomenda",
3   "$schema": "https://json-schema.org/draft/2020-12/schema",
4
5   "type": "object",
6   "properties": {
7     "estado": { "enum": ["Sem stock", "A caminho da loja", "Por levantar"] },
8     "data_entrega": { "type": "string", "_datagen": "date('20-07-2022','23-07-2022','DD-MM-AAAA')",
9     "localizacao_atual": { "type": "string", "_datagen": "position",
10    "produto": { "$ref": "/schemas/produto" },
11    "cliente": { "$ref": "/schemas/cliente" }
12  },
13  "required": ["estado", "data_entrega", "localizacao_atual", "produto", "cliente"]
14 }
```

cliente

encomenda

encomenda

encomenda

x

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <encomenda>
3   <estado>A caminho da loja</estado>
4   <data_entrega>20-07-2022</data_entrega>
5   <localizacao_atual>(-6.02713, -10.80054)</localizacao_atual>
6   <produto>
7     <codigo>2781935073</codigo>
8     <nome>Veniam co</nome>
9     <preco>225.337</preco>
10    <stock>-53</stock>
11  </produto>
12  <cliente>
13    <nome>Teodora Elisa</nome>
14    <apelido>Monte</apelido>
15    <morada_envio>
16      <rua>Incididunt et incididunt laborum nostrud velit ullamco cu</rua>
17      <freguesia>Vale (São Martinho)</freguesia>
18      <cidade>Cascais</cidade>
19    </morada_envio>
20    <morada_faturacao>
21      <rua>Sit ullamco est mollit ad </rua>
22      <freguesia>Cótimos</freguesia>
23      <cidade>Almodôvar</cidade>
24    </morada_faturacao>
25    <contacto>184010062</contacto>
26  </cliente>
27 </encomenda>
```

# DataGen From Schemas

DataGen From

XMLJSONSchemas

DATAGENAPIENTRAR

GERAR

MODELO INTERMÉDIO

XMLJSON

encomenda

DOWNLOAD

morada | cliente | produto | encomenda

```
1 {
2   "$id": "https://datagen.di.uminho.pt/schemas/encomenda",
3   "$schema": "https://json-schema.org/draft/2020-12/schema",
4
5   "type": "object",
6   "properties": {
7     "estado": { "enum": ["Sem stock", "A caminho da loja", "Por levantar"] },
8     "data_entrega": { "type": "string", "_datagen": "date('20-07-2022','23-07-2022','DD-MM-AAAA')",
9     "localizacao_atual": { "type": "string", "_datagen": "position",
10    "produto": { "$ref": "/schemas/produto" },
11    "cliente": { "$ref": "/schemas/cliente" }
12  },
13  "required": ["estado", "data_entrega", "localizacao_atual", "produto", "cliente"]
14 }
```

cliente | encomenda | encomenda

```
1 {
2   "estado": "Por levantar",
3   "data_entrega": "22-07-2022",
4   "localizacao_atual": "(12.27616, -29.59108)",
5   "produto": {
6     "codigo": 8637902916,
7     "nome": "Anim tempor eiusmod culpa non. Quis cillum aute culpa excepteur proident irure irure labori",
8     "preco": 158.435,
9     "stock": -76
10  },
11  "cliente": {
12    "nome": "Leonor Yara",
13    "apelido": "Frances",
14    "morada_envio": {
15      "rua": "Laborum dolor es",
16      "freguesia": "Pedrógão",
17      "cidade": "Arouca"
18    },
19    "morada_faturacao": {
20      "rua": "Excepteur occaecat id sint sit Lorem sit sint. Exce",
21      "freguesia": "Agrochão",
22      "cidade": "Valado de Frades"
23    },
24    "contacto": 770948290
25  }
26 }
```

