DataGen

JSON/XML Dataset Generator

FILIPA SANTOS | HUGO CARDOSO | JOÃO COSTA | VÁLTER CARVALHO | JOSÉ RAMALHO (SUPERVISOR)

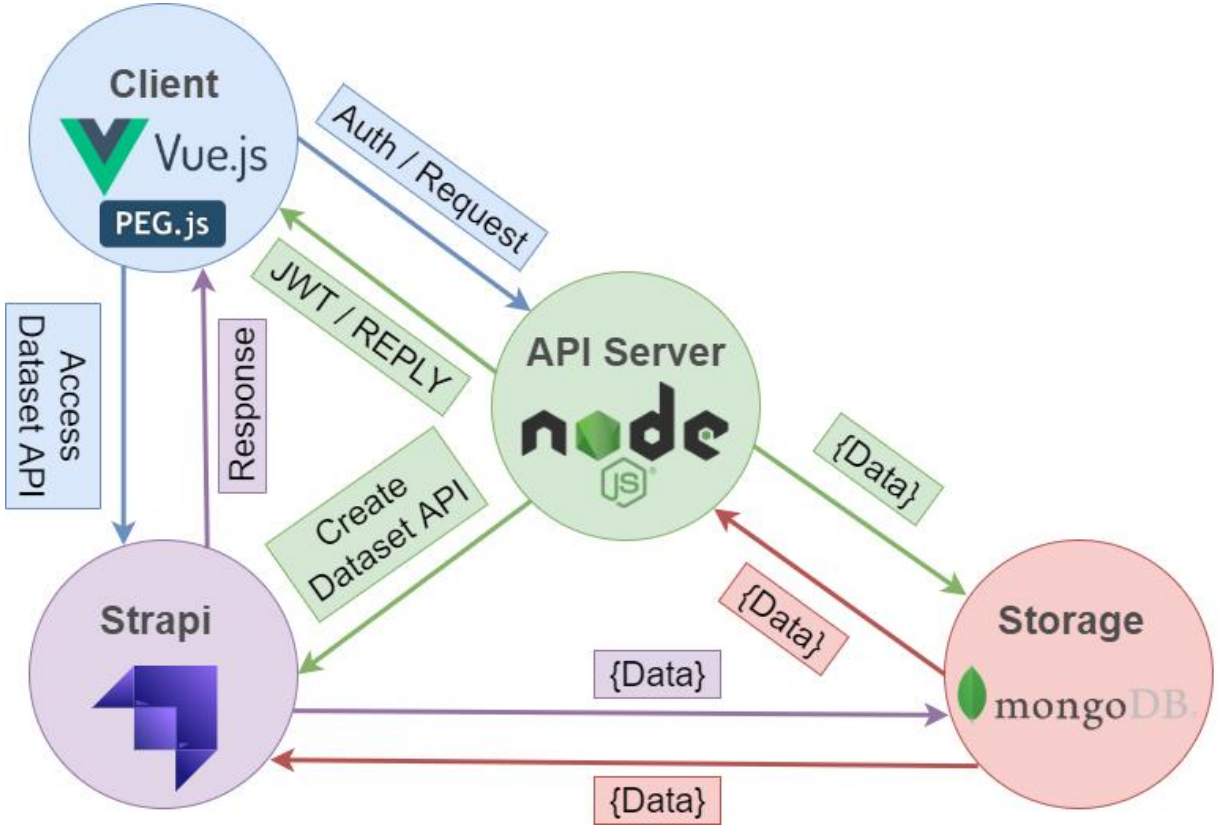
Concept

- Generation of datasets (JSON/XML)
- Generation of REST APIs on the datasets + CRUD
- Download the APIs and datasets
- Application Integration

Motivation

- Synthetic Data (User Privacy)
- Existing tools not complex enough
- Limited formats
- No fuzzy generation
- Multilingual support
- Etc…

Architecture



Strapi vs JSON-Server

- ✓ Generates a REST API
- ✓ Arbitrary Number of APIs
- ✓ Scalable
- × Does not digest datasets
- × Complex “Components”

- ✓ Generates a REST API
- ✓ Simple configuration
- ✓ Digests datasets
- × Only allows 1 API
- × Not scalable

