

PROGRAMA
EDUCACIONAL
EM **SAÚDE
DIGITAL**
DA UNIVERSIDADE
FEDERAL DE GOIÁS

PÓS-GRADUAÇÃO LATO SENSU
EM **SAÚDE DIGITAL**

Integração com a Rede Nacional de Dados em Saúde - RNDS

Organizadores

Fábio Nogueira de Lucena

Plínio de Sá Leitão Júnior

Renata Dutra Braga

2ª EDIÇÃO

Cegraf UFG





Universidade Federal de Goiás

Reitora

Angelita Pereira de Lima

Vice-Reitor

Jesiel Freitas Carvalho

Diretora do Cegraf UFG

Maria Lucia Kons

Conselho Editorial da Coleção Programa Educacional em Saúde Digital

Ana Laura de Sene Amâncio Zara (IPTSP / Universidade Federal de Goiás)

Fábio Nogueira de Lucena (INF / Universidade Federal de Goiás)

Gabriella Nunes Neves (CGISD / DATASUS / Secretaria Executiva / Ministério da Saúde)

Merched Cheheb de Oliveira (DATASUS / Secretaria Executiva / Ministério da Saúde)

Juliana Pereira de Souza Zinader (CGISD / DATASUS / Secretaria Executiva / Ministério da Saúde)

Maria Cristina Ferreira de Abreu (CGISD / DATASUS / Secretaria Executiva / Ministério da Saúde)

Rejane Faria Ribeiro-Rotta (FO / Universidade Federal de Goiás)

Renata Dutra Braga (INF / Universidade Federal de Goiás)

Rita Goreti Amaral (FF / Universidade Federal de Goiás)

Sheila Mara Pedrosa (CGIS / Universidade Federal de Goiás)

Silvana de Lima Vieira dos Santos (FEN / Universidade Federal de Goiás)

Taciana Novo Kudo (INF / Universidade Federal de Goiás)

Thais Lucena de Oliveira (CGISD / DATASUS / Secretaria Executiva / Ministério da Saúde)

Equipe de Produção

Amanda Souza Vitor - graduanda (UFG)

Dandra Alves de Souza - graduanda (UFG)

Gabriela Martins de Souza - graduanda (UFG)

Iêza Dara Costa Portela - graduada (UFG)

Iuri Vaz Miranda - graduando (UFG)

Jéssica Borges de Carvalho - técnica-administrativa (UFG)

Joyce Beatriz Ferreira da Costa Silva - graduanda (UFG)

Layane Grazielle Souza Dias - graduanda (UFG)

Luciana Dantas Soares Alves - analista de TI

Luma Wanderley de Oliveira - doutoranda (UFG)

Patrícia Galúcio Coqueiro Galvão - técnica-administrativa (UFG)

Virgínia de Fernandes Souza - graduanda (UFG)

Suse Barbosa Castilho - mestranda (UFG)

Comissão de Governança da Informação em Saúde (CGIS)

Silvana de Lima Vieira dos Santos

Centro de Inovação em Gestão da Educação e do Trabalho em Saúde (CIGETS) e Laboratório de Pesquisa em Empreendedorismo e Inovação (LAPEI)

Cândido Vieira Borges Júnior

Laboratório de Inovação e Estratégia em Governo (LineGov)

Antônio Isidro da Silva Filho

Ministério da Saúde / Secretaria Executiva / Departamento de Informática do Sistema Único de Saúde (DATASUS)

Merched Cheheb de Oliveira

Coordenação-Geral de Inovação e Informática em Saúde (CGIIS)

Adriano Santiago Dias dos Santos

Allan Nuno Alves de Sousa

André Gustavo Souza dos Santos

Andréia Cristina de Souza Santos

Blanda Helena de Mello

Elivan Silva Souza

Gabriella Nunes Neves

Josélio Emar de Araújo Queiroz

João Marquês Lopes Barbosa

Juliana Pereira de Souza Zinader

Juliana de Souza Santana

Kauara Ferreira

Kelly Neves Pinheiro Brito

Laís Bié Pinto Bandeira

Laíse Figueiredo Rolo de Oliveira

Lara Liz Freire

Larissa Gonçalves Mangabeira da Silva

Lucas da Costa Roriz

Maria Cristina Ferreira de Abreu

Patrícia dos Santos Irigaray Rodrigues

Robson Willian de Melo Matos

Rodrigo André Cuevas Gaete

Silmara Vieira da Silva

Thais Lucena de Oliveira

Vanessa Lora

Vinicius Colonese Mrad

Vitor Rocha de Araújo

Vlândia Barreira Beserra

Esta obra é disponibilizada nos termos da Licença Creative Commons – Atribuição – Não Comercial – Compartilhamento pela mesma licença 4.0 Internacional. É permitida a reprodução parcial ou total desta obra, desde que citada a fonte



Integração com a Rede Nacional de Dados em Saúde - RNDS

Organizadores

Fábio Nogueira de Lucena

Plínio de Sá Leitão Júnior

Renata Dutra Braga

Cegraf UFG

2022

© Cegraf UFG, 2022

© Fábio Nogueira de Lucena; Plínio de Sá Leitão Júnior; Renata Dutra Braga, 2022

© Universidade Federal de Goiás, 2022

© Ministério da Saúde, 2022

Revisão editorial

Ana Laura Sene Amâncio Zara

Revisão técnica

Blanda Helena de Mello

Laíse Figueiredo Rolo de Oliveira

Rodrigo André Cuevas Gaete

Capa

Iuri Vaz Miranda - graduando (UFG)

Editoração Eletrônica

Luma Wanderley de Oliveira - doutoranda (UFG)

Joyce Beatriz Ferreira da Costa Silva - graduanda (UFG)

1ª edição em 2021, pelo Cegraf UFG, ISBN: 978-65-89504-84-9,

DOI: <https://doi.org/10.5216/INT.ebook.978-65-89504-84-9.2021>

<https://doi.org/10.5216/INT.ebook.978-85-495-0573-6/2022>

Dados Internacionais de Catalogação na Publicação (CIP)
GPT/BC/UFG

161 Integração com a Rede Nacional de Dados em Saúde RNDS [Ebook] / organizadores, Fábio Nogueira de Lucena, Plínio de Sá Leitão Júnior; Renata Dutra Braga. - Dados eletrônicos (1 arquivo:PDF). 2. ed. - Goiânia : Cegraf UFG, 2022.

Inclui referências.

ISBN: 978-85-495-0573-6

1. Interoperabilidade em saúde. 2. Padrões. 3. Integração de sistemas. I. Lucena, Fábio Nogueira de. II. Leitão Júnior, Plínio de Sá. III. Braga, Renata Dutra. IV. Programa Educacional em Saúde Digital da Universidade Federal de Goiás. V. Pós-Graduação Lato Sensu em Saúde Digital.

CDU: 614:004.41

Bibliotecária responsável: Rosemarilany Barbosa Guida / CRB1: 3165

Integração com a Rede Nacional de Dados em Saúde - RNDS

Instituição responsável

Universidade Federal de Goiás (UFG)

Comissão de Governança da Informação em Saúde da UFG (CGIS-UFG)

Centro de Inovação em Gestão da Educação e do Trabalho em Saúde (CIGETS)

Laboratório de Pesquisa em Empreendedorismo e Inovação da Universidade Federal de Goiás (LAPEI-UFG)

Instituição financiadora

Ministério da Saúde (MS)

Secretaria Executiva (SE)

Departamento de Informática do Sistema Único de Saúde (DATASUS)

Secretaria de Gestão do Trabalho e da Educação na Saúde (SGTES)

Apoio

Ministério da Saúde (MS)

Secretaria de Atenção Primária à Saúde (SAPS)

Demais parceiros

Laboratório de Inovação e Estratégia em Governo (LineGov)