# DataGen From Schemas

DataGen From

XMLJSONSchemas

DATAGENAPIENTRAR

GERAR

MODELO INTERMÉDIO

XMLJSON

encomenda

DOWNLOAD

morada | cliente | produto | encomenda

cliente | encomenda | encomenda

```
1 {
2   "$id": "https://datagen.di.uminho.pt/schemas/encomenda",
3   "$schema": "https://json-schema.org/draft/2020-12/schema",
4
5   "type": "object",
6   "properties": {
7     "estado": { "enum": ["Sem stock", "A caminho da loja", "Por levantar"] },
8     "data_entrega": { "type": "string", "_datagen": "date('20-07-2022','23-07-2022','DD-MM-AAAA') " },
9     "localizacao_atual": { "type": "string", "_datagen": "position" },
10    "produto": { "$ref": "/schemas/produto" },
11    "cliente": { "$ref": "/schemas/cliente" }
12  },
13  "required": ["estado", "data_entrega", "localizacao_atual", "produto", "cliente"]
14 }
```

```
1 {
2   "estado": "Por levantar",
3   "data_entrega": "22-07-2022",
4   "localizacao_atual": "(12.27616, -29.59108)",
5   "produto": {
6     "codigo": 8637902916,
7     "nome": "Anim tempor eiusmod culpa non. Quis cillum aute culpa excepteur proident irure irure labori",
8     "preco": 158.435,
9     "stock": -76
10  },
11  "cliente": {
12    "nome": "Leonor Yara",
13    "apelido": "Frances",
14    "morada_envio": {
15      "rua": "Laborum dolor es",
16      "freguesia": "Pedrógão",
17      "cidade": "Arouca"
18    },
19    "morada_faturacao": {
20      "rua": "Excepteur occaecat id sint sit Lorem sit sint. Exce",
21      "freguesia": "Agrochão",
22      "cidade": "Valado de Frades"
23    },
24    "contacto": 770948290
25  }
26 }
```

# DataGen From Schemas

DataGen From XML JSON Schemas

DATAGEN API ENTRAR

GERAR MODELO INTERMÉDIO

XML JSON encomenda DOWNLOAD

morada cliente produto encomenda

```
1 {
2   "$id": "https://datagen.di.uminho.pt/schemas/encomenda",
3   "$schema": "https://json-schema.org/draft/2020-12/schema",
4
5   "type": "object",
6   "properties": {
7     "estado": { "enum": ["Sem stock", "A caminho da loja", "Por levantar"] },
8     "data_entrega": { "type": "string", "_datagen": "date('20-07-2022','23-07-2022','DD-MM-AAAA') " },
9     "localizacao_atual": { "type": "string", "_datagen": "position" },
10    "produto": { "$ref": "/schemas/produto" },
11    "cliente": { "$ref": "/schemas/cliente" }
12  },
13  "required": ["estado", "data_entrega", "localizacao_atual", "produto", "cliente"]
14 }
```

cliente encomenda encomenda

```
1 {
2   "estado": "Por levantar",
3   "data_entrega": "22-07-2022",
4   "localizacao_atual": "(12.27616, -29.59108)",
5   "produto": {
6     "codigo": 8637902916,
7     "nome": "Anim tempor eiusmod culpa non. Quis cillum aute culpa excepteur proident irure irure labori",
8     "preco": 158.435,
9     "stock": -76
10  },
11  "cliente": {
12    "nome": "Leonor Yara",
13    "apelido": "Frances",
14    "morada_envio": {
15      "rua": "Laborum dolor es",
16      "freguesia": "Pedrógão",
17      "cidade": "Arouca"
18    },
19    "morada_faturacao": {
20      "rua": "Excepteur occaecat id sint sit Lorem sit sint. Exce",
21      "freguesia": "Agrochão",
22      "cidade": "Valado de Frades"
23    },
24    "contacto": 770948290
25  }
26 }
```

# DataGen From Schemas

The screenshot shows the DataGen From Schemas web application. The interface includes a sidebar with tabs for 'XML', 'JSON', and 'Schemas'. The main content area is titled 'Rotas Aplicacionais' and displays a list of API endpoints. A modal window is open, showing detailed documentation for the POST /api/json\_schema/json endpoint.

**Rotas Aplicacionais**

**JSON SCHEMA**

- POST /api/json\_schema/json
- POST /api/json\_schema/xml

Estas rotas retornam um objeto JSON com o dataset no formato indicado, dado em forma de string, e o respetivo modelo da DSL do DataGen, gerados a partir de uma schema JSON.

