

# Arquitetura baseada em serviços e interoperabilidade: criando um servidor HL7® FHIR com a API Cloud Healthcare

Guilherme Bauer Negrini<sup>1</sup> e Lucas Pacheco Silveira<sup>1</sup>

<sup>1</sup>Informática Biomédica – Universidade Federal de Ciências da Saúde de Porto Alegre  
(UFCSPA)

90050-170 – Porto Alegre – RS – Brazil

{guilhermebn, lucassi}@ufcspa.edu.br

**Abstract.** *The HL7® FHIR interoperability standard describes the structure, Resources and an API for exchanging information from electronic health records. Cloud computing services offer a great option to build healthcare applications in a scalable and easy to maintain manner. In this work we demonstrate the creation of an HL7® FHIR server using cloud computing resources from the Google Cloud platform.*

**Resumo.** *O padrão de interoperabilidade HL7® FHIR descreve a estrutura, os Recursos e uma API para troca de informações de registros eletrônicos de saúde. Serviços de computação em nuvem oferecem uma ótima opção para construir aplicações voltadas à saúde de forma escalável e de fácil manutenção. Neste trabalho demonstramos a criação de um servidor HL7® FHIR na prática utilizando recursos de computação em nuvem da plataforma Google Cloud.*

## 1. Introdução

Em aplicações voltadas para a área da saúde encontram-se diversas peculiaridades durante o desenvolvimento quando comparadas com aplicações em diferentes áreas. Uma dessas diferenças é como os dados devem ser armazenados e transferidos. Para trabalhar com estas singularidades, em específico com os formatos, elementos e comunicação destes dados, surgiram diversos padrões que visam facilitar e normalizar como estes dados são gerados e trocados entre aplicações. Entre eles, pode-se destacar o Health Level Seven Version 3 (HL7® V3), apresentado oficialmente em 2005, baseado em uma metodologia formal e em princípios orientados a objetos. Essas estratégias adotadas, caso seguidas rigorosamente, eram capazes de atingir o objetivo para que foram desenvolvidas. Porém, no momento de adaptar os sistemas de saúde a esse padrão, existiam algumas dificuldades, pois ele era muito rígido e complexo, além de ser baseado em regras de comunicações antigas para a época. Após dois anos do seu lançamento, o HL7® V3 já era visto por alguns como um padrão que havia falhado. Como alternativa, Graham Grieve escreveu um primeiro rascunho, intitulado “*Resources For Healthcare (RFH)*”, no seu blog, abordando sobre um novo padrão, que visava resolver esses problemas não contemplados no HL7® V3, posteriormente

este novo padrão fora denominado Health Level Seven Fast Healthcare Interoperability Resources (HL7® FHIR ou FHIR) [*RFH: Resources For Healthcare*, 2011].

Atualmente, o FHIR está em sua versão 4 (HL7® FHIR R4), publicada no segundo semestre de 2019, esta versão será utilizada durante este trabalho para construção dos dados e comunicações. O padrão especifica como preferência a utilização de verbos HTTP para compartilhamento de dados, que é uma das fundações das aplicações Web atualmente. Além de estabelecer os modelos de dados e os métodos de comunicação, ele sugere os tipos de dados que você pode utilizar. Ele organiza sua estrutura interna composta de Recursos, que devem ter definidos: um objetivo, um ID único, uma identidade conhecida, conjunto de itens de dados estruturados e identificador de versão. Para um Recurso ser válido, ele deve seguir suas definições e pode ser representado em um formato XML ou JSON [*HL7 FHIR Resource Index*, n.d.]. Estes formatos são amplamente utilizados na Web e comuns para maioria dos desenvolvedores. Os Recursos podem abstrair conceitos desde níveis clínicos em saúde, como procedimentos cirúrgicos, até níveis administrativos, como dispositivos físicos utilizados para prover cuidados em saúde.