Grammar

PEG.js

```
<!LANGUAGE en>
{
  profile: [
    'repeat(3)': {
        name: '{{fullName()}}',
        age: '{{integer(15,60)}}',
        or() {
            BI: '{{integerOfSize(8)}}-{{integer(0,9)}}',
            CC: '{{integerOfSize(8)}}-{{integer(0,9)}}-{{letter("uppercase")}}{{letter("uppercase")}}{{integer(0,9)}}'
        },
        having(70) {
            bio: '{{lorem(1,"sentences")}}'
        },
        nr_books: '{{integer(1,10)}}',
        books: [ 'repeat(this.nr_books)': {
            title: '{{lorem(1,"words")}}',
            support: '{{random("Físico","Digital")}}',
            rating: '{{integer(1,5)}}'
        } ],
        favorite_book(gen) {
          var titles = this.books.map(x => x.title)
          return gen.random(...titles)
        }
    }
  ]
}
```

```
{
  "profile": [
    {
      "name": "Lester Saianda",
      "age": 34,
      "CC": "23268383-7-FH4",
      "bio": "Non id eiusmod consectetur duis et consectetur.",
      "nr_books": 2,
      "books": [
        {
          "title": "ad",
          "support": "Físico",
          "rating": 4
        },
        {
          "title": "cillum",
          "support": "Digital",
          "rating": 1
        }
      ],
      "favorite_book": "ad"
    },
    {
      "name": "Daldip Rot",
      "age": 50,
      "BI": "96657177-2",
      "nr_books": 8,
      "books": [
        {
          "title": "velit",
          "support": "Digital",
```

JSON-like

```
<!LANGUAGE en>
{
  profile: [
    'repeat(3)': {
        name: '{{fullName()}}',
        age: '{{integer(15,60)}}',
        or() {
            BI: '{{integerOfSize(8)}}-{{integer(0,9)}}',
            CC: '{{integerOfSize(8)}}-{{integer(0,9)}}-{{letter("uppercase")}}{{letter("uppercase")}}{{integer(0,9)}}'
        },
        having(70) {
            bio: '{{lorem(1,"sentences")}}'
        },
        nr_books: '{{integer(1,10)}}',
        books: [ 'repeat(this.nr_books)': {
            title: '{{lorem(1,"words")}}',
            support: '{{random("Físico","Digital")}}',
            rating: '{{integer(1,5)}}'
        } ],
        favorite_book(gen) {
          var titles = this.books.map(x => x.title)
          return gen.random(...titles)
        }
    }
  ]
}
```

Interpolation Functions

```
<!LANGUAGE en>
{
  examples: [
    'repeat(2)': {
        objectId: '{{objectId()}}',
        guid: '{{guid()}}',
        index: '{{index()}}',
        letter: '{{letter()}}',
        position: '{{position()}}',
        date: '{{date("05-01-2001", "10-12-2020", "DD/MM/AAAA")}}',
        random: '{{random("Blue", false, 31, [1,2,3])}}'
    }
  ]
}
```

```
{
  "examples": [
    {
      "objectId": "60dcae00780814f12eca083c",
      "guid": "571390f0-771f-4a6e-9ff6-a3a02af4d758",
      "index": 0,
      "letter": "l",
      "position": "(-56.85308, -84.15255)",
      "date": "06/12/2008",
      "random": "Blue"
    },
    {
      "objectId": "60dcae0035668e02d5290843",
      "guid": "ffe962c6-2cbc-44f6-b496-94191a4bc148",
      "index": 1,
      "letter": "y",
      "position": "(46.54091, 16.24033)",
      "date": "22/10/2015",
      "random": 31
    }
  ]
}
```

Interpolation Functions

```
<!LANGUAGE en>
{
    examples: [
      'repeat(2)': {
          pt_district: '{{pt_district()}}',
          name: '{{fullName()}}',
          actor: '{{actor()}}',
          animal: '{{animal()}}',
          brand: '{{brand()}}',
          day: '{{weekday()}}',
          job_title: '{{job()}}',
          musician: '{{musician()}}'
      }
    ]
}
```

```
{
  "examples": [
    {
      "pt_district": "Braga",
      "name": "Tyler-Autumn Rollon",
      "actor": "Tom Cruise",
      "brand": "Netflix",
      "day": "Wednesday",
      "job_title": "Proof coins inspector",
      "musician": "R.E.L."
    },
    {
      "pt_district": "Viana do Castelo",
      "name": "Tyianna Escayola",
      "actor": "Toni Collette",
      "brand": "YouTube",
      "day": "Saturday",
      "job_title": "Fitness services manager",
      "musician": "Kid Rock"
    }
  ]
}
```

