

## GUIA FHIR PARA O DESENVOLVEDOR TICS NA ASSISTÊNCIA MÉDICA

*Permitir que a troca de informações de saúde seja mais rápida e segura tem sido um dos principais objetivos do setor de saúde....*



O compartilhamento contínuo de informações não é apenas uma necessidade iminente aos prestadores de serviços de saúde, mas também é um ponto problemático para desenvolvedores de software e profissionais de TICS encarregados de integrar aplicativos e sistemas médicos.

Compartilhar informações de saúde precisa ser mais simples e fácil, mas sem colocar em risco a segurança de informações confidenciais. O problema é a variedade de métodos usados para mover os dados. A falta de padrões e de métodos acordados sempre faltava na 'fórmula'.

Reunimos neste guia, para fornecer aos desenvolvedores de TICS uma visão abrangente do FHIR, incluindo o básico de como o FHIR funciona, como implementá-lo e valiosos recursos de aprendizado para quem deseja aprofundar-se ainda mais no FHIR.

Neste guia, abordamos:

1. [O QUE É FHIR?](#)
2. [COMO FUNCIONA](#)
3. [BENEFÍCIOS](#)
4. [UMA VISÃO GERAL DO FHIR PARA DESENVOLVEDORES DE TICS](#)
5. [COLOCANDO O FHIR PARA TRABALHAR PARA VOCÊ](#)
6. [RECURSOS DE APRENDIZADO](#)
7. [LEITURAS ADICIONAIS PARA DESENVOLVEDORES TICS](#)

---

## 1 - O QUE É FHIR?

O FHIR ([Fast Healthcare Interoperability Resources](#)) é um conjunto relativamente novo de padrões que fornece um mecanismo para a troca de dados entre aplicativos de assistência médica.

[O Health Level Seven International](#) patrocinou pela primeira vez o FHIR (pronuncia-se “fire”) em 2011 e permanece em desenvolvimento ativo, sendo sua versão normativa R4 publicada no início de janeiro de 2019.

Assim o [HL7](#) define o FHIR:

*“O escopo do FHIR é amplo, cobrindo os cuidados humanos e veterinários, atendimento clínico, saúde pública, ensaios clínicos, administração e aspectos financeiros. O padrão é destinado ao uso global e em uma ampla variedade de arquiteturas e cenários.”*

Você pode se perguntar por que o FHIR é mais interessante quando comparado a outros padrões de interoperabilidade, como a arquitetura para documentos clínicos ([CDA](#)) ou a continuidade dos registros de atendimento ([CCR](#)).

A diferença mais significativa entre os padrões mais populares de interoperabilidade e o FHIR é sua velocidade de desenvolvimento, implantação e entrada em produção. O FHIR suporta quatro métodos para a troca de informações, incluindo o [REST](#), que é o padrão usado pela Internet.

---

## 2- COMO FUNCIONA O FHIR

Para colocar em perspectiva, o FHIR funciona como a internet, só que para registros de saúde. O Google usa tecnologias semelhante as do FHIR para fornecer resultados de pesquisas quando você procura, por exemplo, pelo melhor restaurante de sushi em sua região.

Como não há um único servidor centralizado com um banco de dados contendo todos os restaurantes de sushi em sua região, o Google rastreia a Internet e cataloga os dados para fornecer resultados rápidos e precisos, assim que você os solicita.

Por mais grosseira que essa descrição possa parecer, isso é o FHIR em poucas palavras. Os mesmos padrões usados para proteger as informações bancárias e outros meios de autenticação e segurança também estão presentes no FHIR.

O FHIR é seguro, além de rápido e flexível. O FHIR atende a todos, desde o desenvolvedor familiar local até uma clínica voluntária no meio de um deserto ou selva. Porém, assim como os antecessores padrões HL7, o FHIR tem sua parcela de desafios e isto é bom saber.