Acompanha-se um crescimento exponencial de utilização de aplicações em nuvem desde a última década. A área da saúde, certamente não estaria fora desse crescimento, podendo aproveitar diversas vantagens que essas aplicações oferecem, como: maior escalabilidade, maior segurança nos dados, maior capacidade de integração, entre outras. Entre as soluções disponíveis no mercado, existe a Google Cloud Computing, que oferece diversos serviços em nuvem, inclusive abrangendo a área da saúde. Na Google Cloud, é possível utilizar uma ferramenta de linha de comando denominada *gcloud* para criar e gerenciar recursos do Google Cloud. Esta ferramenta é utilizada durante o tutorial, em conjunto com o recurso que disponibiliza uma API FHIR para utilização. Nesta API é possível representar e trocar informações médicas eletrônicas utilizando o padrão FHIR.

Os serviços em nuvem em conjunto com o padrão FHIR oferecem uma ótima opção para construir aplicações voltadas à saúde de forma escalável e de fácil manutenção. Ao relacionar esta opção com uma arquitetura baseada em serviços, é possível definir aplicações de responsabilidade únicas e bem definidas. Como consequência, pode-se obter ganhos na interoperabilidade e disponibilidade dos sistemas entre diversos parceiros.

O trabalho aqui desenvolvido tem como objetivo demonstrar a criação de um servidor HL7® FHIR na prática utilizando recursos de computação em nuvem. Na seção 2, é apresentado o tutorial completo, desde a criação de conta no Google Cloud até a criação do servidor HL7® FHIR. Na seção 3, o código criado é disponibilizado em um repositório. Na seção 4, é demonstrado como evitar cobranças adicionais no serviço Google Cloud e na seção 5 são apresentadas as considerações finais sobre o trabalho.

## **2. Tutorial**

### **2.1. Criação de conta na Google Cloud**

Primeiramente, é necessário criar uma conta na Google Cloud, onde está o serviço que será utilizado no andamento do tutorial. Para criar uma conta gratuita, siga o passo a

passo disponível no seguinte endereço: <<https://console.cloud.google.com/freetrial>>. Se você não estiver com uma sessão iniciada em uma conta Google, a página irá solicitar que você entre com uma. Caso não possua uma conta Google, será necessário criar uma para prosseguir. Após terminar o cadastro, você será direcionado para a tela inicial do Google Cloud.

O Google Cloud oferece um teste gratuito de 90 dias, disponibilizando um crédito de 300 dólares para serem utilizados durante esse período. Além do valor fornecido, o serviço possui um nível gratuito de acesso respeitando alguns limites de utilização. Importante ressaltar que caso os limites sejam ultrapassados, os créditos serão utilizados. Após o término dos créditos, o serviço pode possuir cobranças adicionais. Ressaltamos que durante os testes de execução sequencial deste tutorial os custos totais não ultrapassaram centavos de dólares americanos (USD). Entretanto, é fundamental ficar atento aos limites, pois quaisquer cobranças serão realizadas no cartão cadastrado na etapa dois de registro na plataforma. Para detalhes sobre os preços destes serviços consulte a documentação específica em <<https://cloud.google.com/healthcare/pricing>>.

## **2.2. Criação do projeto e ativação da API Healthcare**

Após a criação da conta, é necessário criar um projeto para ativar e usar os recursos da plataforma. Algumas informações sobre os serviços criados agora serão necessárias na seção 2.3 do tutorial e, portanto, aparecem destacadas em letras maiúsculas entre parênteses a seguir no texto. Sugere-se a criação de um projeto exclusivo para o uso neste tutorial, pois facilitará a exclusão de todos os recursos ao final. Siga as orientações da documentação oficial para a criação de um projeto: <<https://cloud.google.com/resource-manager/docs/creating-managing-projects>>. Anote a identificação do projeto criado (PROJECT\_ID).

Para criação do servidor API HL7® FHIR é necessário ativar a API Healthcare. Na plataforma Google Cloud, acesse o menu de navegação e clique em “APIs e serviços”. Na página, utilize a barra de buscas para encontrar a API Healthcare e depois clique em “ativar”.