Javascript Functions

```
<!LANGUAGE en>
{
    examples: [
      'repeat(2)': {
          name: "Oliver",
          email(gen) {
            var i = gen.integer(1,30);
            return `${this.name}.${gen.surname()}${i}@gmail.com`.toLowerCase();
          },
          probability: gen => { return Math.random() * 100; }
      }
    ]
}
```

```
{
  "examples": [
    {
      "name": "Oliver",
      "email": "oliver.barlehner20@gmail.com",
      "probability": 29.6027670967689
    },
    {
      "name": "Oliver",
      "email": "oliver.volking21@gmail.com",
      "probability": 55.616499949869635
    }
  ]
}
```

Fuzzy generation

Probabilities

```
<!LANGUAGE en>
{
    examples: [ 'repeat(3)': {
        missing(50) {
            prop1: 1,
            prop2: 2
        },
        having(80) { prop3: 3 }
    } ]
}
```

```
{
  "examples": [
    {},
    {
      "prop1": 1,
      "prop2": 2,
      "prop3": 3
    },
    {
      "prop3": 3
    }
  ]
}
```

Logical conditions

```
<!LANGUAGE en>
{
    examples: [ 'repeat(3)': {
        type: '{{random("A","B","C")}}',
        if (this.type == "A") { A: "type is A" }
        else if (this.type == "B") { B: "type is B" }
        else { C: "type is C" }
    } ]
}
```

```
{
  "examples": [
    {
      "type": "A",
      "A": "type is A"
    },
    {
      "type": "A",
      "A": "type is A"
    },
    {
      "type": "C",
      "C": "type is C"
    }
  ]
}
```

Fuzzy generation

Or (mutual exclusivity)

```
<!LANGUAGE en>
{
    examples: [ 'repeat(3)': {
        or() {
            prop1: 1,
            prop2: 2,
            prop3: 3
        }
    } ]
}
```

```
{
  "examples": [
    {
      "prop3": 3
    },
    {
      "prop1": 1
    },
    {
      "prop1": 1
    }
  ]
}
```

At least

```
<!LANGUAGE en>
{
    examples: [ 'repeat(3)': {
        at_least(2) {
            prop1: 1,
            prop2: 2,
            prop3: 3
        }
    } ]
}
```

```
{
  "examples": [
    {
      "prop3": 3,
      "prop1": 1
    },
    {
      "prop2": 2,
      "prop1": 1
    },
    {
      "prop2": 2,
      "prop1": 1,
      "prop3": 3
    }
  ]
}
```

Functional capabilities

```
<!LANGUAGE en>
{
    example: [ 'repeat(1)': {
        map: range(5).map(value => { return value+1 }),

        filter: [0,1,2].filter(function(value, index) {return [0,1,2][index]>0}),

        reduce: range(5).reduce((accum, value, index, array) => {
                return accum + array[index]
        }),

        combined: range(5).map((value) => { return value+3 })
                          .filter(x => { return x >= 5 })
                          .map(x => { return x*2 })
                          .reduce((a,c) => {return a+c})
    } ]
}
```

```
{
  "examples": [
    {
      "map": [
        1,
        2,
        3,
        4,
        5
      ],
      "filter": [
        1,
        2
      ],
      "reduce": 10,
      "combined": 36
    }
  ]
}
```

Real-life study case

```
<!LANGUAGE pt>
{
  autoEliminação: {
        fonteLegitimação: {
            tipo: '{{random("PGD/LC", "TS/LC", "PGD", "RADA", "RADA/CLAV")}}',
            diploma(gen) {
                var portaria = `Portaria ${gen.integer(300,500)}/${gen.integer(2000,2021)}`
                var despacho = `Despacho DGLAB ${gen.integer(100,500)}/${gen.integer(2000,2021)}`
                return gen.random("LC", portaria, despacho)
            }
        },
        fundos(gen) {
            if (["PGD/LC","TS/LC","PGD"].includes(this.fonteLegitimação.tipo))
                return [gen.pt_entity()]
            else {
                var arr = []
                for (var i = 0; i < gen.integer(1,5); i++) arr.push(gen.pt_entity())
                return arr
            }
        },
```

