



CNL – COVID-19 Notification Language

PPGTI1007 LINGUAGENS DE
DOMÍNIO ESPECÍFICO T01

Cezar Miranda
Denis Orozco

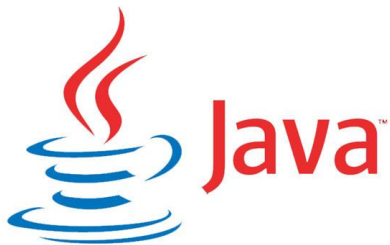
Motivação

A CNL é uma DSL (*Domain Specific Language*) com o intuito de favorecer a interoperabilidade na notificação de casos de COVID-19.

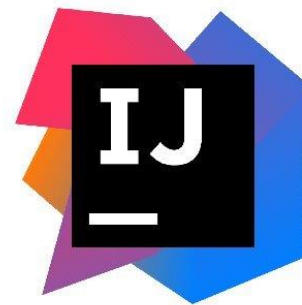
- Cenários de utilização vislumbrados:
 - Padronização para interoperabilidade entre diferentes sistemas, plataformas de saúde e mesmo países.
 - Fornecer uma estrutura de dados mais alto nível que possa ser utilizada na comunicação entre profissionais de saúde e profissionais de Tecnologia da Informação

Visão Geral

Stack e Ferramentas



A CNL foi desenvolvida utilizando ANTLv4, Java 8, Python 3.9 e utilizando as IDEs IntelliJ Idea e PyCharm da JetBrains.



Tipos de Dados

Paciente

Estrutura de dados segue o padrão abaixo.

Obrigatório apresentar nome, CPF e data de nascimento.

```
{
  nome : texto*, //aceita acentuação, maiúsculas, minúsculas e espaços, sem números
  cpf : texto*, //aceita máscara de CPF ou 11 números corridos
  nascimento : data, //mascara "yyyy-mm-dd"
  diagnostico : TIPODIAGNOSTICO**,
  comorbidades : Lista de TIPOCOMORBIDADE**,
  sinais : Lista de Sinal***,
  sintomas : Lista de Sintoma***,
  exames : Lista de Exame***,
  vacinas : Lista de Vacina***
}
```

*possui restrições conforme comentários

**constantes literais conforme seção "Campos de valores fixos e literais" abaixo

***Lista do tipo de objeto conforme próximas seções

Sintoma

Estrutura de dados segue o padrão abaixo. Obrigatório apresentar tipo e data.

```
{
  cpf: texto*, //aceita máscara de CPF ou 11 números corridos
  tipo: TIPOSINTOMA**,
  data : data-e-hora //mascara "yyyy-mm-dd hh:mm:ss"
}
```

*possui restrições conforme comentários. Preenchimento opcional!

**constantes literais conforme seção "Campos de valores fixos e literais" abaixo

Os tipos de dados reconhecidos pelo programa podem ser dos tipos Paciente, Sintoma, Sinal, Comorbidade, Exame, Vacina.

Tipos de Dados

Sinal Vital

Estrutura de dados segue o padrão abaixo. Obrigatório apresentar tipo, data e leitura.

```
{
  cpf: texto*, //aceita máscara de CPF ou 11 números corridos
  tipo: TIPOSINAL**,
  data : data-e-hora //mascara "yyyy-mm-dd hh:mm:ss"
  leitura : Temperatura ou Percentual //34.0 °C a 42.9 °C, 0% a 99%
}
```

*possui restrições conforme comentários. Preenchimento opcional!

**constantes literais conforme seção "Campos de valores fixos e literais" abaixo

Exame

Estrutura de dados segue o padrão abaixo. Obrigatório apresentar tipo, data e resultado. Se o tipo é SOROLOGIA, obrigatório apresentar IgG e IgM.

```
{
  cpf: texto*, //aceita máscara de CPF ou 11 números corridos
  tipo: TIPOEXAME**,
  data : data-e-hora //mascara "yyyy-mm-dd hh:mm:ss"
  resultado : TIPORESULTADO**,
  igg : Double, //apenas para o tipo SOROLOGIA
  igm : Double //apenas para o tipo SOROLOGIA
}
```

*possui restrições conforme comentários. Preenchimento opcional!