O maior desafio é uma questão importante que existe na [implementação do HL7](#): é possível que dois aplicativos diferentes implementem versões diferentes do FHIR, o que significa que os sistemas não seriam interoperáveis, apesar de ambos terem implementado o FHIR de alguma forma.

Também é possível que os desenvolvedores implementem apenas algumas das APIs FHIR disponíveis ou implementem parcialmente estas APIs. O resultado final em qualquer cenário é o mesmo: *aquém da interoperabilidade que o setor está buscando.*

---

### 3 - A NECESSIDADE DE IMPLEMENTAÇÃO DO FHIR

Para os desenvolvedores de TICS, implementar o FHIR é mais rápido que outros padrões de interoperabilidade, porque é mais poderoso, mas fácil de aprender e desenvolver.

Começaremos explicando o básico e fornecendo algumas ideias para você desenvolver, mas é realmente um padrão que você pode moldar às suas necessidades.

Atualmente, os serviços de saúde estão enterrados em montanhas de silos de dados, sendo grande parte deles, desorganizada ou fora de alcance.

Os profissionais responsáveis pelo atendimento de emergência, podem não ter o histórico do paciente de que precisam, o que pode forçá-los a adivinhar ou basearem seu tratamento nas informações fornecidas pelo próprio paciente.

Porém, se o paciente não puder fornecer informações sobre seu histórico ou alergias, o problema se agrava.

O FHIR, se usado como padrão, remove esses problemas e coloca a história do paciente na ponta dos dedos do médico. Um excelente exemplo de como o FHIR facilita a troca de dados é esse [navegador de pacientes](#) disponível no GitHub.

É um aplicativo feito em HTML5, simples, mas que permite navegar pelos dados da mesma maneira que você procura filmes no seu serviço de streaming favorito. Você pode conferir uma [demonstração aqui](#).

---

## 4 - VISÃO GERAL DO FHIR PARA OS DESENVOLVEDORES TICS

O pensamento inicial quando você lê sobre o FHIR, e o para que ele foi projetado, você pensará que seu foco sempre estará na saúde humana. Os provedores de assistência médica lutam com a troca de informações e os desenvolvedores lutam para fornecer maneiras de fazê-lo.

Todos têm suas próprias ideias e usam métodos que podem não funcionar bem uns com os outros. O FHIR trabalha para todos, tanto para a saúde humana como para os veterinários.

[Os recursos](#) no FHIR representam os dados da assistência médica. Os recursos são usados para criar uma imagem dos dados para troca no nível de instância. Estes recursos facilitam a troca de dados e fornecem um meio para armazená-los, como uma página da Web faz na Internet.

Para manter esse processo e a estrutura dos recursos consistentes, os recursos *devem* incluir o seguinte:

- Um URL para identificar o recurso
- Metadados para ajudar nas pesquisas e na catalogação
- Um resumo XHTML para que humanos possam lê-lo
- Definições para os elementos de dados
- Uma estrutura de extensibilidade específica para assistência médica

Os [145 tipos de recursos](#) suportados pelo FHIR devem estar nos formatos XML, JSON ou RDF. Um recurso do tipo Patient representado em XML se parecer com [isso](#), já o mesmo paciente representado no JSON se parece com [isso](#).

A captura de tela abaixo ilustra como um paciente pode ser representado como um objeto FHIR em JSON, com a codificação XML definida na especificação:

```
1.  {
2.    "resourceType": "Patient",
3.    "id" : "23434",
4.    "meta" : {
5.      "versionId" : "12",
6.      "lastUpdated" : "2014-08-18T15:43:30Z"
7.    }
8.    "text": {
9.      "status": "generated",
10.     "div": "<!-- Snipped for Brevity -->"
11.    },
12.    "extension": [
13.      {
14.        "url": "http://example.org/consent#trials",
15.        "valueCode": "renal"
16.      }
17.    ],
18.    "identifier": [
19.      {
20.        "use": "usual",
21.        "label": "MRN",
22.        "system": "http://www.goodhealth.org/identifiers/mrn",
23.        "value": "123456"
24.      }
25.    ],
26.    "name": [
27.      {
28.        "family": [
29.          "Levin"
30.        ],
31.        "given": [
32.          "John"
33.        ],
34.        "prefix": [
35.          "Mr"
36.        ],
37.        "suffix": [
38.          "MD"
39.        ]
40.      }
41.    ]
42.  }
```