Real-life study case

```
21          classes: [ 'repeat(2,5)': {
22              if (["PGD/LC","TS/LC"].includes(this.fonteLegitimação.tipo)) {
23                código: gen => {
24                    var nivel1 = gen.random(...gen.range(100,950,50))
25                    var nivel2 = gen.random(10,20,30,40,50)
26                    var nivel3 = gen.integer(1,999,3)
27                    var nivel4 = gen.random("01","02")
28
29                    var classe = nivel1 + '.' + nivel2 + '.' + nivel3
30                    if (Math.random() > 0.5) classe += '.' + nivel4
31                    return classe
32                }
33              }
34              else {
35                  at_least(1) {
36                      código(gen) {
37                          var nivel1 = gen.random(...gen.range(100,950,50))
38                          var nivel2 = gen.random(10,20,30,40,50)
39                          var nivel3 = gen.integer(001,999)
40                          var nivel4 = gen.random("01","02")
41
42                          var classe = nivel1 + '.' + nivel2 + '.' + nivel3
43                          if (Math.random() > 0.5) classe += '.' + nivel4
44                          return classe
45                      },
46                      referência: '{{random(1,2,3,55,56)}}'
47                  }
48              },
```

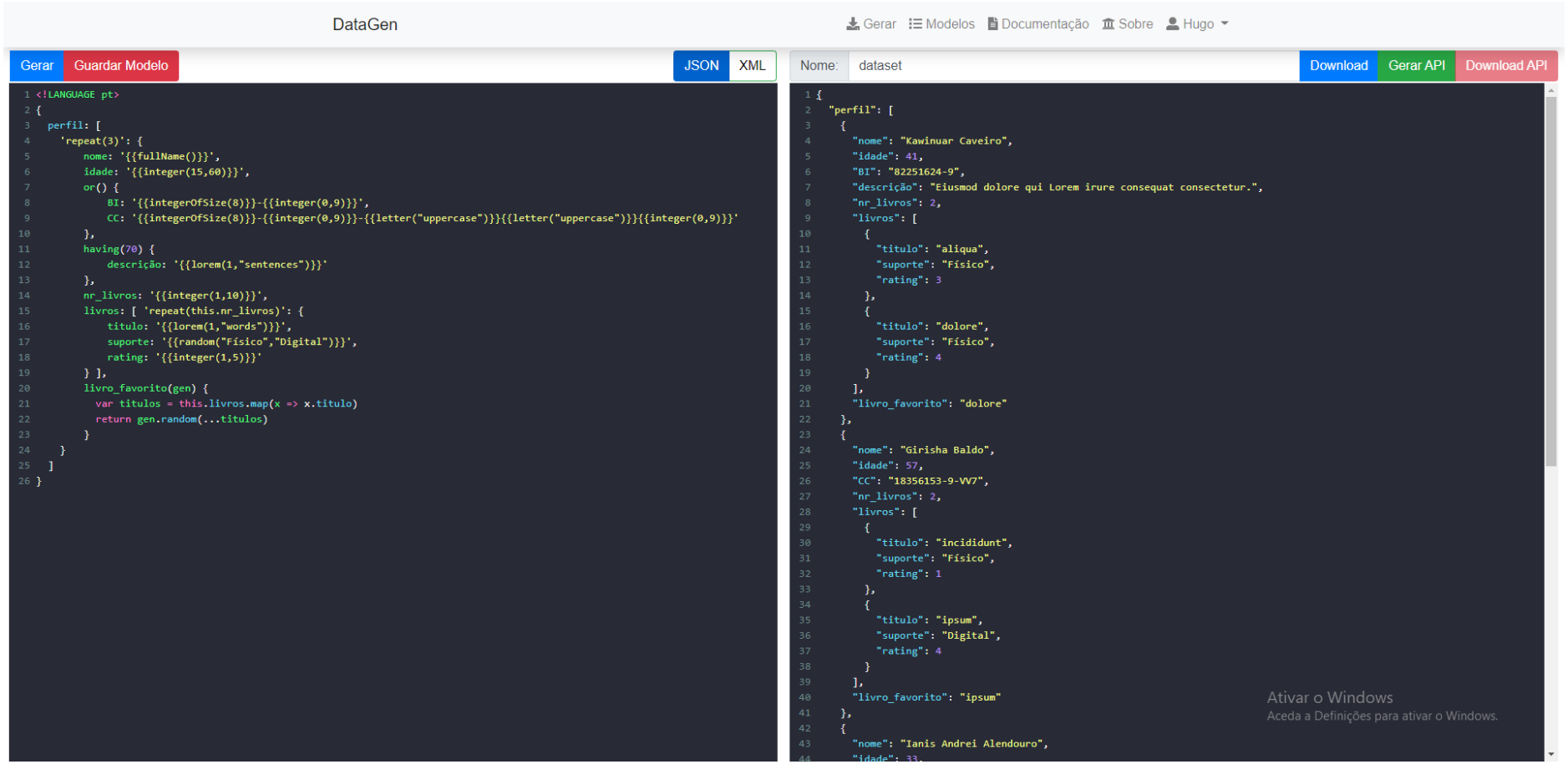
Real-life study case

```
49          if (["PGD/LC","TS/LC"].includes(this.fonteLegitimação.tipo)) {
50              naturezaIntervenção: '{{random("DONO", "PARTICIPANTE", "DONO/PARTICIPANTE")}}',
51              if (["PARTICIPANTE","DONO/PARTICIPANTE"].includes(this.naturezaIntervenção)) {
52                  donos: [ 'repeat(1,5)': '{{pt_entity_abbr()}}' ]
53              }
54          },
55          anoInício: '{{integer(1921,2021)}}',
56          anoFim(gen) {
57              var ano = gen.integer(1921,2021)
58              while (ano < this.anoInício) ano = gen.integer(1921,2021)
59              return ano
60          },
61          dimensãoSuporte: {
62              at_least(1) {
63                  papel: '{{integer(1,2000)}}',
64                  digital: '{{integer(1,2000)}}',
65                  outro: {
66                      valor: '{{integer(1,2000)}}',
67                      unidade: '{{lorem(1,"words")}}'
68                  }
69              }
70          },
```