**constantes literais conforme seção "Campos de valores fixos e literais" abaixo

Os tipos de dados reconhecidos pelo programa podem ser dos tipos Paciente, Sintoma, Sinal, Comorbidade, Exame, Vacina.

Tipos de Dados

Vacina

Estrutura de dados segue o padrão abaixo. Obrigatório apresentar tipo, data, dose, serial e lote.

```
{
  cpf: texto*, //aceita máscara de CPF ou 11 números corridos
  tipo: TIPOVACINA**,
  data : data-e-hora //mascara "yyyy-mm-dd hh:mm:ss"
  dose : 1ª ou 2ª
  serial : numero serial (números, hifens e letras)
  lote : lote (números, hifens e letras)
}
```

*possui restrições conforme comentários. Preenchimento opcional!

**constantes literais conforme seção "Campos de valores fixos e literais" abaixo

Os tipos de dados reconhecidos pelo programa podem ser dos tipos Paciente, Sintoma, Sinal, Comorbidade, Exame, Vacina.

Tipos de Dados*

```
sinais:
[
  {
    tipo: OXIMETRIA,
    data : "2020-12-05 14:06:55",
    leitura: 97%
  },
  {
    cpf: 010.452.754-41,
    tipo: TEMPERATURA,
    data : "2020-12-03 14:02:17",
    leitura: 41.5C
  },
  {
    cpf: 010.452.754-41,
    tipo: TEMPERATURA,
    data : "2021-02-03 14:02:17",
    leitura: 35.8C
  }
]
```

Listas de objetos são suportadas.

Tipos de Dados*

Comorbidades são consideradas objetos também.

Listas de comorbidades são suportadas.



Literais / Tipos

Diversos campos possuem valores possíveis pré-determinados.

Como uma regra geral, **onde houver espaços** será necessário utilizar aspas duplas. Isto também vale para datas, nomes, etc.

Tipos de Exames:

RT-PCR
"TESTE RÁPIDO"
SOROLOGIA
"ANTICORPOS TOTAIS"

Comorbidades :

IDOSO
OBESIDADE
DIABETES
HIPERTENSÃO
CARDIOPATIA
NEOPLASIA
NEFROPATIA
IMUNODEFICIÊNCIA
TRANSPLANTADO
HEPATOPATIA
HEMATOPATIA
NEUROPATIA
PNEUMOPATIA

Tipos de Sinal Vital:

OXIMETRIA
TEMPERATURA

Tipos de Vacina (Obs: para fins de simplificação, nomes de fabricantes e de vacinas serão considerados sinônimos)

OXFORD
ASTRA-ZENACA
MODERNA
PFIZER
BIONTECH
GAMALEYA
"SPUTNIK V"
SINOVAC
CORONAVAC

Resultados:

POSITIVO
REAGENTE
NEGATIVO
"NÃO REAGENTE"

Tipos de Sintomas:

"CORRIMENTO NASAL"
FEBRE
DIARRÉIA
"TOSSE SECA"
CANSAÇO
"DOR DE GARGANTA"
DESCONFORTO
"DOR DE CABEÇA"
ANOSMIA
"PERDA DO OLFATO"
AGUSMIA
"PERDA DO PALADAR"
MIALGIA
"DOR NO CORPO"
CONJUNTIVITE
"ERUPÇÕES CUTÂNEAS"
"FALTA DE AR"
"DOR NO PEITO"

Operações

Atribuição (=)

Atribuição de variáveis é suportada.

variavel = objeto

```
examList = [{  
    tipo: RT-PCR,  
    data : "2020-12-03 14:24:32",  
    resultado: NEGATIVO  
},  
{  
    cpf: 010.452.754-41,  
    tipo: SOROLOGIA,  
    data : "2020-12-28 09:14:53",  
    resultado: REAGENTE,  
    igg: 8.0,  
    igm: 0.4  
}]
```

Atribuição (=)


Atribuição de variáveis é suportada.