*Captura parcial de tela via [HL7](#)*

O JSON é mais fácil e você pode encontrar usos para ele em outros aplicativos também. O XML pode ser usado se for o mais popular entre sua equipe de desenvolvimento. Aqui está uma visão mais detalhada de quais [elementos compõem um recurso](#):

- **resourceType** (linha 2): um dos 145 tipos de recursos definidos pelo FHIR. Este é um elemento obrigatório.



- **ID** (linha 3): este elemento é o ID do recurso e deve estar sempre presente quando um recurso é trocado. A única exceção é durante a operação de criação.
- **meta** (linhas 4 a 7): esse elemento fornece usos comuns e outras informações de contexto para todos os recursos. Geralmente está presente, mas nem sempre, e alguns [itens de metadados](#) podem ser necessários em determinadas implementações ou contextos de uso.
- **text** (linhas 8 a 11): este elemento consiste em XHTML que fornece uma representação legível por humanos para o recurso. Não é um elemento obrigatório, mas é recomendado.
- **extension** (linhas 12 a 17): esse elemento opcional inclui extensões definidas pela estrutura de extensibilidade.
- **data** (linhas 18 a 43): elemento opcional, esta seção inclui um conjunto diferente de elementos de dados definidos para cada tipo de recurso.

É improvável que você encontre um software que não consiga lidar com XML ou JSON. No entanto, caso ocorra, você pode usar os formatos [RDF e Turtle](#) para criar seus recursos.

Os formatos XML, JSON e RDF são usados regularmente pelos sites e aplicativos da Web controlados por dados. Sendo honesto, não importa qual seu predileto, desde que permaneça consistente e com a correta sintaxe.

Os recursos podem receber um URL atribuído pelo servidor para ajudar a identificá-lo, além de permitir que este seja recuperado. Semelhante a uma página WEB, que pode usar um URL relativo, o URL nem sempre é definido no recurso.

Se você deseja que cópias do recurso apontem para o original o tempo todo, é necessário adicionar uma URL dentro do recurso. No entanto, isso é desnecessário na maioria dos casos. Você terá que tomar essa decisão.

Você pode manipular os dados usando uma [API REST](#) definida pelo FHIR. Você pode usar POST, DELETE, GET e PUT para criar, ler ou gerenciar recursos.

Por exemplo, você pode usar **"PUT <https://www.example.com/path-to-your-resource/resource-type/resource-id>"** para atualizar um recurso quando os dados forem alterados ou precisarem ser atualizados por qualquer motivo.

Nosso exemplo é bastante básico e você acabará usando um sistema de identidades de URL e recursos mais elaborados, mas essa é a ideia geral por trás do uso de URLs e da API REST, isto é, desde que os URLs sejam codificados usando UTF-8 e consistentes, tudo deve funcionar como navegar no seu site ou blog favorito para ler uma postagem.

As mesmas camadas de segurança e criptografia que protegem as informações do seu cartão de crédito enquanto você faz compras na Amazon ou monitora sua conta corrente de forma online, estão, entre a pessoa que está tentando acessar o servidor e os recursos.

Embora pareça que você está criando uma Internet para registros de assistência médica, os dados são seguros e criptografados e estão disponíveis apenas para pessoas autorizadas, assim como em sua instituição financeira.

---

## 5 - COLOCANDO O FHIR PARA TRABALHAR PARA VOCÊ

Na área da saúde, os recursos mais comuns incluem históricos, condições e tratamentos. Na realidade, é um pouco mais complicado que isso, mas para nossos propósitos, vamos supor que seja simples assim.