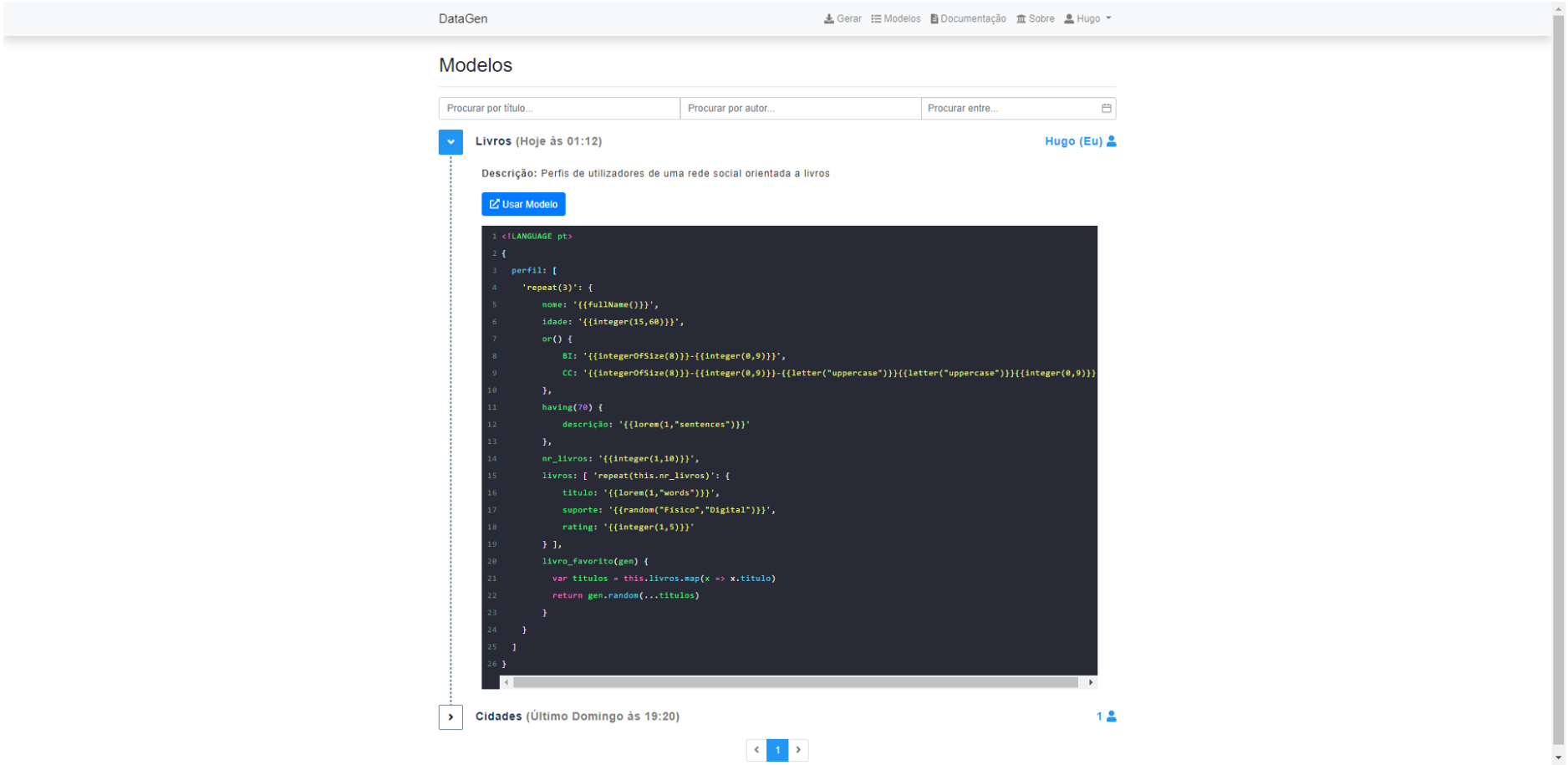
Real-life study case

```
71              númeroAgregações: '{{integer(1,50)}}',
72              agregações: [ 'repeat(this.númeroAgregações)': {
73                  código: '{{pt_entity_abbr()}} - {{integer(1,200)}}',
74                  título: '{{lorem(3,"words")}}',
75                  ano: '{{integer(1921,2021)}}',
76                  if (["PGD/LC","TS/LC"].includes(this.fonteLegitimação.tipo)) {
77                      naturezaIntervenção: '{{random("PARTICIPANTE","DONO")}}'
78                  }
79              }]
80          }]
81      }
82  }
```

