

Artigo

[Danusa Calixto](#) · Out. 27, 2022 6min de leitura

O Servidor FHIR alimenta o Project Vulcan na HL7 FHIR Connectathon



Coloquei o Servidor FHIR da InterSystems para trabalhar na [HL7 FHIR Connectathon em Baltimore](#) no último final de semana, postando pacotes, usando recursos Rest e interrogando o [Guia de Implementação Vulcan](#). Nós montamos os projetos Real World Data (RWD) e Schedule of Activity (SoA) para o pessoal do Vulcan IG, indo em frente e conectando pesquisa clínica e dados de saúde. Empregamos uma abordagem bastante decente para atender aos requisitos,

Com os colegas da InterSystems (Huy, Russell, Regilo), estávamos responsáveis por dar suporte ao Servidor FHIR e à apropriação de dados para o evento. Se você já foi encarregado de digitar Pacotes FHIR para um caso de uso específico, acho que pode apreciar como a dificuldade pode ser enganosa.

Provemos alguns Servidores FHIR da InterSystems do tamanho de um aparelho de som, 16 núcleos/128 GB, um para cada projeto, e começamos a trabalhar para cada equipe. As equipes vieram preparadas com seus notebooks Python contra nossos endpoints, com a expectativa de escrever uma História FHIR para a SCA, com um script de teste processual validando as chamadas retornadas com os resultados esperados.

A meta da coorte para o RWD era bastante simples nas consultas de exemplo da [Síndrome coronária aguda](#) para mim:

Não é?

```
/Patient?birthdate=le2002-09-01&gender=male,female  
/Encounter?reason-code:below=I20,I21,I22,I23,I24,I25&date=ge2020-09-01&date=le2021-09-31&status=finished&dischargeDisposition:not=exp  
/MedicationAdministration?status=completed&effective-time=ge[Encounter-Start-Date]&  
code=http://www.nlm.nih.gov/research/umls/rxnorm|1116632,http://www.nlm.nih.gov/resea  
rch/umls/rxnorm|613391,http://www.nlm.nih.gov/research/umls/rxnorm|32968,http://www.n  
lm.nih.gov/research/umls/rxnorm|687667,http://www.nlm.nih.gov/research/umls/rxnorm|15  
3658
```

Claro...

Fracasso Falha do Pacote de Submódulo Synthea De maneira esperançosa, nossa primeira ave-maria foi para gerar 2 milhões de recursos e milhares de Pacotes Synthea (usando este adorável [repositório da InterSystems](#)) para o endpoint de transação e o recurso de upload do Gerenciamento de Dados, incluindo vários submódulos centrados no coração e computadores sofisticados...

```
docker run --rm -v $PWD/output:/output -v $PWD/modules:/modules --name synthea-docker  
intersystemsdemo/irisdemo-base-synthea:version-$VERSION --exporter.practitioner.fhir.e  
xport true --exporter.hospital.fhir.export true --exporter.fhir.use_us_core_ig true -  
p 500 -s 21 -d /modules
```

Não alcançamos isso, o que não é surpreendente, eu acho, mas eu meio que esperava pelo menos chegar perto. Não chegamos.

Vitórias O Processo de Geração de Dados Então, o [Geoff Low](#) nos deu uma pista sobre seu processo, nos separamos e prosperamos... é assim que o processo acontece.

1. Pegue um pacote de ponto de partida, em especial um que seja relevante para a Pesquisa e que tenha um mínimo de pacientes e encontros e use o pacote SDTM no [sourceforge](#):
2. Agora, implemente uma [etapa de patching](#), para colocar o pacote no lugar, certifique-se de que ele seja carregado com sucesso no Servidor FHIR, carregue ou poste o pacote.
3. Depois de um novo ponto de partida, [aumente os encontros](#) com os códigos de motivos previstos (I20,I21,I22,I23,I24,I25).

```
reasons = ["I21","I22","I23","I24","I25"]````
```

4. Adicione medicamentos aos encontros, incluindo todos os medicamentos esperados para o encontro para a Pesquisa da SCA.

```
medchoices = [  
    {  
        "code": "1116632",  
        "display": "ticagrelor"  
    },  
    {  
        "code": "613391",  
        "display": "prasurgrel"  
    },  
    {  
        "code": "32968",  
        "display": "clopidogrel"  
    }  
]  
medchoice = random.choice(medchoices)  
medconcept = CodeableConcept(  
    coding=[  
        Coding(  
            code=medchoice["code"],  
            display=medchoice["display"],  
            system="http://www.nlm.nih.gov/research/umls/rxnorm",  
        )  
    ]  
)
```

IMPORTANTE! Verifique se os medicamentos estão dentro das datas/horários do encontro.

5. Em seguida, confira se os Encontros realmente são internações e inclua todos os tipos aplicáveis.

```
statuses = ["in-progress", "finished"]
dischargeCodes =
  {
    "code": "home",
    "display": "Home"
  },
  {
    "code": "hosp",
    "display": "Hospice"
  },
  {
    "code": "exp",
    "display": "Expired"
  },
  {
    "code": "long",
    "display": "Long-term care"
  },
  {
    "code": "alt-home",
    "display": "Alternative home"
  },
]
```

Outra vitória Implementando o Parâmetro de Pesquisa do GI O Guia de Implementação Vulcan inclui um parâmetro de pesquisa que é necessário para retornar encontros por disposição de alta. Abordamos isso de duas maneiras:

a. Carregar todo o GI b. Carregar o Parâmetro de Pesquisa

Carregar só o Parâmetro de Pesquisa era o caminho mais curto e estávamos em uma situação complicada por tempo suficiente, então aqui está como foi esse processo.

Criada a pasta /tmp/mypackage, adicione o parâmetro de pesquisa do GI e seu próprio arquivo do pacote.

```
[irisdeploy@ip-192-168-0-37 tmp]$ tree /tmp/mypackage
/tmp/mypackage
??? package.json
??? parameter.json
```

O arquivo do pacote pode parecer assim:

```
{
  "name": "ron.sweeney.r4",
  "version": "0.0.1",
  "dependencies": {
    "hl7.fhir.r4.core": "4.0.1"
  }
}
```

```
}
```

Em seguida, carregue isso no IRIS, desta maneira:

```
TL4:IRIS:FHIRDB>do ##class(HS.FHIRMeta.Load.NpmLoader).importPackages($lb("/tmp/mypackage"))  
Saving ron.sweeney.r4@0.0.1  
Load Resources: ron.sweeney.r4@0.0.1
```

A próxima etapa é associar o pacote ao endpoint:

Em seguida, indexe novamente os encontros no repositório:

Agora, você já deve ter o parâmetro de pesquisa/pacote carregado no endpoint:

Com o parâmetro de pesquisa carregado, podemos agora usá-lo assim!

```
curl https://fhir.ggyxlz8lbozu.workload-prod-fhiraas.isccloud.io/Encounter?dischargeDisposition=hosp
```

Enfim, o processo funcionou para nós, e considerar a geração de pacotes pela visão do Python foi uma abordagem muito melhor do que o Synthea às vezes é para conjuntos de dados minuciosos.

O conjunto de dados resultante com que saímos no Domingo está [aqui](#).

Só queria compartilhar a experiência e agradecer às equipes dos projetos por tornarem o que pensei que seria um pesadelo em uma experiência divertida e educativa!

[#FHIR](#) [#integration-required](#) [#InterSystems](#) [IRIS](#)

URL de
origem: <https://pt.community.intersystems.com/post/o-servidor-fhir-alimenta-o-project-vulcan-na-hl7-fhir-connectathon>