A primeira tarefa da cadeia de desenvolvimento é criar um servidor e uma API para navegadores ou aplicativos de telefone. O processo é semelhante a criar um aplicativo web para acessar dados, o que faz parte do apelo do FHIR.

Nesse caso, você precisa de um servidor, um banco de dados e os recursos do FHIR para pesquisar e consumir.

Você pode armazenar dados do FHIR em um banco de dados e pular algumas das etapas de criação de recursos. No entanto, você precisará de uma maneira de mapear os recursos do FHIR como objetos e manter esse mapeamento e o banco de dados.

Em vez disso, é mais fácil armazenar os recursos do FHIR como documentos e usar algo como o [MongoDB](#) para rastreá-los. O MongoDB simplifica muito a organização e a alteração de documentos.

Explicar o MongoDB está fora do escopo deste artigo, mas recomendamos que você o analise mais profundamente e avalie suas opções antes de criar um banco de dados relacional e os aplicativos para FHIR.

É rápido e flexível, além disso, eles fizeram isso com os documentos em mente. Esse é um forte ponto de venda para o uso do MongoDB com recursos FHIR.

Estamos fazendo algumas suposições agora e assumindo que você seguiu nossos conselhos sobre o uso de um banco de dados de documentos ao invés do MySQL ou algo semelhante.

Muitas das etapas a seguir requerem conhecimento de suas necessidades e capacidade de desenvolvimento, que não possuímos. Por esse motivo, consulte essas informações apenas como um guia e faça sua lição de casa no software e nos dispositivos que planeja usar.

Se você está apenas desenvolvendo um aplicativo ou navegador, pode testá-los sem criar muitos pacientes fictícios e criar um servidor usando alguns dos servidores públicos de testes encontrados [aqui](#).

Você encontrará servidores configurados para praticamente qualquer cenário ou linguagem que você deseja usar.

FHIR © HL7.org | Server Home | Health Intersections FHIR Server v1.0.275 | FHIR Version 4.0.0 | User: ANONYMOUS (Unknown)

## Health Intersections FHIR Server

Welcome ANONYMOUS (Unknown) [or use [Secure API](#)]

This is an implementation of the RESTful FHIR specification found at <http://hl7.org/fhir>. This server has a baseURL (see the FHIR specification) of "http://test.fhir.org/r4". This server defines some [extensions to the API](#), and also offers [Terminology Services](#) or (or you can browse [SNOMED-CT](#) or [LOINC](#) directly) . If you login through OAuth, you can also [Register a client](#) or get your bearer token (use this to get access to the secure API without needing OAuth login in the application).

GDPR-Disclosure: All access to this server is logged as AuditEvent Resources, and these store your ip address (and logged in user, if one exists). Also, your IP address is logged with Google Analytics for building geomaps of server usage. Your continued use of the API constitutes agreement to these terms. See [\[link\]](#) for erasure requests.

System Operations:

- [Conformance Profile](#) (or as xml (or JSON)
- [Tags defined on this System](#)
- [General Search](#)
- [History for the whole system](#) (History of all resources)
- [Upload Services](#)
- [Create/Edit a new resource based on the profile:](#)

This server hosts the following resource types:

Type	# stored	Operations	Type	# stored	Operations
Account	4	<a href="#">Profile</a> <a href="#">Updates</a> <a href="#">Search</a> <a href="#">Tags</a>	Measure	7	<a href="#">Profile</a> <a href="#">Updates</a> <a href="#">Search</a> <a href="#">Tags</a>
ActivityDefinition	9	<a href="#">Profile</a> <a href="#">Updates</a> <a href="#">Search</a> <a href="#">Tags</a>	MeasureReport	23	<a href="#">Profile</a> <a href="#">Updates</a> <a href="#">Search</a> <a href="#">Tags</a>
AdverseEvent	1	<a href="#">Profile</a> <a href="#">Updates</a> <a href="#">Search</a> <a href="#">Tags</a>	Media	4	<a href="#">Profile</a> <a href="#">Updates</a> <a href="#">Search</a> <a href="#">Tags</a>
AllergyIntolerance	7	<a href="#">Profile</a> <a href="#">Updates</a> <a href="#">Search</a> <a href="#">Tags</a>	Medication	23	<a href="#">Profile</a> <a href="#">Updates</a> <a href="#">Search</a> <a href="#">Tags</a>