A linguagem reconhece o tipo das variáveis nas operações posteriores.

variavel = objeto

```
examList = [{
  tipo: RT-PCR,
  data : "2020-12-03 14:24:32",
  resultado: NEGATIVO
},
{
  cpf: 010.452.754-41,
  tipo: SOROLOGIA,
  data : "2020-12-28 09:14:53",
  resultado: REAGENTE,
  igg: 8.0,
  igm: 0.4
}]
```

Tipo é inferido e
armazenado

variavel = objeto

Atribuição (=)

Atribuição de variáveis é
suportada.

A linguagem reconhece o tipo
das variáveis nas operações
posteriores.

```
examList = [{  
    tipo: RT-PCR,  
    data : "2020-12-03 14:24:32",  
    resultado: NEGATIVO  
},  
{  
    cpf: 010.452.754-41,  
    tipo: SOROLOGIA,  
    data : "2020-12-28 09:14:53",  
    resultado: REAGENTE,  
    igg: 8.0,  
    igm: 0.4  
}]
```

Concatenação (+)

Dados de outros objetos podem ser agregados ao Paciente.

Permite agregar exames, vacinas, sinais vitais, sintomas, comorbidades. Individualmente ou em listas.

paciente + objeto

```
paciente + [NEOPLASIA, NEFROPATIA]
paciente + PNEUMOPATIA
paciente + NEUROPATIA
paciente + {
    tipo: OXFORD,
    data : "2021-02-10 07:39:21",
    dose: 1,
    serial: 21329AS3292230A-AFF31,
    lote: 34BC
}
paciente + examList
```

Concatenação (+)

Dados de outros objetos podem
ser agregados ao Paciente.

(tipos de dados são conferidos)

paciente + objeto

```
paciente + [NEOPLASIA, NEFROPATIA]
paciente + PNEUMOPATIA
paciente + NEUROPATIA
paciente + {
    tipo: OXFORD,
    data : "2021-02-10 07:39:21",
    dose: 1,
    serial: 21329AS3292230A-AFF31,
    lote: 34BC
}
paciente + examList
```


Métodos

Considerações gerais sobre métodos com saída de dados

- Dados temporais são sempre exibidos na ordem cronológica, independente da ordem em que foram armazenados.
- CPFs podem ser fornecidos com ou sem a máscara, e serão exibidos formatados.
- Datas e horas serão formatadas no padrão “dd/mm/yyyy HH:MM:SS”

print

Imprime um “extrato” com todos os dados do paciente.

O parâmetro [data] é opcional e limita “até quando” são consideradas as informações temporais (exames, sintomas, sinais vitais, vacinas).

```
print paciente [data]
```

```
-----  
Paciente: Joao da Silva Biro  
CPF: 010.452.754-41  
Data de Nascimento: 12/07/1979
```

```
-----  
Comorbidades
```

```
-----  
DIABETES  
CARDIOPATIA
```

```
-----  
Sinais Vitais
```

```
-----  
Tipo: TEMPERATURA  
Data: 03/12/2020 14:02:17  
leitura: 41.5C
```

```
-----  
Tipo: OXIMETRIA  
Data: 05/12/2020 14:06:55  
leitura: 97%
```

```
-----  
Sintomas
```

```
-----  
Tipo: FALTA DE AR  
Data: 03/12/2020 14:02:17
```

```
-----  
Exames
```

```
-----  
Tipo: TESTE RAPIDO
```

comorbidade

Imprime apenas as
comorbidades do paciente.

sinal

Imprime apenas os sinais vitais
do paciente. O parâmetro [data]
limita “até quando” são
consideradas as informações.