Após, é necessário criar um conjunto de dados. Acesse <[https://console.cloud.google.com/healthcare/browser?\\_ga=2.183729954.589953169.1603326473-595185282.1602989361](https://console.cloud.google.com/healthcare/browser?_ga=2.183729954.589953169.1603326473-595185282.1602989361)> e crie um conjunto de dados. Após, clique no conjunto criado e clique em “criar armazenamento de dados”. Selecione “FHIR” como armazenamento e a versão R4 ao final da página. Em caso de dúvidas consulte a documentação oficial [*FHIR | Cloud Healthcare API*], n.d.; [*Creating and Managing FHIR Resources | Cloud Healthcare API*], n.d.]. Anote a localização do conjunto de dados (LOCATION), o nome do conjunto de dados (DATASET\_ID), o nome do armazenamento de dados (FHIR\_STORE\_ID).

## **2.3. Criação e consumo da API HL7® FHIR**

O único pré-requisito para prosseguir com o tutorial é possuir o Docker instalado. Após certificar-se que ele está instalado e configurado em sua máquina, faça o download do repositório do tutorial:

```
git clone https://github.com/gbnegrini/hl7-fhir-tutorial.git
```

Dentro da pasta está disponível uma *Dockerfile* para criação do ambiente do tutorial. A imagem do ambiente é baseada na distribuição Alpine do Linux e conta com a instalação do Cloud SDK, o qual é necessário para a utilização dos recursos da Google Cloud via terminal. Além disso, será automaticamente instalado o Windows PowerShell, terminal escolhido devido à facilidade de interação com REST APIs e estruturas no formato JSON.

```
# processe a imagem definida na Dockerfile
docker build -t gcloud-fhir .

# instancia um container nomeado `gcloud` com powershell interativo
docker run -ti --name gcloud gcloud-fhir pwsh
```

Defina algumas variáveis (precedidas por `\$`) de configuração com as informações registradas anteriormente na interface web durante a criação do projeto e ativação da API Healthcare. Substitua, nos comandos abaixo, conforme o exemplo: ``$PROJECT_ID=ID_DO_SEU_PROJETO``.

```
# defina as variáveis de configuração do projeto criado anteriormente na
interface web
$PROJECT_ID=<PROJECT_ID>
$LOCATION=<LOCATION>
$DATASET_ID=<DATASET_ID>
$FHIR_STORE_ID=<FHIR_STORE_ID>
$NAME="usertest"
```

Agora, vamos realizar o login e autenticar sua conta que foi criada anteriormente. Digite o comando abaixo e clique no link que será impresso no terminal. Você será redirecionado para o serviço de login do Google. Autorize a aplicação e copie e cole o código disponibilizado de volta no seu terminal.

```
gcloud auth login
```

Os comandos seguintes completam o processo de criação de uma conta de serviço e autenticação. Será criada uma conta com permissões de proprietário e chaves de acesso serão criadas localmente no container. Sempre mantenha protegida sua conta e suas credenciais.

```

especificar o projeto que estamos utilizando
gcloud config set project $PROJECT_ID

# cria um usuário para a conta de serviço
gcloud iam service-accounts create $NAME

# vincular o usuário ao projeto com escopo de proprietário
gcloud projects add-iam-policy-binding $PROJECT_ID
--member=serviceAccount:$NAME@$PROJECT_ID.iam.gserviceaccount.com --role
"roles/owner"

# cria localmente as chaves de autenticação
gcloud iam service-accounts keys create CREDENTIALS.json --iam-account
$NAME@$PROJECT_ID.iam.gserviceaccount.com
$env:GOOGLE_APPLICATION_CREDENTIALS="CREDENTIALS.json"

```

Pronto, agora o ambiente está configurado e autenticado para uso com o projeto criado na Google Cloud. Defina variáveis com as credenciais criadas e com um cabeçalho de autorização, elas serão necessárias em todas as requisições que faremos para o servidor FHIR:

```

$cred = gcloud auth application-default print-access-token
$headers = @{ Authorization = "Bearer $cred" }