*Captura de tela via [HL7](#)*

Este é, novamente, um exemplo altamente simplificado, mas assumiremos, para fins ilustrativos, que as necessidades associadas a este projeto são para um hospital ou consultório. Sob esses termos, as três coisas que você precisa fazer incluem o seguinte:

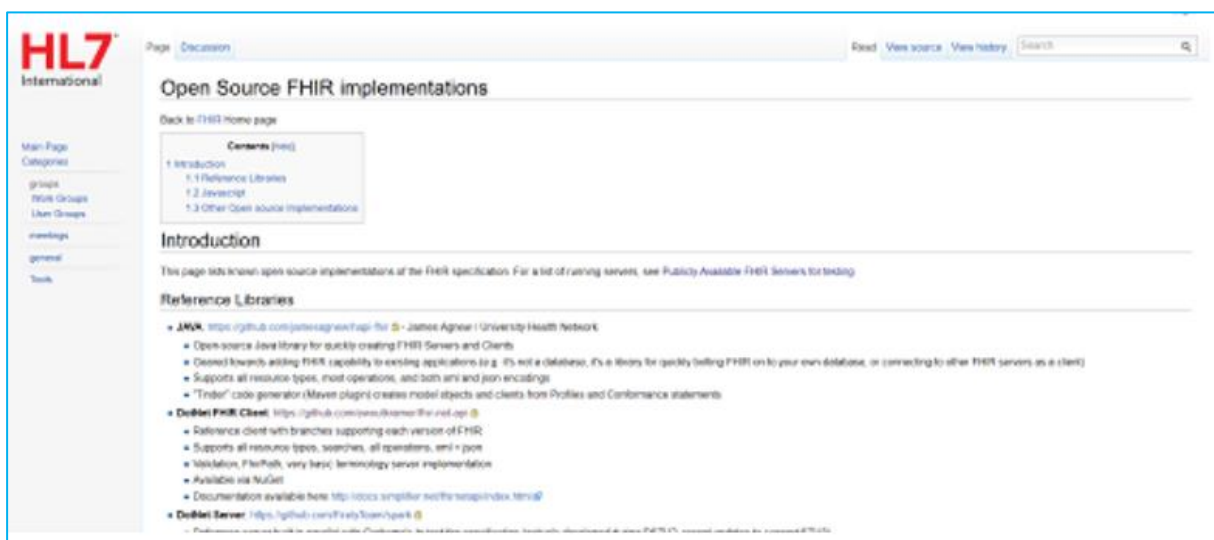
- Pesquisar por um ou mais pacientes
- Criar e armazenar pacientes
- Recuperar dados específicos do paciente

A primeira etapa é criar um servidor, mas você precisa fazer algumas opções neste momento antes de começar e perceber que existe uma opção melhor. O uso de

servidores de código aberto é uma maneira econômica de começar. O software de código aberto é gratuito na maioria das vezes, e você pode obter suporte on-line através de fóruns e wikis. Alguns desenvolvedores também oferecem ajuda e suporte para sua implementação por uma taxa.

A menos que você tenha um plano para um servidor, um excelente lugar para começar sua jornada e obter algumas dicas sobre o que outros desenvolvedores estão fazendo, é consultando algumas [implementações e servidores FHIR de](#) código aberto.