MINISTÉRIO DA SAÚDE



Abreviaturas e Siglas

ABNT	Associação Brasileira de Normas Técnicas
API	<i>Application Programming Interface</i>
Auth	<i>Authentication</i>
AWS	<i>Amazon Web Services</i>
CNES	Cadastro Nacional dos Estabelecimentos de Saúde
CNS	Cartão Nacional de Saúde
CQL	<i>Clinical Quality Language</i>
CRBIO	Conselho Regional de Biologia
CRBM	Conselho Regional de Biomedicina
CRF	Conselho Regional de Farmácia
CRM	Conselho Regional de Medicina
CRQ	Conselho Regional de Química
Datasus	Departamento de Informática do Sistema Único de Saúde
DDD	<i>Domain-Driven Design</i>
e-CNPJ	Cadastro Nacional de Pessoa Jurídica (digital)
e-CPF	Cadastro de Pessoa Física (digital)
EHR	<i>Electronic Health Record</i>
ESD28	Estratégia de Saúde Digital para o Brasil
FHIR	<i>Fast Healthcare Interoperability Resources</i>
GAL	Gerenciador de Ambiente Laboratorial
GM	Gabinete Ministerial
HL7	<i>Health Level Seven International</i>
HTML	<i>HyperText Markup Language</i>
HTTP	<i>HyperText Transfer Protocol</i>
HTTPS	<i>HyperText Transfer Protocol Secure</i>
ICP	Infraestrutura de Chaves Públicas
JDK	<i>Java Development Kit</i>
JSON	<i>JavaScript Object Notation</i>
JWT	<i>JSON Web Tokens</i>
LOINC	<i>Logical Observation Identifiers, Names, and Codes</i>
MS	Ministério da Saúde
NBR	Norma Brasileira

PNIS	Política Nacional de Informação e Informática em Saúde
RAC	Registro de Atendimento Clínico
RDF	<i>Resource Description Framework</i>
RDM	Registro de Dispensação de Medicamentos
REL	Resultado de Exame Laboratorial
RIA-C	Registro de Imunobiológico Administrado - Campanha
RNDS	Rede Nacional de Dados em Saúde
RPM	Registro de Prescrição de Medicamentos
SA	Sumário de Alta
SARS-CoV-2	Coronavírus 2 da Síndrome Respiratória Aguda Grave
SGHEM	Código para designar “sangue”
SIS	Sistema de Informação em Saúde
SNS	<i>Simple Notification Service</i>
SUS	Sistema Único de Saúde
SWEBOK	<i>Software Engineering Body of Knowledge</i>
TI	Tecnologia da Informação
TIC	Tecnologia da Informação e Comunicação
UFG	Universidade Federal de Goiás
UML	<i>Unified Modeling Language</i>
UNA-SUS	Universidade Aberta do SUS
URL	<i>Uniform Resource Locator</i>
XML	<i>eXtensible Markup Language</i>

Lista de Figuras, Tabelas e Videoaulas

Figura 1. Percurso sugerido para aprendizado: integração com a Rede Nacional de Dados em Saúde	18
Figura 2. Integração com a Rede Nacional de Dados em Saúde usando o padrão FHIR	21
Figura 3. Fluxo de informações de um laboratório para a Rede Nacional de Dados em Saúde após a promulgação da Portaria nº. 1.792, de 17 de julho de 2020	22
Figura 4. Envio de notificação de resultados de exames pelo laboratório à Rede Nacional de dados em saúde por meio de um software de integração (conector)	22
Figura 5. Papéis, funções e atribuições técnicas dos atores na Rede Nacional de Dados em Saúde	23
Figura 6. Infraestrutura técnica da Rede Nacional de Dados em Saúde	24
Figura 7. Uso de padrões para troca de informações com a Rede Nacional de Dados em Saúde	35
Figura 8. Modelo brasileiro de interoperabilidade em saúde	37
Figura 9. Modelo de informação e modelo computacional para registro de dados na Rede Nacional de Dados em Saúde	38
Figura 10. Relações entre as Portarias no contexto da informação em saúde	43
Figura 11. Função do integrador responsável pela criação do software que se conecta à Rede Nacional de Dados em Saúde	46
Figura 12. Sistema de informação em saúde	49
Figura 13. Modelo de Sistema de Informação em Saúde não conectado à Rede Nacional de Dados em Saúde	50
Figura 14. Modelo de Sistema de Informação em Saúde conectado à Rede Nacional de Dados em Saúde	51
Figura 15. Possível percurso de adoção da Rede Nacional de Dados em Saúde	53
Figura 16. Notificação de exame de COVID-19 na perspectiva de processos e do fluxo de informações entre os requisitos	54
Figura 17. Sequência típica de execução dos processos do Conector	54
Figura 18. Interações entre um estabelecimento de saúde e a Rede Nacional de Dados em Saúde registradas por meio de casos de uso	55
Figura 19. Interações com a Rede Nacional de Dados em Saúde	55
Figura 20. Bounded contexts e relação entre o sistema de informação em saúde (Upstream) e o Conector (Downstream)	59
Figura 21. Decisão estratégica representada por diagrama de componentes	59
Figura 22. Esboço inicial de classes de acordo com os casos de uso identificados	60
Figura 23. Bibliotecas com classes que promovem a reutilização de código	60
Figura 24. Esboço resultante	61
Figura 25. Interface gráfica disponibilizada pelo Portal de Serviços da Rede Nacional de Dados em Saúde	69

Figura 26. Snippet para compartilhar e validar recursos FHIR	70
Figura 27. Salvar e validar recurso	71
Figura 28. Resultado de validação de recurso	71
Figura 29. Colaboradores nos estabelecimentos de saúde	77
Figura 30. Processo de credenciamento	78
Figura 31. Solicitar acesso no Portal de Serviços	79
Figura 32. Página de destino após solicitar acesso no Portal de Serviços	80
Figura 33. Origem da solicitação no Portal de Serviços	80
Figura 34. Recorte da definição do recurso Capability Statement	84
Figura 35. Abstração para um Bundle (“envelope” ou “pacote”) no qual é depositado um resultado de exame laboratorial	90
Figura 36. Local onde o gestor pode localizar o identificador do solicitante no Portal de Serviços	91
Figura 37. Modelo de objetos Unified Modeling Language (UML) para um resultado de exame laboratorial inclui três recursos, entre os quais há duas referências	93
Figura 38. Componentes do protocolo HTTP cujo uso está bem definido para a conformidade com o padrão FHIR.	101
Figura 39. Pré-requisitos para interação com a Rede Nacional de Dados em Saúde	104
Figura 40. Configuração do importar collection	105
Figura 41. Requisições disponíveis	106
Figura 42. Indicação do certificado digital a ser utilizado pelo <i>Postman</i>	106
Figura 43. Configurar <i>Postman</i> para usar o certificado	107
Figura 44. Requisição para submissão do serviço “Obter token de acesso”	107
Figura 45. Variáveis específicas para esta coleta e requisição	108
Figura 46. Variável <i>auth</i> indica o endereço do web service empregado para autenticação	108
Figura 47. Variáveis geradas pelo próprio <i>Postman</i>	109
Figura 48. Requisição após a submissão com destaque para a URL, os dois headers de segurança e o retorno, código HTTP 200, que representa sucesso e a resposta em JSON	109
Tabela 1. Endereços de ambiente de homologação no Datasus	25
Tabela 2. Endereços do ambiente de produção no Datasus	25
Tabela 3. Modelo de informação do Resultado de Exame Laboratorial COVID-19	87
Videoaula 1. Introdução sobre a Rede Nacional de Dados em Saúde	20
Videoaula 2. Integração com a Rede Nacional de Dados em Saúde - introdução	21
Videoaula 3. Contexto	32
Videoaula 4. Regulamentação	40
Videoaula 5. Integrador	45
Videoaula 6. Conector	48
Videoaula 7. Ambiente de desenvolvimento	66
Videoaula 8. Processo de credenciamento	76

Videoaula 9. Entradas e saídas

[82](#)

Videoaula 10. FHIR API

[100](#)

Sumário

Unidade 1 - Apresentação	19
Rede Nacional de Dados em Saúde	20
O que muda com a RNDS?	22
Atores (foco no integrador)	23
Os ambientes	24
Organização	27
Sugestão de aprendizado	27
Unidade 2 - Contexto	30
O quê mais?	33
Padrão é imprescindível	35
FHIR é a opção nacional	36
Juntando saúde, papéis, tarefas, FHIR, ...	36
Como registrar o que flui de/para a RNDS?	38
Unidade 3 - Regulamentação	39
Criação da RNDS	40
Notificação obrigatória de COVID	41
Marco histórico	41
Resultado de exame laboratorial	41
Registro de vacinação contra COVID	42
Integração dos resultados de exames da COVID-19 com a RNDS	42
Registro de Prescrição de Medicamentos e Registro de Dispensação de Medicamentos	42
Registro de Atendimento Clínico (RAC)	42
Relações relevantes entre portarias	43
Unidade 4 - Integrador	44
Papel	45
Habilidades	45
Insumos	46
Macroatividades	46