O corpo do pedido deve ter apenas três propriedades: **main\_schema**, **other\_schemas** e **settings**, sendo que **main\_schema** é a schema a partir da qual se pretende gerar o dataset, **other\_schemas** é um array com as restantes schemas e tanto a **main\_schema** como as restantes devem estar em forma de objeto JSON. As definições devem ser enviadas num objeto com as seguintes propriedades:

- **datagen\_language** - língua dos resultados de funções (com suporte de datasets) do DataGen; deve ser "pt" (português) ou "en" (inglês).
- **recursivity** - objeto com os limites de recursividade, deve possuir as seguintes propriedades:
  - **lower** - limite inferior de recursividade; deve ser um inteiro não-negativo.
  - **upper** - limite superior de recursividade; deve ser um inteiro não-negativo.
- **prob\_if** - probabilidade de a schema da condição validar a instância; deve ser um número entre 0 e 100.
- **prob\_patternProperty** - probabilidade de gerar uma propriedade a partir de uma *patternProperty*; deve ser um número entre 0 e 100.
- **random\_props** - possibilidade de gerar propriedades aleatórias (dentro do tamanho indicado para o objeto) se não forem especificadas *additionalProperties* nem *unevaluatedProperties*; deve ser um booleano.
- **extend\_objectProperties** - modo de proceder ao estender propriedades repetidas nas chaves *properties* e *patternProperties*; deve ser uma das seguintes strings:
  - **extend/overwrite** - se as chaves tiverem propriedades repetidas, estende/substitui a schema de cada propriedade da chave-base com a respetiva schema da mesma propriedade da chave nova. Todas as propriedades originais da nova chave são atribuídas à chave-base.
- **extend\_schemaProperties** - modo de proceder ao estender chaves cujo valor é uma subschema (*propertyNames*, *additionalProperties*, *unevaluatedProperties*, *items* ou *unevaluatedItems*); deve ser uma das seguintes strings:
  - **extend/overwrite** - estende/substitui a schema da chave-base com a schema da chave nova.
- **extend\_prefixItems** - modo de proceder ao estender a chave *prefixItems*; deve ser uma das seguintes strings:
  - **extend** - para todas as schemas que se encontram no mesmo índice, estende as da chave-base com as respetivas schemas da nova chave. Se a chave nova tiver mais elementos do que a base, os elementos extra são também concatenados.
  - **append** - os elementos do novo *prefixItems* são concatenados aos da chave-base.



# Exemplo 1

```
morada x cliente x produto x encomenda x + ↕
1 {
2   "$id": "https://datagen.di.uminho.pt/schemas/encomenda",
3   "$schema": "https://json-schema.org/draft/2020-12/schema",
4   "type": "object",
5   "properties": {
6     "estado": { "enum": ["Sem stock", "A caminho da loja", "Por levantar"] },
7     "data_entrega": { "type": "string", "_datagen": "date('20-07-2022','23-07-2022','DD-MM-AAAA') " },
8     "localizacao_atual": { "type": "string", "_datagen": "position" },
9     "produto": { "$ref": "/schemas/produto" },
10    "cliente": { "$ref": "/schemas/cliente" }
11  },
12  "required": ["estado", "data_entrega", "localizacao_atual", "produto", "cliente"]
13 }
14 }

12 "nome": "Leonor Yara",
13 "apelido": "Frances",
14 "morada_envio": {
  morada x cliente x produto x encomenda x + ↕
1 {
2   "$id": "https://datagen.di.uminho.pt/schemas/cliente",
3   "type": "object",
4   "properties": {
5     "nome": { "type": "string", "_datagen": "firstName" },
6     "apelido": { "type": "string", "_datagen": "surname" },
7     "contacto": { "type": "integer", "_datagen": "integerOfSize(9)" },
8     "morada_envio": { "$ref": "/schemas/morada" },
9     "morada_faturacao": { "$ref": "/schemas/morada" }
10  },
11  "required": ["nome", "apelido", "morada_envio", "morada_faturacao"]
12 }
```

```
morada x cliente x produto x encomenda x + ↕
1 {
2   "$id": "https://datagen.di.uminho.pt/schemas/produto",
3   "type": "object",
4   "properties": {
5     "codigo": { "type": "integer", "_datagen": "integerOfSize(10)" },
6     "nome": { "type": "string" },
7     "preco": { "type": "number" },
8     "stock": { "type": "integer" }
9   },
10  "required": ["codigo", "nome", "preco", "stock"]
11 }

12 <cliente>
13   <nome>Teodora Elisa</nome>
14   <apelido>Monte</apelido>

morada x cliente x produto x encomenda x + ↕
1 {
2   "$id": "https://datagen.di.uminho.pt/schemas/morada",
3   "type": "object",
4   "properties": {
5     "rua": { "type": "string" },
6     "freguesia": { "type": "string", "_datagen": "pt_parish" },
7     "cidade": { "type": "string", "_datagen": "pt_city" }
8   },
9   "additionalProperties": false,
10  "required": ["rua", "freguesia", "cidade"]
11 }
```