Interface - Home

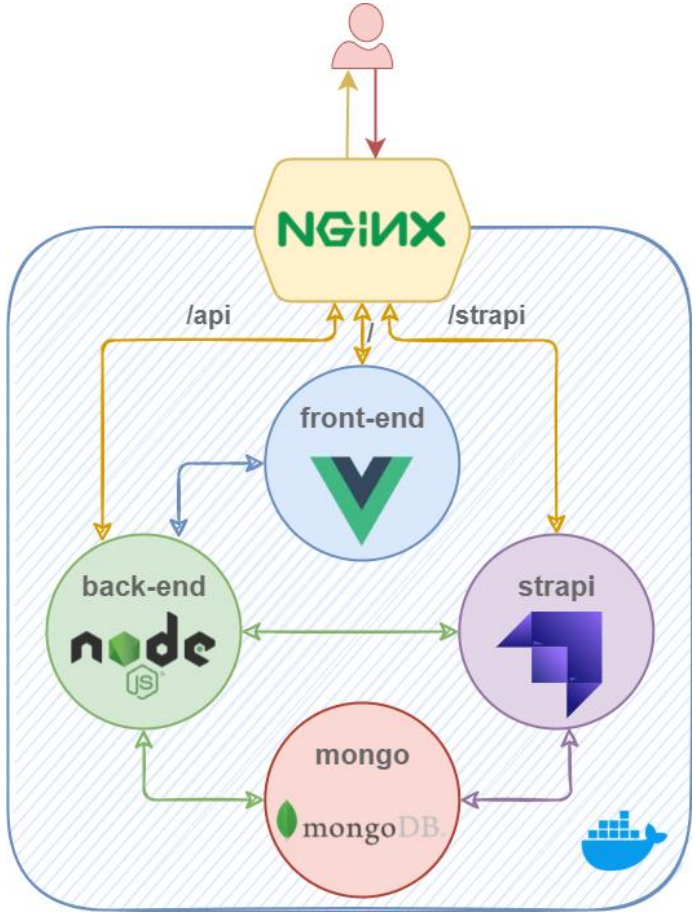


Interface – A User's Models



Interface - Documentation

Deployment


NGINX
/api
/
/strapi
front-end
back-end
node
strapi
mongo
mongoDB

THE END

Questions?

DataGen

JSON/XML Dataset Generator

fliper6/DataGen
Abjiri/DataGen

Jcc20/DataGen
wurzy/DataGen