	47
Unidade 5 - Conector	49
Sistema de Informação em Saúde (SIS)	49
Sem integração com a RNDS	51
Com integração com a RNDS	53
Modelo de conector	54
Requisitos	59
Projeto (design)	64
Implementação	
	65
Unidade 6 - Ambiente de Desenvolvimento	66
Obter token	67
Recurso FHIR no formato JSON	67
Validar recurso FHIR	67
Adaptação nacional (definições)	68
Recurso válido (e outro não)	69
Como validar um recurso?	71
FHIR Resource Editor (FRED)	71
Servidor FHIR	72
Gerar informação em saúde (para teste)	73
FHIR	73
Forge (editor de perfis)	73
Simplifier.NET	73
JSON (há ainda XML e RDF)	74
FHIRPath	74
Bibliotecas	74
Keytool	
	75
Unidade 7 - Processo de Credenciamento	76
Público-alvo	78
Passo a passo	
	81
Unidade 8 - Entradas e Saídas	82
Capacidade (Capability Statement)	85
Autenticação	86
Informações em saúde	86
Resultado de exame laboratorial	88
Modelo computacional	

Unidade 9 - FHIR API

- Requisição https para servidor FHIR
- Como me ambientar com esses detalhes?
- O que devo saber sobre a API?
- Pré-requisitos
- Segurança (*headers*)
- Passos
- Código que submete requisições
- Cumprimentos feitos

[99](#)

[101](#)

[103](#)

[104](#)

[104](#)

[105](#)

[105](#)

[110](#)

[110](#)

[111](#)

Unidade 10 - Unidade de Encerramento

[113](#)

Referências



Introdução

Prezado(a) Participante,

Seja bem-vindo(a) ao Microcurso **Integração com a Rede Nacional de Dados em Saúde (RNDS)**!

Este Microcurso faz parte do Programa Educacional em Saúde Digital da Universidade Federal de Goiás (UFG), criado em 2020, por meio da colaboração entre a UFG (Comissão de Governança da Informação em Saúde; Centro de Inovação em Gestão da Educação e do Trabalho em Saúde; Laboratório de Pesquisa em Empreendedorismo e Inovação), a Universidade Aberta do SUS – **UNA-SUS**; e o **Ministério da Saúde** (Secretaria de Gestão do Trabalho e da Educação na Saúde, Secretaria de Atenção Primária à Saúde e o Departamento de Informática do Sistema Único de Saúde do Brasil – Datasus).

Todos esses e outros agentes têm despendido esforços para que a RNDS seja implementada e que também tenha profissionais aptos a colaborarem nessa implementação. Um dos pontos importantes para que isso aconteça é o compartilhamento de informações entre os estabelecimentos de saúde com a RNDS, possibilitando que estes possam contribuir com informações em saúde pertinentes aos usuários que assistem, bem como consumir informações geradas por outros estabelecimentos, a qualquer instante, em qualquer lugar do Brasil permitindo, assim, a interoperabilidade em saúde.

O material produzido inclui o conhecimento necessário acerca das especificidades de integração de um Sistema de Informação em Saúde (SIS) com a RNDS, priorizando a formação de um agente fundamental para essa integração, aqui denominado integrador.

Para a confecção deste ebook, buscamos utilizar uma linguagem clara e direta, acompanhada de ilustrações para melhor entendimento, baseada na microaprendizagem.

Você, integrador, venha ampliar os seus conhecimentos e contribuir com a implementação da RNDS!

Excelente estudo!



Integração

É A VEZ DA SAÚDE

O mundo está integrado pela internet, e chegou a hora do Brasil conectar os atores da saúde. A solução tecnológica para tal, a RNDS, já está disponível!



INTENÇÃO

Interoperabilidade de informação e saúde visando a continuidade do cuidado do usuário, a qualquer instante, em qualquer lugar do Brasil.



Com a RNDS

A RNDS É FUNDAMENTAL



**MAS É
SUFICIENTE?**

Cada estabelecimento de saúde cria o *software* por meio do qual a informação em saúde é enviada para a RNDS. Este documento identifica e aborda tópicos pertinentes à tal interação.

Habilitar integradores a criar um *software* que interaja com a RNDS.



OBJETIVO

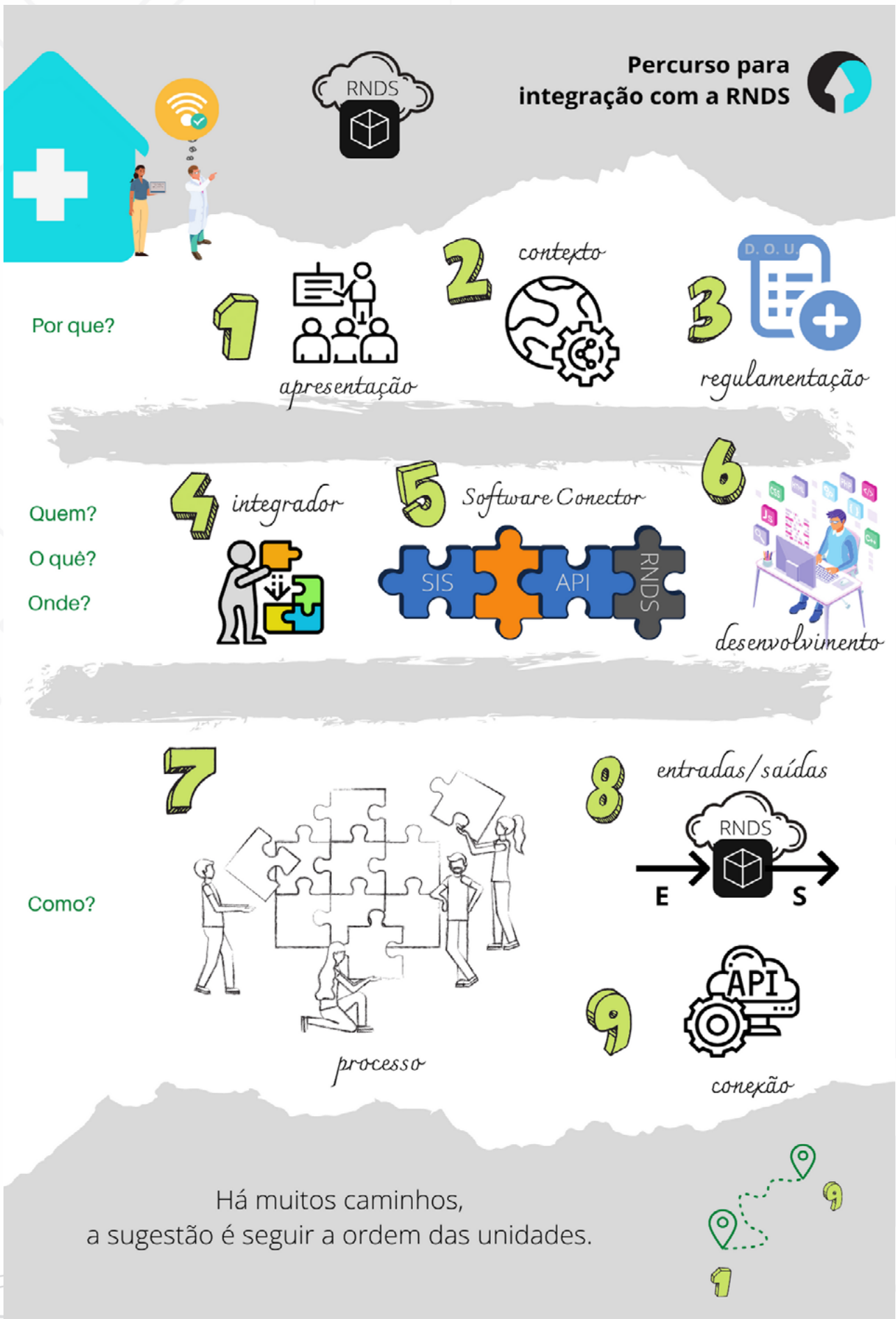


EXPECTATIVA

Se você é desenvolvedor e trabalha na saúde, ou em empresa que mantém um Sistema de Informação em Saúde (SIS), então está ou estará envolvido em um projeto de integração com a RNDS, este material foi preparado com foco na sua capacitação para tal tipo de projeto.

✨ ✨ ✨ Bom proveito!

Figura 1. Percurso sugerido para aprendizado: integração com a Rede Nacional de Dados em Saúde



METAS

estar apto a ...

A sugestão é seguir o conteúdo das nove unidades na ordem em que são apresentadas. Esta ordem contempla um integrador que desconhece a RNDS, discute questões pertinentes à Engenharia de Software e, finalmente, apresenta o contrato de integração com a RNDS.