```

Vamos criar um paciente fictício utilizando apenas alguns dados, como nome (*name.given*), sobrenome (*name.family*), gênero (*gender*), data de nascimento (*birthDate*) e tipo do recurso (*Patient*). Para isso, estruture os dados em formato JSON e de acordo com a especificação do recurso Patient do padrão HL7® FHIR R4:

```

$patient = '{
  "name": [
    {
      "use": "official",
      "family": "Nusite",
      "given": [
        "Darcy"
      ]
    }
  ],
  "gender": "female",
  "birthDate": "1980-01-01",
  "resourceType": "Patient"
}'

```

Agora, envie uma requisição HTTP POST para a Uri específica do recurso Patient no servidor FHIR:

```

Invoke-RestMethod `
  -Method Post `
  -Headers $headers `
  -ContentType: "application/fhir+json; charset=utf-8" `
  -Body $patient `

```

```
-Uri  
"https://healthcare.googleapis.com/v1/projects/$PROJECT_ID/locations/$LOCATION/datasets/$DATASET_ID/fhirStores/$FHIR_STORE_ID/fhir/Patient" |  
ConvertTo-Json
```

Caso a requisição tenha sido bem-sucedida o servidor retornará como resposta o mesmo JSON com informações extras como a `id` criada para o paciente (“id”: “PATIENT\_ID”) e um campo `meta` com informações sobre a data de atualização e a versão do recurso criado. Agora, substitua no comando abaixo e armazene o `id` do paciente em uma variável:

```
$PATIENT_ID=<id>
```

Suponha que a paciente tenha procurado atendimento pois estava com dificuldades de respirar. Crie um encontro para descrever a interação da paciente com um profissional, utilizando a estrutura do recurso Encounter do padrão HL7® FHIR R4 e os dados da queixa (*reasonCode.text*), identificação da paciente (*subject.reference*) e tipo de recurso (*resourceType*):

```
$encounter = '{  
  "status": "finished",  
  "class": {  
    "system": "https://www.hl7.org/fhir/R4/",  
    "code": "IMP",  
    "display": "inpatient encounter"  
  },  
  "reasonCode": [  
    {  
      "text": "A paciente relatou dificuldades de respiração."  
    }  
  ],  
  "subject": {  
    "reference": "Patient/$PATIENT_ID"  
  },  
  "resourceType": "Encounter"  
}'  
  
# substitua automaticamente o valor da variável do paciente ($) no JSON  
$encounter = $ExecutionContext.InvokeCommand.ExpandString($encounter)  
  
# realize o POST do encontro para o recurso Encounter  
Invoke-RestMethod `   
  -Method Post `   
  -Headers $headers `   
  -ContentType: "application/fhir+json; charset=utf-8" `   
  -Body $encounter `   
  -Uri  
"https://healthcare.googleapis.com/v1/projects/$PROJECT_ID/locations/$LOCATION/datasets/$DATASET_ID/fhirStores/$FHIR_STORE_ID/fhir/Encounter" |  
ConvertTo-Json
```

Caso a requisição tenha sido bem-sucedida o servidor retornará novamente o JSON submetido com a inserção de informações adicionais como um campo `meta` e a identificação do encontro (“id”: “ENCOUNTER\_ID”). Substitua no comando abaixo e armazene o `id` do encontro em uma variável:

```
$ENCOUNTER_ID=<id>
```

Durante o encontro, o profissional de saúde realizou a medição da taxa respiratória da paciente. Dessa forma, crie uma observação para registrar essa medição que ocorreu no encontro. Utilize a estrutura do recurso *Observation* do padrão HL7® FHIR R4 para inserir os dados do código da terminologia LOINC referente a taxa de respiração (*code.coding*), valores e unidade de medição (*valueQuantity*) e identificação da paciente (*subject.reference*) e do encontro (*encounter.reference*).

```
$observation = '{
  "resourceType": "Observation",
  "status": "final",
  "subject": {
    "reference": "Patient/$PATIENT_ID"
  },
  "effectiveDateTime": "2020-01-01T00:00:00+00:00",
  "code": {
    "coding": [
      {
        "system": "http://loinc.org",
        "code": "9279-1",
        "display": "Respirações"
      }
    ]
  },
  "valueQuantity": {
    "value": 8,
    "unit": "respirações/min"
  },
  "encounter": {
    "reference": "Encounter/$ENCOUNTER_ID"
  }
}'

# substitua automaticamente o valor da variável na mensagem
$observation = $ExecutionContext.InvokeCommand.ExpandString($observation)

# realize o POST para o recurso Observation
Invoke-RestMethod `
  -Method Post `
  -Headers $headers `
  -ContentType: "application/fhir+json; charset=utf-8" `
  -Body $observation `
  -Uri
  "https://healthcare.googleapis.com/v1/projects/$PROJECT_ID/locations/$LOCATION/datasets/$DATASET_ID/fhirStores/$FHIR_STORE_ID/fhir/Observation" |
  ConvertTo-Json
```