## Exemplo 2

peessoa

peessoa

```
1 {
2   "name": "Sidónio Vaquinhos",
3   "children": [
4     {
5       "name": "Carminho Miguel Aveleiro",
6       "children": [
7         {
8           "name": "Yuming Florido",
9           "children": [
10            {
11              "name": "Tânia Maiara Barbas",
12              "children": [
13                {
14                  "name": "Amandla Isabella Pisoeiro"
15                },
16                {
17                  "name": "Sofya Cacheira"
18                },
19                {
20                  "name": "Rafael Agostinho Beloo"
21                }
22              ]
23            },
24            {
25              "name": "Rahela Albano",
26              "children": [
27                {
28                  "name": "Luisandro José Jua"
29                }
30              ]
31            }
32          ]
33        },
34        {
35          "name": "Diego Artur Murilhas",
36          "children": [
37            {
```

peessoa

peessoa

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <peessoa>
3   <name>Kija Letra</name>
4   <children>
5     <elem_1>
6       <name>Ilusinda Servo</name>
7       <children>
8         <elem_1>
9           <name>Hasan Certal</name>
10        </elem_1>
11        <elem_2>
12          <name>Marvin Macario</name>
13        </elem_2>
14      </children>
15    </elem_1>
16    <elem_2>
17      <name>Gurleen Trepado</name>
18      <children>
19        <elem_1>
20          <name>Milene Susana Carpalhoso</name>
21        </elem_1>
22      </children>
23    </elem_2>
24    <elem_3>
25      <name>Kaven Rogado</name>
26      <children>
27        <elem_1>
28          <name>Maria Constança Serralheiro</name>
29        </elem_1>
30        <elem_2>
31          <name>Miriam Briana Barrocal</name>
32        </elem_2>
33      </children>
34    </elem_3>
35  </children>
36 </peessoa>
```

# Exemplo 3

```
1 {
2   "nodos": [
3     {
4       "id": 0,
5       "nome": "Qui proident dolor qui minim fugiat. Lorem dolor fugiat nostrud deserunt. Ea eu sit minim re",
6       "valor": -680
7     },
8     {
9       "id": 1,
10      "nome": "Ut duis laboris sunt nul",
11      "valor": -76
12     },
13     {
14       "id": 2,
15       "nome": "Lorem ipsum do laboris excepteur Lorem adipisicing non officia. Tempor",
16       "valor": -760
17     },
18     {
19       "id": 3,
20       "nome": "Proident mollit esse minim elit. Nisi duis elit ad ullamco eu ",
21       "valor": -278
22     },
23     {
24       "id": 4,
25       "nome": "Ut consectetur mollit nostrud in amet mini",
26       "valor": 55
27     },
28     {
29       "id": 5,
30       "nome": "Et voluptate velit ",
31       "valor": 594
32     },
33     {
34       "id": 6,
35       "nome": "Duis aliquip commodo est aute a",
36       "valor": -642
37     }
38   ]
39 }
```

```
12472   <origem>30</origem>
12473   <destino>79</destino>
12474   <peso>140.25</peso>
12475   </elem_1995>
12476   <elem_1996>
12477     <id>1995</id>
12478     <origem>64</origem>
12479     <destino>37</destino>
12480     <peso>98.835</peso>
12481   </elem_1996>
12482   <elem_1997>
12483     <id>1996</id>
12484     <origem>42</origem>
12485     <destino>14</destino>
12486     <peso>23.751</peso>
12487   </elem_1997>
12488   <elem_1998>
12489     <id>1997</id>
12490     <origem>8</origem>
12491     <destino>65</destino>
12492     <peso>209.256</peso>
12493   </elem_1998>
12494   <elem_1999>
12495     <id>1998</id>
12496     <origem>11</origem>
12497     <destino>4</destino>
12498     <peso>9.687</peso>
12499   </elem_1999>
12500   <elem_2000>
12501     <id>1999</id>
12502     <origem>14</origem>
12503     <destino>29</destino>
12504     <peso>193.672</peso>
12505   </elem_2000>
12506   </ligacoes>
12507 </grafo>
12508 }
```



Universidade do Minho  
Escola de Engenharia  
Departamento de Informática

# Geração de Dados Sintéticos a partir de Schemas JSON

---

Hugo André Coelho Cardoso  
José Carlos Leite Ramalho