comorbidade paciente

```
-----  
Comorbidades  
-----  
IDOSO  
DIABETES  
CARDIOPATIA  
NEOPLASIA  
NEFROPATIA  
PNEUMOPATIA  
NEUROPATIA
```

sinal paciente [data]

```
-----  
Sinais Vitais  
-----  
CPF: 010.452.754-41  
Tipo: TEMPERATURA  
Data: 03/12/2020 14:02:17  
leitura: 41.5C  
-----  
Tipo: OXIMETRIA  
Data: 05/12/2020 14:06:55  
leitura: 97%  
-----  
CPF: 010.452.754-41  
Tipo: TEMPERATURA  
Data: 03/02/2021 14:02:17  
leitura: 35.8C
```

sintoma

Imprime apenas os sintomas. O parâmetro [data] limita “até quando” são consideradas as informações.

vacina

Imprime apenas as vacinas. O parâmetro [data] limita “até quando” são consideradas as informações.

sintoma paciente [data]

Sintomas

CPF: 010.452.754-41

Tipo: FALTA DE AR

Data: 03/12/2020 14:02:17

Tipo: FEBRE

Data: 08/12/2020 10:15:41

vacina paciente [data]

vacinas

Tipo: ASTRA-ZENACA

Data: 10/02/2021 07:39:21

dose: 2

serial: 238409D23098DA2-23A-ZZ0A

lote: 234D

Tipo: OXFORD

Data: 10/02/2021 07:39:21

dose: 1

serial: 21329AS3292230A-AFF31

lote: 34BC

exame

Imprime apenas os exames. O parâmetro [data] limita “até quando” são consideradas as informações.

O tipo “SOROLOGIA” obriga a informação de IgG e IgM.

(Faz parte das validações de tipo de objeto).

exame paciente [data]

```
-----  
Exames  
-----  
Tipo: TESTE RAPIDO  
Data: 01/12/2020 10:12:04  
resultado: NAO REAGENTE  
-----  
Tipo: RT-PCR  
Data: 03/12/2020 14:24:32  
resultado: NEGATIVO  
-----  
Tipo: ANTICORPOS TOTAIS  
Data: 12/12/2020 07:43:18  
resultado: POSITIVO  
-----  
CPF: 010.452.754-41  
Tipo: SOROLOGIA  
Data: 28/12/2020 09:14:53  
resultado: REAGENTE  
IgG: 8.0  
IgM: 0.4
```

diagnostico

Calcula o diagnóstico do paciente com base nos dados de sinais vitais, exames e vacinas.

O parâmetro [data] limita “até quando” são consideradas as informações, possibilitando calcular o diagnóstico do paciente em momentos passados.

diagnostico paciente [data]

Paciente: Joao da Silva Biro
CPF: 010.452.754-41
Diagnóstico: INFECCIOSO EM 12/12/2020 07:43:18

Dados do Exame

Tipo: ANTICORPOS TOTAIS
resultado: POSITIVO

OXIMETRIA 97%
TEMPERATURA 41.5C (HIPERtermIA)
Estado Geral: GRAVÍSSIMO

Validação e Interpretação

Validação (CNLValidatorListener)

A CNL:

- Valida e infere os tipos de objetos válidos
- Valida os pares obrigatórios e critica se há informações repetidas nos objetos
- Valida os tipos de dados das variáveis
- Valida os operandos nas operações e métodos.

Interpretação (CNLTranslatorListener)

A CNL:

- Gera código Python para execução
- Possui uma biblioteca de métodos pré-definidos que é importada em tempo de execução
- Tem uma economia de tamanho de código de cerca 400% em comparação com o código Python gerado.

Demonstração

Problemas e Limitações conhecidas

- *Problemas no reconhecimento de acentuação e caracteres especiais (a, °).
- CPF não é validado entre paciente e outros objetos (sintomas, exames, sinais vitais, vacinas)

*Problemas não ocorrem na análise com o plugin ANTLR v4 do IntelliJ, mas ocorrem em execução

Dúvidas?

Obrigado!