Motivação (unidades 1,2 e 3)

- Conhecer a organização do material.
- Conhecer os pré-requisitos e o que se espera como resultado.
- Conhecer a existência de documentos como a Política Nacional de Informação e Informática em Saúde.
- Conhecer conceitos básicos de interoperabilidade.
- Compreender papéis relevantes no contexto da RNDS.



Fundamentos (unidades 4, 5 e 6)

- Conhecer as funções atribuídas ao integrador.
- Compreender o modelo de software de integração.
- Conhecer o ambiente típico de desenvolvimento.



Contrato (unidades 7, 8 e 9)

- Compreender detalhes do processo de credenciamento.
- Conhecer e compreender os serviços oferecidos pela RNDS.
- Localizar informações sobre o FHIR.
- Localizar informações sobre os perfis nacionais.
- Identificar restrições impostas pelos perfis nacionais.
- Montar requisições para os serviços oferecidos.

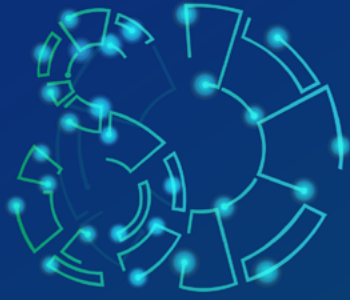
Fonte: autoria própria.



Direitos autorais

HL7 e FHIR são marcas registradas da *Health Level Seven International*. Estas marcas são usadas neste documento e este uso não significa que a HL7 aprova ou endossa este conteúdo. Confira [aqui](#) a política de uso destas marcas.





EDUCAÇÃO E CAPACITAÇÃO
DE RECURSOS HUMANOS
EM **SAÚDE DIGITAL**

Integração com a Rede Nacional de Dados em Saúde - RNDS

Unidade 1 **Apresentação do microcurso**

Fábio Nogueira de Lucena
Renata Dutra Braga
Silvana de Lima Vieira dos Santos



Apresentação

É verdade, a saúde depende dos profissionais de saúde, mas esse senso comum também é uma simplificação. Não inclui o ator principal, o usuário (paciente), nem a cooperação necessária de muitos outros, dentre os quais estão os desenvolvedores de *software*, aqui denominados de integradores, que produzem código para conexão com a RNDS.

RNDS é a sigla para Rede Nacional de Dados em Saúde.

Este material tem como foco contribuir com a formação de integradores aptos a desenvolver *software* para conexão com a RNDS. Esta é uma necessidade nacional. Este material inclui o que é necessário e suficiente para a integração com a RNDS, na perspectiva de um integrador, em uma linguagem clara e direta, acompanhada de ilustrações.

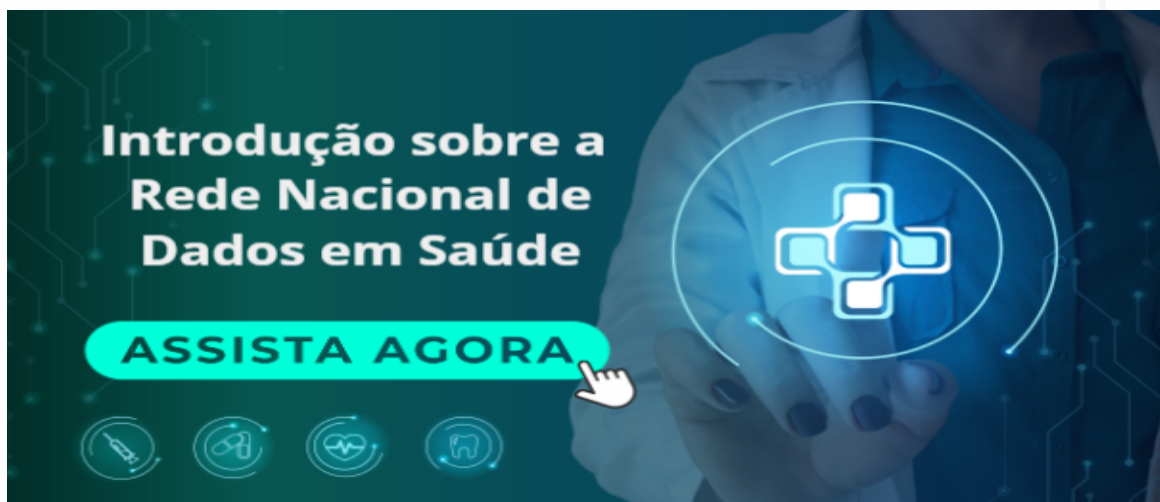
O papel do integrador, dentre outros relevantes para a RNDS, é identificado na Figura 11 (Unidade 4), assim como a expectativa de conhecimento acerca da saúde, Tecnologia da Informação (TI) e FHIR.

FHIR (Fast Healthcare Interoperability Resources) é o padrão adotado para troca de informação em saúde com a RNDS.

Rede Nacional de Dados em Saúde

Na perspectiva de um estabelecimento de saúde, a RNDS oferece serviços para a interoperabilidade de informações em saúde no território nacional. É por meio da RNDS que um estabelecimento de saúde disponibiliza a informação que será consumida por outro, em benefício do usuário (paciente), a qualquer instante, em qualquer lugar do Brasil (Videoaula 1).

Videoaula 1: Introdução sobre a Rede Nacional de Dados em Saúde



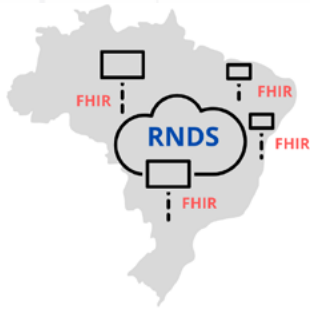
Fonte: autoria própria.



Quando um estabelecimento de saúde se integra à RNDS, cria-se a possibilidade de contribuir com informações em saúde pertinentes aos usuários que são assistidos pela mesma, bem como consumir informações geradas por outros estabelecimentos.

Conforme ilustrado abaixo, a integração com a RNDS segue o padrão FHIR, o que significa permitir a independência do estabelecimento de saúde em relação à tecnologia e ao seu ecossistema. Noutras palavras, a RNDS não impõe, não restringe e tampouco orienta decisões do estabelecimento de saúde.

Figura 2. Integração com a RNDS usando o padrão FHIR



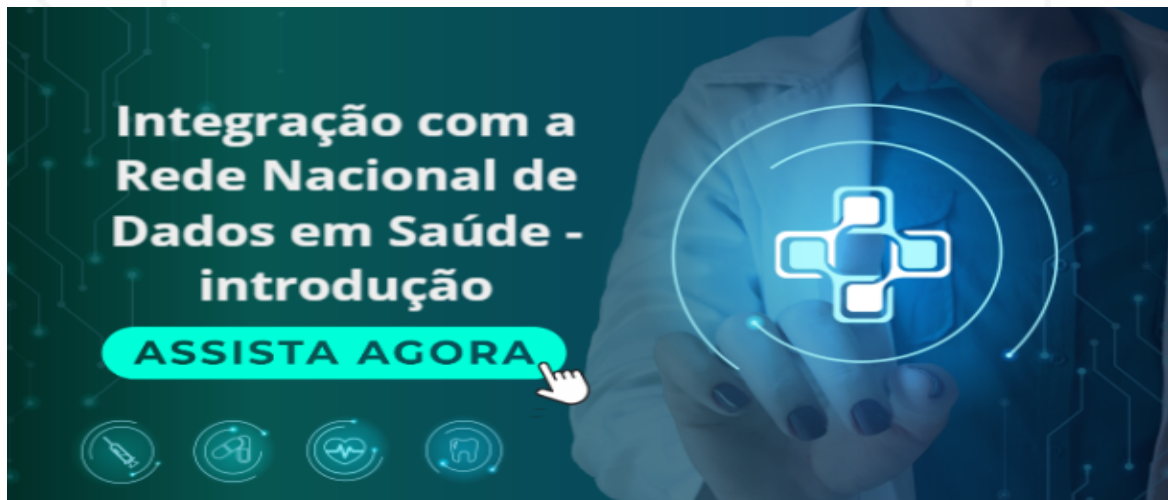
Página oficial da RNDS

<https://rnds.saude.gov.br/>

Fonte: autoria própria.

Nesse sentido, a interoperabilidade é realizada por meio de requisições *https*, que obedecem ao padrão FHIR e à adaptação definida pela RNDS, e partem da infraestrutura do estabelecimento de saúde para atingir o ambiente de produção da RNDS, seja para enviar ou requisitar informação em saúde (Videoaula 2).

Videoaula 2: Integração com a Rede Nacional de Dados em Saúde - introdução



Fonte: autoria própria.