Você pode analisar o código, como eles abordam os problemas e os resolvem. Você pode encontrar uma resposta boa e evitar o desenvolvimento de sua própria solução de servidor.



Captura de tela via [HL7](#)

Isto resolve o armazenamento dos pacientes e o acesso aos dados. Agora, você só precisa de uma API. Você pode encontrar um excelente exemplo de servidor e API FHIR [aqui](#) disponível no Microsoft Azure.

O Azure é uma boa opção se você não tiver recursos ou orçamento para criar seu próprio data center. Uma rápida pesquisa no Google fornecerá soluções semelhantes para Amazon, DigitalOcean ou seu serviço favorito.

## FHIR Server Samples

This repository contains example applications and scenarios that show use of the [FHIR Server for Azure](#) and the [Azure API for FHIR](#).

The scenario is meant to illustrate how to connect a web application to the FHIR API. The scenario also illustrates features such as the SMART on FHIR Active Directory Proxy. It can be deployed using the Open Source version of the FHIR server:



Or the Azure API for FHIR PaaS server:



In both cases a storage account will be deployed and in this storage account there is a BLOB container called `fhirimport`, patient bundles generated with [Synthea](#) can be dumped in this storage container and they will be ingested into the FHIR server. The bulk ingestion is performed by an Azure Function.

*Captura de tela via [GitHub](#)*

Um recurso importante e um problema com o FHIR é a sua flexibilidade. Os dois parágrafos anteriores podem ser tudo o que você precisa para começar a desenvolver um servidor FHIR e uma API funcional.

É verdade que não adicionamos nenhum método de segurança ou autenticação à nossa configuração, mas nosso objetivo é fazê-la funcionar e realizar testes.

[O HAPI FHIR](#) pode ser uma solução mais direta, sendo que você pode pular toda a codificação e/ou usar qualquer servidor comercial. O HAPI FHIR fornece este [guia](#) que possui a capacidade de criar um banco de dados de assistência médica e a API FHIR em 15 minutos.

Demoramos cerca de 40 minutos para rodar usando o Ubuntu como nosso servidor em uma máquina virtual. No entanto, esta é uma excelente oportunidade de aprendizado, e sugerimos que tente. Ele usa Java, Tomcat e Apache Derby para o banco de dados. Você pode usar outro banco de dados se não gostar do Derby.

Felizmente, demonstramos a flexibilidade do FHIR e fornecemos alguns exemplos de trabalho do mundo real para suas análises e validações. No entanto, mal arranhamos a superfície do que o FHIR é capaz e do que os desenvolvedores estão fazendo com ele.

O próximo passo é aprofundar-se no código e encontrar oportunidades de aprendizado. Você pode aprender muito olhando o trabalho e as ideias de outras pessoas.

---



## 6 - RECURSOS DE APRENDIZAGEM FHIR

### Interoperabilidade

**HANDS-ON MIRTH CONNECT E PADRÕES HL7 & DICOM**

---

AULAS ON-LINE AO VIVO

Edição Nacional - Noturna  
Edição Internacional - Diurna

Consulte o programa e inscreva-se

MÓDULOS

**HL7**  
HEALTH LEVEL 7

**mirthconnect**



**FHIR**



**IHTSDO**  
SNOMED  
International

*Captura de tela via [INTEROPERA](#) - Clique na imagem para mais informações*

O melhor lugar para aprender mais sobre o FHIR é através dos cursos da [InterOpera](#). Acesse este [link](#) para consultar os módulos disponíveis e realize sua inscrição.

Os instrutores são especialistas em FHIR e trabalham com FHIR, portanto, não há professores de livros didáticos. Você não precisa entender o FHIR para ter sucesso no curso.

A [InterOpera](#) oferece, além de treinamentos em padrões HL7 e terminologias, toda a consultoria necessária para seus projetos, sejam de interoperabilidade e/ou para a construção de data-lakes para mineração e descoberta de valor em seus dados.