Mais uma vez, em caso de requisição bem-sucedida, o servidor retornará uma resposta em JSON. Observe na resposta se as identificações do paciente e do encontro estão corretamente referenciados na observação. Por fim, para realizar uma consulta ao servidor basta realizar uma requisição HTTP GET para a Uri do recurso que deseja e fornecer um parâmetro de busca. Abaixo, um exemplo para consulta da paciente utilizando seu identificador:

```
Invoke-RestMethod `
  -Method Get `
  -Headers $headers `
  -Uri
  "https://healthcare.googleapis.com/v1/projects/$PROJECT_ID/locations/$LOCATION/datasets/$DATASET_ID/fhirStores/$FHIR_STORE_ID/fhir/Patient/$PATIENT_ID" | ConvertTo-Json
```

## 2.4. Encerramento e deleção dos recursos

A utilização de recursos computacionais da Google Cloud pode implicar na cobrança pelos serviços enquanto são utilizados. A forma mais fácil de garantir o encerramento da cobrança é a exclusão do projeto criado para esse tutorial. Para isso, faça login na plataforma Google Cloud e acesse a página de projetos (Menu de navegação > IAM e administrador > Gerenciar recursos). Após, selecione o projeto que foi criado para o tutorial e clique em excluir.

Para liberar os recursos computacionais da sua máquina (excluir o contêiner e a imagem) execute os comandos abaixo:

```
# saia do contêiner em que estava executando o tutorial
exit

# parar a execução do contêiner
docker stop gcloud

# excluir o contêiner
docker rm gcloud

# excluir a imagem
docker image rm gcloud-fhir
```

## 3. Disponibilização do código

Os arquivos, bem como a totalidade dos códigos e comandos utilizados, podem ser encontrados no repositório: <<https://github.com/gbnegrini/hl7-fhir-tutorial>>.

## 4. Considerações finais

Neste tutorial foi demonstrada a implementação prática de um servidor HL7® FHIR e a criação de alguns recursos utilizando o padrão. A utilização de APIs e serviços em



nuvem pode ser uma alternativa para construção rápida e sob demanda de aplicações, pois fornece uma estrutura pronta e escalável de fácil configuração. Aqui, demonstramos apenas alguns dos recursos existentes no padrão.

Para aqueles que desejam explorar mais a ferramenta e o padrão, sugere-se o cadastro de outros recursos; a construção de consultas complexas para diferentes endpoints do servidor; e até mesmo o desenvolvimento de uma aplicação cliente simulando um formulário de cadastro e consulta para consumir a API. Para testes avançados recomendamos a utilização do simulador de geração de pacientes Synthea™, gratuito, que fornece dados fictícios (porém realísticos) de histórico médico completo, incluindo medicamentos, alergias, encontros médicos, determinantes sociais da saúde, no padrão HL7® FHIR.

## **Referências**

Creating and managing FHIR resources | Cloud Healthcare API. (2020). Google Cloud.  
<https://cloud.google.com/healthcare/docs/how-tos/fhir-resources>

FHIR | Cloud Healthcare API |. (2020). Google Cloud.  
<https://cloud.google.com/healthcare/docs/concepts/fhir>

HL7 FHIR Resource Index. (2020). HL7. <https://www.hl7.org/fhir/resourcelist.html>

RFH: Resources For Healthcare. (2011). HL7. <https://hl7.org/fhir/2011Aug/>