Os serviços oferecidos pela RNDS para interoperabilidade serão estendidos, ao longo do tempo, para viabilizar as necessidades de troca de informação em saúde no Brasil. A primeira necessidade contemplada foi a notificação de resultados de exames laboratoriais de SARS-CoV-2.

Neste texto, o padrão FHIR é ilustrado para notificar um resultado de exame laboratorial conforme as especificidades nacionais estabelecidas pela RNDS, contudo, essa notificação é “similar” à submissão de um Registro de Atendimento Clínico (RAC) ou Sumário de Alta (SA), dentre outros.



O que muda com a RNDS?

Em um cenário convencional, sem a integração com a RNDS, laudos produzidos permanecem restritos ao sistema de *software* do laboratório em questão. Em algum momento são enviados para uma base de dados segura, e tornam-se inacessíveis a outros estabelecimentos de saúde, e infelizmente, até aos próprios pacientes e profissionais dispersos que os assistem.

A informação em saúde, contudo, “precisa chegar onde é necessária”. A [Portaria GM/MS nº 1.792, de 17 de julho de 2020](#), do Ministério da Saúde, é um marco histórico nesse sentido. Essa Portaria estabelece a obrigatoriedade de notificar o Ministério da Saúde acerca dos resultados de testes de SARS-CoV-2 produzidos em todo o território nacional.

Na Figura 3 é ilustrado o envio de laudos de um laboratório para a RNDS, como consequência dessa Portaria.

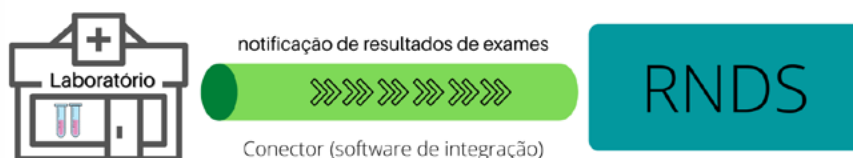
Figura 3. Fluxo de informações de um laboratório para a Rede Nacional de Dados em Saúde após a promulgação da Portaria nº. 1.792, de 17 de julho de 2020



Fonte: autoria própria.

Na prática, isso significa que cada laboratório terá que viabilizar a sua integração com a RNDS por meio do desenvolvimento de um *software* (Conector). Noutras palavras, esse *software* de integração do laboratório (estabelecimento de saúde), denominado de Conector, deve enviar a informação em saúde pertinente de cada laudo que produz ao Ministério da Saúde (veja a Figura 4).

Figura 4. Envio de notificação de resultados de exames pelo laboratório à Rede Nacional de dados em saúde por meio de um *software* de integração (conector)



Fonte: autoria própria.

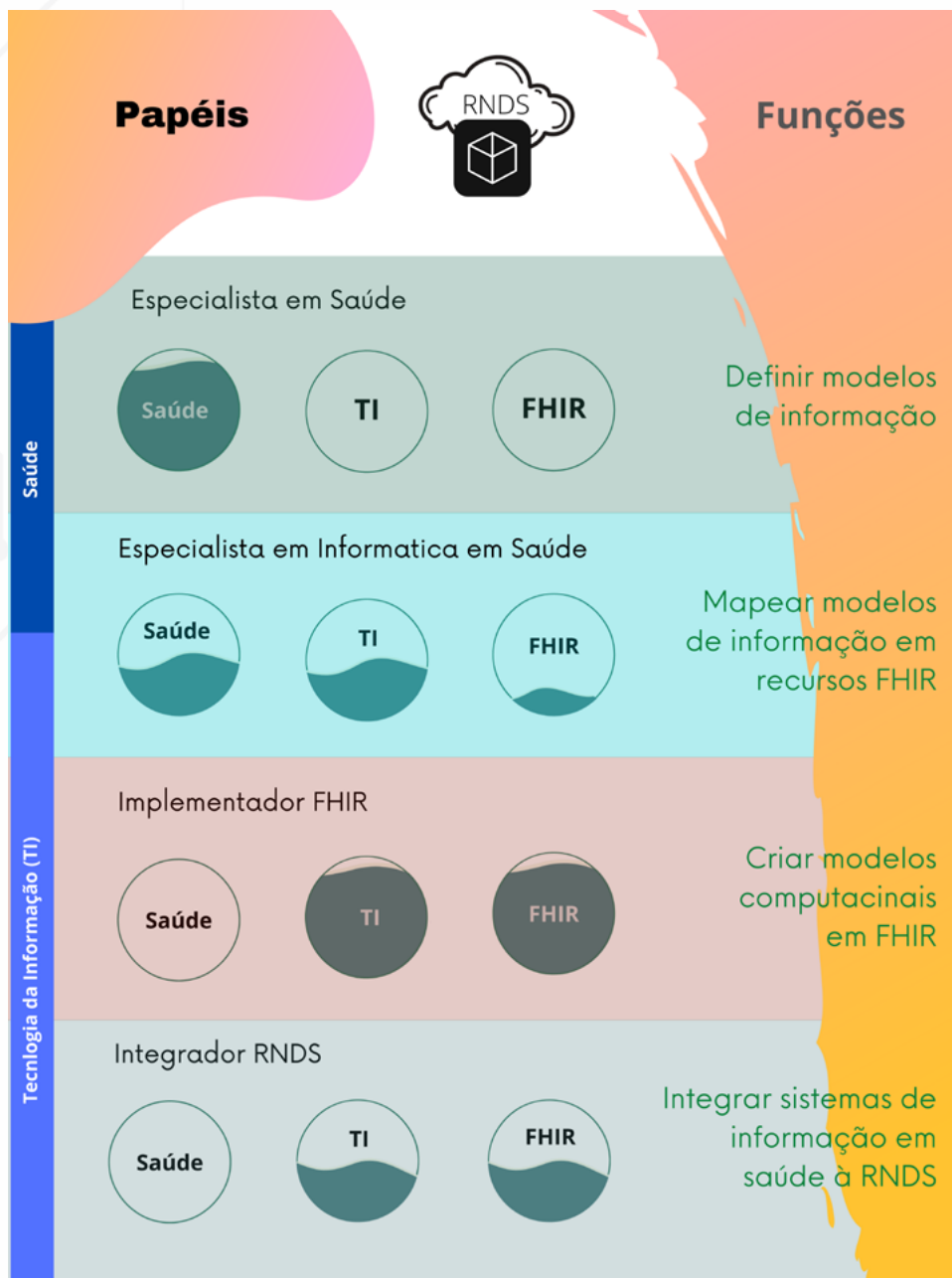


Atores (foco no integrador)

O principal ator da saúde é o cidadão (paciente), contudo, a atuação de vários outros é necessária. Dentre eles está o integrador, que é peça-chave no aspecto da integração de um estabelecimento de saúde com a RNDS. Todo este texto direciona as atenções para a formação do integrador. Consulte detalhes em “Integrador” (pág. 45).

Dentre os atores há: especialista em saúde, especialista em informática em saúde, implementador FHIR e integrador. Este texto não tem a pretensão de definir esses papéis, apenas faz uso de termos “geralmente” empregados no contexto, mesmo que de forma vaga. De qualquer forma, permite localizar o perfil do integrador e compará-lo com o de outros, conforme ilustrado na Figura 5.

Figura 5. Papéis, funções e atribuições técnicas dos atores na Rede Nacional de Dados em Saúde



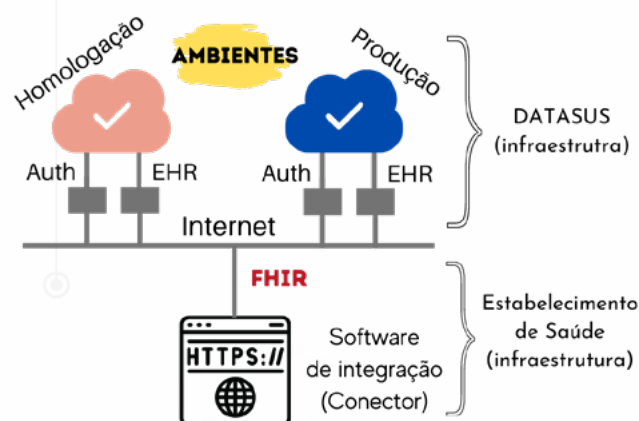
Gestor de estabelecimento de saúde
(sem atribuição técnica de saúde, TI e FHIR)

Fonte: autoria própria.

Ambientes

Pode-se observar a RNDS pela infraestrutura técnica oferecida pelo Datasus. Tal infraestrutura está encapsulada em dois ambientes. Há um ambiente de homologação e outro de produção. Esses ambientes e o “software Conector” (pág. 48), integrados pela adoção do padrão FHIR, são ilustrados na Figura 6.