A InterOpera o leva mais a fundo no FHIR oferecendo treinamentos avançados usando recursos adicionais que incluem:

- [O Centro de Aprendizagem](#): aqui você encontra uma ampla variedade de treinamentos em saúde, incluindo o HL7 FHIR.
- [Biblioteca Nacional de Medicina](#): uma pesquisa rápida aqui traz 263 tópicos relacionados ao FHIR para você conferir.
- [HL7](#): eles oferecem, mais do que o treinamento em fundamentos do HL7 FHIR.

Verifique com faculdades e universidades em sua área para ver se eles oferecem aulas de educação continuada no FHIR ou as ferramentas necessárias para que o FHIR funcione para você.

Não confie somente nos tutoriais da Internet e no treinamento gratuito para começar. Você perderá questões críticas, como resolver problemas relacionados à proteção de dados para uso com aplicativos móveis ou criar navegadores FHIR que resistem a hackers.

---

## NOTAS FINAIS

O FHIR é poderoso e pode derrubar a hierarquia atual de interoperabilidade na área da saúde. Indiscutivelmente, o FHIR está atropelando outros padrões, mas é improvável que soluções funcionais sejam substituídas em curto prazo por FHIR. Novos projetos, sim, estão implementando FHIR.

Uma vez que o mercado tenha uma solução funcional, estes tendem a mantê-la. O FHIR é uma melhoria significativa em relação aos padrões anteriores e atuais, mas a implementação inconsistente do FHIR continua sendo um obstáculo no caminho para a interoperabilidade.

O FHIR não é a única maneira de aplicar os benefícios das APIs aos sistemas de informações de saúde. Uma solução de API como o [Integrate](#) pode tornar a interoperabilidade prática e econômica, eliminando as barreiras usuais à interoperabilidade.

A integração oferece recursos de leitura e gravação com qualquer EHR por meio de um conjunto robusto de [APIs REST universais em tempo real](#) e um modelo de dados unificado para padronizar a integração do EHR.

Isso significa que não há necessidade de se preocupar com aplicativos implementando diferentes versões do FHIR ou um conjunto diferente de APIs; para conectar-se a um EHR, você só precisa codificar uma vez para as APIs Integrate para suportar a troca contínua de informações de assistência médica entre plataformas de EHR, sistemas clínicos e administrativos.

Graças ao Integrate, os desenvolvedores de TI podem fornecer acesso em tempo real a dados clínicos e administrativos sem comprometer a segurança da PHI. Visite nosso [Centro de aprendizado de API](#) para saber mais.

---

## 7 - LEITURAS ADICIONAIS FHIR PARA DESENVOLVEDORES DE TICS

Deseja aprender mais sobre o FHIR? Os seguintes recursos fornecem informações úteis sobre o FHIR na prática, incluindo tutoriais e outros recursos para desenvolvedores de TICS que desejam implementar o FHIR:

- [Beyond One-Off Integrations: A Commercial, Substitutable, Reusable, Standards-Based, Electronic Health Record–Connected App](#)
- [HL7 FHIR for Developers](#)
- [Interoperability Proving Ground \(IPG\)](#)
- [API Resource Collection in Health \(ARCH\)](#)
- [SMART on FHIR Tutorials](#)
- [FHIR for Developers](#)
- [Terminologies: Critical, but often deferred in FHIR application development](#)
- [Playing with FHIR – TUT002 : Creating a PATIENT Resource](#)
- [Using FHIR to Simplify Healthcare Application Development](#)
- [Tutorial: Access FHIR API with Postman](#)
- [HAPI on FHIR 10.16.2018](#)
- [Building a FHIR IG](#)
- [FHIR Tutorial](#)

---

A [InterOpera](#) é uma empresa de consultoria e treinamentos em padrões HL7 que vem ajudando o mercado brasileiro e internacional a romperem os desafios da troca de informações em saúde de forma rápida e segura. Entre em contato com nossos especialistas e o ajudaremos.