Figura 6. Infraestrutura técnica da Rede Nacional de Dados em Saúde



Fonte: autoria própria.

A interoperabilidade em saúde no território nacional se realiza pelo acesso a estes ambientes, em particular, o ambiente de produção, por um *software* externo ao escopo do Ministério da Saúde, aqui denominado de Conector.

Qual ambiente usar?

Durante o desenvolvimento e/ou teste de *software* que se conecta à RNDS, o ambiente a ser utilizado é o ambiente de homologação.

E quando o *software* for considerado pronto para cumprir sua função de integração com a RNDS? Nesse caso, evidências deverão ser coletadas e submetidas pelo estabelecimento de saúde em questão para serem apreciadas pelo Datasus. Se a resposta é positiva, então o acesso ao ambiente de produção é disponibilizado ao estabelecimento, como detalhado no “Processo de Credenciamento” (pág. 76).

Após a aprovação da requisição de acesso ao ambiente de produção, parte do processo de credenciamento, o *software* conector do estabelecimento de saúde poderá acessá-lo para envio e recuperação de informação em saúde.

Auth e EHR (dois endereços)

Tanto o ambiente de homologação quanto o de produção oferecem dois endereços para acesso: um para finalidade de segurança, denominado *Auth*, e outro para acesso aos serviços de troca de informações em saúde, denominado de *EHR*. Não há nada de especial nesses nomes, apenas são referências a *Authentication* e *Electronic Health Record*, respectivamente.

O endereço *Auth* é exclusivo para obtenção do *token* de acesso (código de acesso). O uso desse endereço exige o emprego de certificado digital.

O *token* de acesso é exigido como parte de cada uma das requisições enviadas ao endereço *EHR*. Essa exigência é satisfeita por meio de *header* próprio, *X-Authorization-Server*. O valor desse *header*

assim como o conteúdo do outro *header* de segurança exigido, *Authorization*, são discutidos em detalhes em “Segurança (*headers*)” (pág. 105).

Ambiente de homologação (endereços)

O ambiente de homologação existe para testes e experimentações. Esse ambiente é único para todo o Brasil (Tabela 1).

Tabela 1. Endereços de ambiente de homologação no Datasus

Nome	Endereços do ambiente de homologação
Auth (único)	ehr-auth-hmg.saude.gov.br
EHR (único)	ehr-services.hmg.saude.gov.br

Fonte: autoria própria.

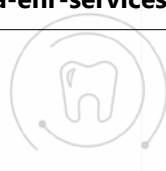
Ambiente de produção (endereços)

O ambiente de produção oferece os serviços que, de fato, permitem o envio e o recebimento de informações em saúde dos usuários espalhados por todo o território nacional.

Apesar do ambiente de produção oferecer um único endereço *Auth* para obtenção do *token* de acesso, o endereço *EHR* é definido por Unidade da Federação (UF). Ou seja, o endereço *EHR* a ser utilizado por um estabelecimento é identificado pela UF em que se localiza esse estabelecimento, conforme a Tabela 2.

Tabela 2. Endereços do ambiente de produção no Datasus

Nome/estado	Endereços do ambiente de produção
Auth (único)	ehr-auth.saude.gov.br
EHR (por UF)	Conforme a UF do estabelecimento de saúde
Acre	ac-ehr-services.saude.gov.br
Alagoas	al-ehr-services.saude.gov.br
Amapá	ap-ehr-services.saude.gov.br
Amazonas	am-ehr-services.saude.gov.br
Bahia	ba-ehr-services.saude.gov.br
Ceará	ce-ehr-services.saude.gov.br
Distrito Federal	df-ehr-services.saude.gov.br
Espírito Santo	es-ehr-services.saude.gov.br
Goiás	go-ehr-services.saude.gov.br
Maranhão	ma-ehr-services.saude.gov.br



Nome/estado	Endereços do ambiente de produção
Mato Grosso	mt-ehr-services.saude.gov.br
Mato Grosso do Sul	ms-ehr-services.saude.gov.br
Minas Gerais	mg-ehr-services.saude.gov.br
Pará	pa-ehr-services.saude.gov.br
Paraíba	pb-ehr-services.saude.gov.br
Paraná	pr-ehr-services.saude.gov.br
Pernambuco	pe-ehr-services.saude.gov.br
Piauí	pi-ehr-services.saude.gov.br
Rio de Janeiro	rj-ehr-services.saude.gov.br
Rio Grande do Norte	rn-ehr-services.saude.gov.br
Rio Grande do Sul	rs-ehr-services.saude.gov.br
Rondônia	ro-ehr-services.saude.gov.br
Roraima	rr-ehr-services.saude.gov.br
Santa Catarina	sc-ehr-services.saude.gov.br
São Paulo	sp-ehr-services.saude.gov.br
Sergipe	se-ehr-services.saude.gov.br
Tocantins	to-ehr-services.saude.gov.br

Fonte: autoria própria.

O endereço EHR a ser utilizado pelo Laboratório Rômulo Rocha, localizado no Estado de Goiás (CNES 2337991), por exemplo, é **go-ehr-services.saude.gov.br**. Em outro exemplo, o *software* do Hospital de Base de Brasília deve usar o endereço EHR **df-ehr-services.saude.gov.br**, e assim por diante.

Em tempo, o Cadastro Nacional de Estabelecimentos de Saúde (CNES) é público e qualquer cidadão pode localizar estabelecimentos, a localização deles e os profissionais de saúde que neles estão lotados, dentre outras informações. Basta apontar o seu navegador para o endereço abaixo. Ou seja, o próprio leitor pode fazer suas buscas e identificar o endereço EHR correto para o estabelecimento de interesse.

<http://cnes.datasus.gov.br/>

Acesse aqui a wiki do CNES



Organização

O conteúdo deste material está dividido em 10 Unidades.

As três primeiras apresentam o porquê da integração com a RNDS. Isso inclui, além da presente Apresentação, o “Contexto” (pág. 31), no qual SIS e a RNDS estão inseridos, bem como a “Regulamentação” (pág. 40), que institui legalmente a integração.

As três unidades seguintes estabelecem os elementos básicos da integração (fundamentos), respectivamente, caracterizam o perfil do “Integrador” (pág. 45), um modelo de *software* de integração, aqui denominado de “Conector” (pág. 48), e o “Ambiente de Desenvolvimento” (pág. 66), que apresenta uma lista de tecnologias à disposição do integrador. Neste texto, integrador é o perfil de desenvolvedor de *software* que cria o Conector.

Enquanto as unidades anteriores apresentam o contexto e os fundamentos, as três últimas especificam o “contrato” da integração, ou seja, como a integração se realiza em conformidade com as restrições existentes. Isso é feito ao elucidar o “Processo de Credenciamento” (pág. 76), “Entradas e Saídas” (pág. 82) e “FHIR API” (pág. 100).

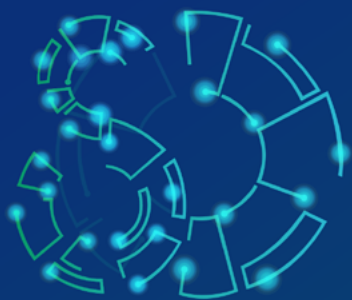
Por último, a “Unidade de Encerramento” (pág. 112) fornece orientações adicionais aos interessados.

Sugestão de aprendizado

Siga as unidades na ordem apresentada, elas tem o objetivo de dividir as várias questões que fazem parte da integração com a RNDS. A intenção é permitir um caminho para aquisição de conhecimento e habilidades.

As 10 unidades estão distribuídas em três grupos, denominados de **motivação** (compreender o contexto), **fundamentos** (entender as bases da integração) e **contrato** (observar as restrições). Essa classificação reúne as unidades sem perder a coerência de cada grupo e também contribui com a identificação de expectativas (metas) pertinentes ao conhecimento e habilidades atribuídos a cada grupo.





EDUCAÇÃO E CAPACITAÇÃO
DE RECURSOS HUMANOS
EM **SAÚDE DIGITAL**

Integração com a Rede Nacional de Dados em Saúde - RNDS

Unidade 2 **Contexto**

Fábio Nogueira de Lucena
Plínio de Sá Leitão Júnior



Contexto



“Temos de saber o que fomos e o que somos, para saber o que seremos.”

Paulo Freire

Paulo Freire, na citação acima, nos orienta a atuar no presente, alterar o *status quo*, para obter o que desejamos, tendo em vista que não há como mudar o passado. A Política e a Estratégia comentadas abaixo são exemplos concretos de esforços nacionais visando um futuro melhor para a saúde no Brasil.

A Política Nacional de Informação e Informática em Saúde (PNIIS) tem como finalidade definir os princípios e as diretrizes a serem observados pelas entidades públicas e privadas de saúde no âmbito do Sistema Único de Saúde (SUS) [3] e, desta forma, alinhar ações no território nacional.

A PNIIS foi atualizada recentemente pela Portaria GM/MS no. 1.768, de 30 de julho de 2021 e inclui um conjunto bem-definido de prioridades, dentre as sete prioridades, destacamos três: (a) informatização das instituições de saúde públicas e privadas; (b) ambiente de conectividade em saúde e (c) ecossistema de inovação. Todas as prioridades são igualmente relevantes, contudo, aquelas citadas estão diretamente associadas aos integradores. A primeira ressalta a dependência da infraestrutura disponível, a segunda o ambiente criado pela RNDS por meio do qual esta integração deve ocorrer e, por último, reitera o enorme potencial de inovação decorrente da integração.

Não é surpresa, portanto, que dentre os elementos norteadores da Política, o principal deles é a interoperabilidade dos SIS. Esta interoperabilidade tem sua origem no art. 47 da Lei nº 8.080/90, repetido no próprio texto da Política:

“O Ministério da Saúde, em articulação com os níveis estaduais e municipais do Sistema Único de Saúde (SUS) organizará, no prazo de 02 (dois) anos, um sistema nacional de informação em saúde, integrado em todo o território nacional, abrangendo questões epidemiológicas e de prestação de serviços”

A Estratégia de Saúde Digital para o Brasil [4], orientada pela PNIIS, que por sua vez está alinhada com a Lei 8.080/90, define como prioritária, dentre várias atividades a serem realizadas no período de 2020 a 2028, a criação de um “Ambiente de Interconectividade”. A interconectividade citada na Estratégia e a interoperabilidade citada na Política são dois termos alusivos à integração entre SIS, estabelecida pelo art. 47 da Lei nº 8.080/90 (Videoaula 3).

Videoaula 3: Contexto



Fonte: autoria própria.

A **Política Nacional de Informação e Informática em Saúde (PNIIS)** e a **Estratégia de Saúde Digital para o Brasil (ESD28)** orientam esforços no escopo de Tecnologia da Informação e Comunicação (TIC) de todo o sistema brasileiro de saúde.



Estratégia o percurso

RNDS é uma das prioridades.

RNDS potencializa o trabalho colaborativo entre atores da saúde.

O quê mais?

Na Estratégia de Saúde Digital para o Brasil [4], página 11, lê-se:

“... fazer com que todas as unidades de saúde se conectem à RNDS”.

A integração com a RNDS é meta fundamental da Estratégia, que é apoiada por outra prioridade da Estratégia, a capacitação de recursos humanos. Ou seja, o leitor deste texto contribui com essa prioridade por meio da sua própria formação.

A Estratégia inclui a conexão de todo estabelecimento de saúde com a RNDS, mas qual o objetivo da RNDS?

“promover a troca de informações entre os pontos da Rede de Atenção à Saúde (RAS), permitindo a transição e continuidade do cuidado nos setores público e privado”.

Dito de outra forma, há necessidade de troca de informações em saúde para viabilizar a transição e a continuidade do cuidado, entre e pelas unidades de saúde, sejam elas públicas ou privadas.





Em geral, dados dois Sistemas de Informação (SISs) quaisquer, não há fluxo de informações entre eles.

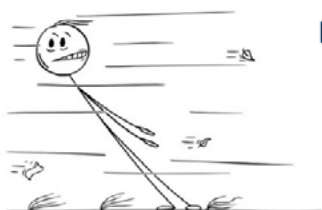
A informação flui pelo próprio usuário, seja pela memória, pedidos, laudos e outros que transporta, ocasionando inconvenientes entre a continuidade do cuidado.

A PNIIS orienta para a interoperabilidade, a ESD prioriza e a RNDS implementa.



REDE NACIONAL DE DADOS EM SAÚDE SOLUÇÃO

Estabelecimentos ligados pelas informações que geram e consomem.



DESAFIOS

A complexidade e o volume do tráfego de dados além do conjunto significativo de atores, explicam porque a interoperabilidade em saúde é difícil.



Padrão é imprescindível

A troca de informação em saúde depende da existência e adoção de padrões.

Padrões são amplamente empregados, e não apenas no domínio da saúde. Por exemplo, a extensão ou comprimento atribuído a um metro ou o tamanho de uma folha de papel A4 são estabelecidos por meio de padrões. No Brasil, não faz muito tempo que o formato de tomadas elétricas foi padronizado. A intenção é fazer com que aparelhos possam se conectar à rede elétrica nacional.

De forma análoga, para que um SIS, empregado por um estabelecimento de saúde, possa trocar informações em saúde com a RNDS, é preciso fazer uso de padrões. Em consequência, a Portaria GM/MS nº 1.434, de 28 de maio de 2020 do Ministério da Saúde presta serviço análogo ao da norma ABNT NBR 14136:2012, que define o padrão nacional de tomadas, conforme ilustrado abaixo.

Figura 7. Uso de padrões para troca de informações com a Rede Nacional de Dados em Saúde

ABNT NBR 14136:2012	Outros padrões...	
		

Portaria 1.434, de 28/05/2020	Esta portaria cria a RNDS e estabelece o sítio https://servicos-datusus.saude.gov.br para divulgação dos padrões nacionais de interoperabilidade.	
	Fonte: autoria própria.	

**Fast Healthcare Interoperability Resources, ou
FHIR, é a escolha nacional**



FHIR é a opção nacional

Talvez a mais proeminente proposta para troca de informações em saúde, no momento.

*E as informações
são grátis!*

<https://hl7.org/fhir/>

Juntando saúde, papéis, tarefas, FHIR,...

Uma perspectiva simplificada (modelo) da interoperabilidade adotada pelo Brasil é útil para compreender o papel do integrador.

Tudo começa com uma necessidade na saúde, caracterizada por especialistas em saúde. Essa necessidade é registrada por um modelo de informação, mapeado em recursos FHIR, ou “personalizações” (perfis) para atender às especificidades nacionais.

Depois, há quem implemente um servidor FHIR, naturalmente, em conformidade com os perfis adotados no Brasil, coloque à disposição os *web services* necessários, por meio do Ambiente de Produção, para atender a necessidade de troca de informação, e os mantenha em operação. Essa é uma simplificação da sustentação da RNDS, é o território do Datasus (MS).

Finalmente, entra em cena o integrador, que é o responsável por produzir o *software* de conexão com a RNDS. Esse *software* conector irá interagir com a RNDS por meio do Ambiente de Produção. A interação visa a enviar e receber informações em saúde. Em ambos os casos, é preciso saber exatamente o que e como enviar para que a informação possa ser interpretada corretamente no extremo oposto. FHIR é a “parafernália” tecnológica adotada para viabilizar a interação entre um SIS qualquer (devem existir milhares no Brasil) e a RNDS.

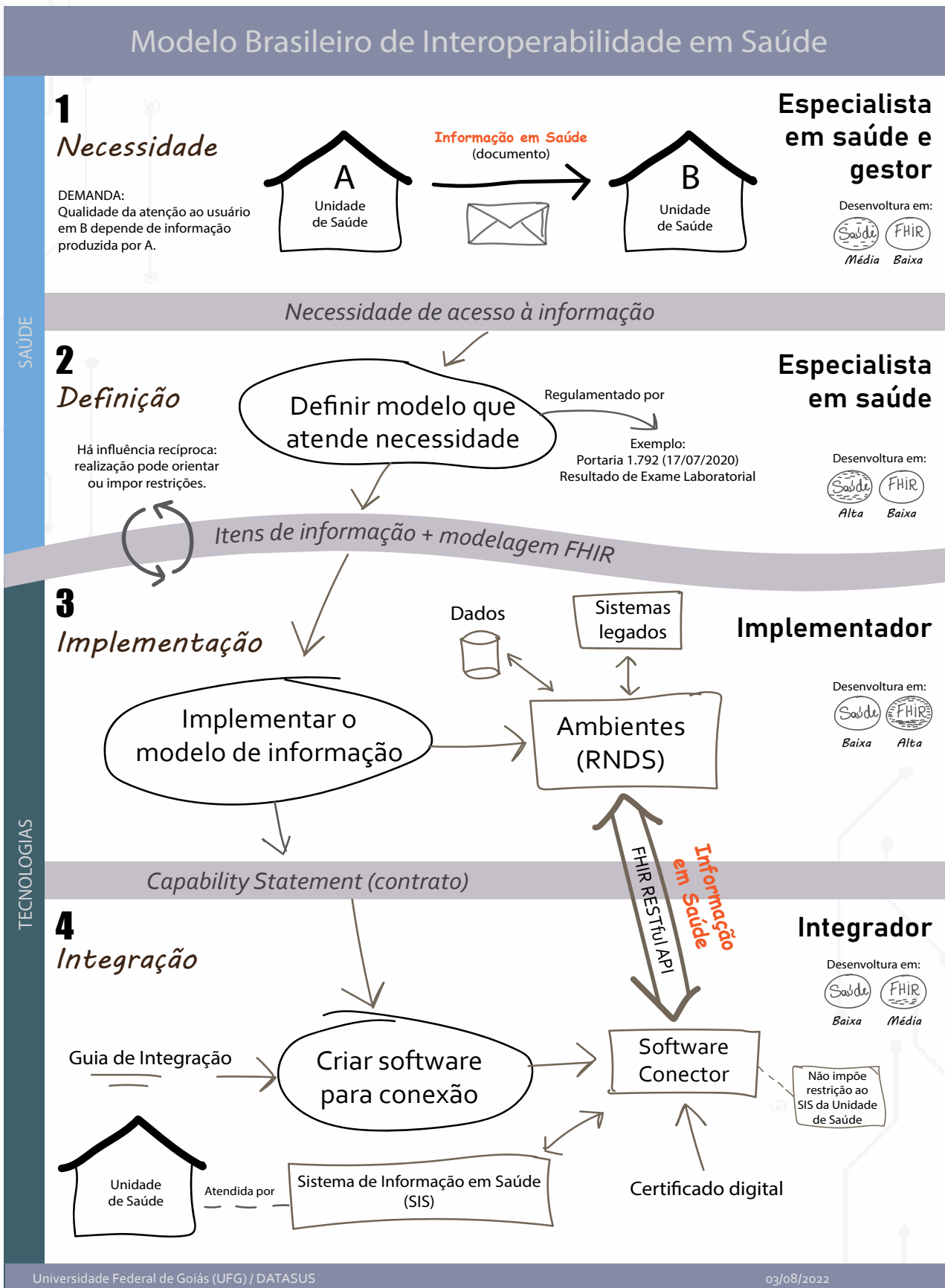
Quando o integrador desenvolve tal *software*, aqui chamado de Conector, o SIS correspondente pode gerar a informação que será enviada pelo Conector para a RNDS que, por sua vez, poderá fornecê-la a outro SIS, qualquer que seja esse, em execução em qualquer estabelecimento de saúde no Brasil. Naturalmente, com o consentimento do paciente e respeitando as exigências éticas e legais.

A integração tem a capacidade de contribuir com a qualidade da saúde no Brasil, não é verdade?

O foco aqui, contudo, é o integrador. Vamos continuar. Precisamos saber como construir parte da “ponte” que cabe a cada estabelecimento de saúde!

*Como tudo isso
se encaixa?*

Figura 8. Modelo brasileiro de interoperabilidade em saúde



Como registrar o que flui de/para a RNDS?

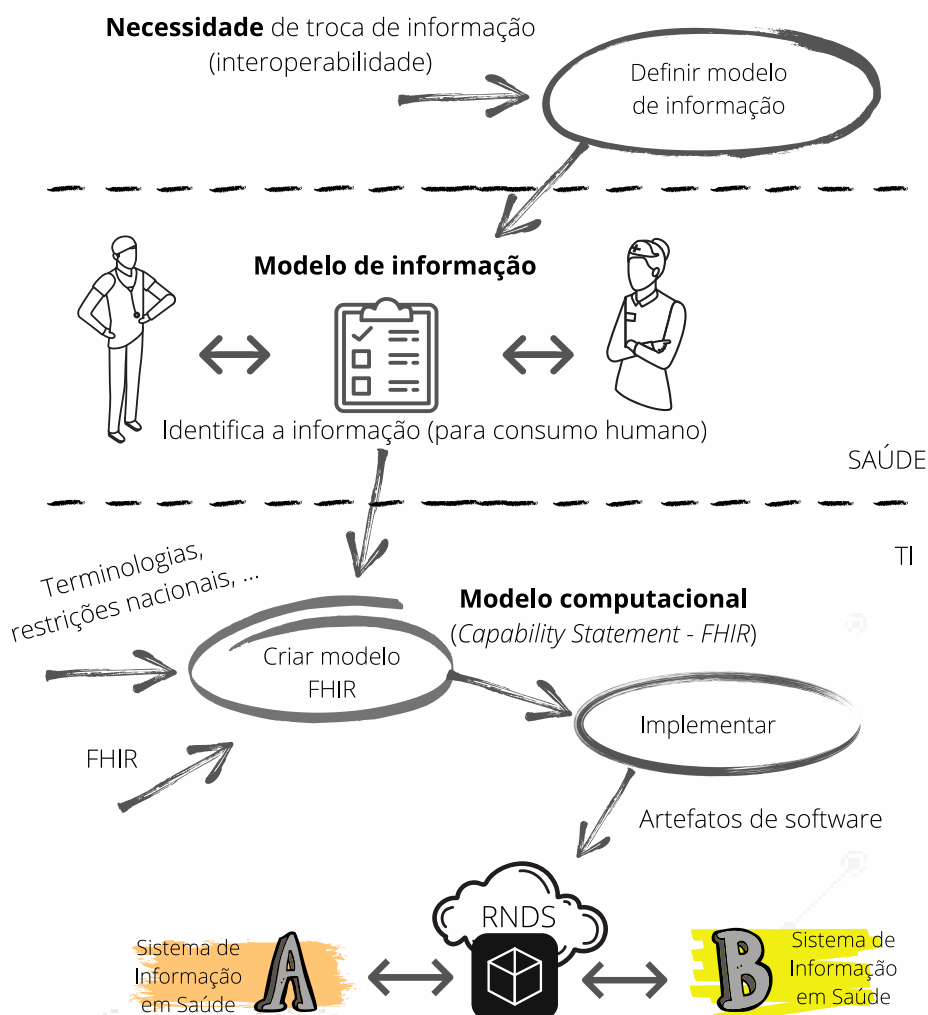
Afinal, estabelecimentos de saúde precisam saber exatamente o que se espera que enviem para a RNDS e, no sentido oposto, o que eles esperam receber quando recuperam informações da RNDS. Esse conhecimento é registrado e publicado por dois modelos complementares. Ambos são definidos para cada necessidade de troca de informação em saúde e, indistintamente, se a informação é enviada ou recebida.

O **modelo de informação** (ou informacional) define itens de informação que participam de uma troca de informação com a RNDS. Por exemplo, o registro de um Sumário de Alta é feito por um conjunto de itens, enquanto um Registro de Atendimento Clínico (RAC) possui o seu próprio conjunto de itens. Esse modelo fornece uma representação conceitual de tais itens, adequada para consumo (entendimento e uso) humano, juntamente com a definição semântica correspondente.

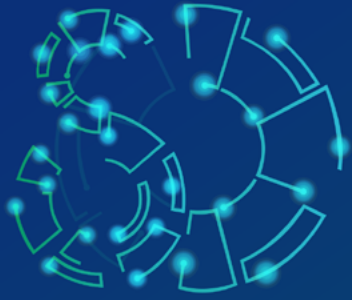
O **modelo computacional** é definido em função do modelo de informação correspondente. Ou seja, é a representação técnica do modelo de informação do qual foi derivado. O modelo computacional faz uso de uma linguagem computacional para a sua representação e está voltado para o consumo por computadores. O modelo computacional inclui os recursos FHIR e os perfis que ajustam tais recursos ao contexto nacional.

Modelo de informação e modelo computacional são equivalentes, mas produzidos para públicos distintos, respectivamente, ser humano e computador. A definição de informações em saúde trocadas com a RNDS empregam esses modelos, conforme “Entradas e Saídas” (pág. 82).

Figura 9. Modelo de informação e modelo computacional para registro de dados na Rede Nacional de Dados em Saúde



Fonte: autoria própria.



EDUCAÇÃO E CAPACITAÇÃO
DE RECURSOS HUMANOS
EM **SAÚDE DIGITAL**

Integração com a Rede Nacional de Dados em Saúde - RNDS

Unidade 3 **Regulamentação**

Fábio Nogueira de Lucena
Plínio de Sá Leitão Júnior
Renata Dutra Braga
Silvana de Lima Vieira dos Santos





Regulamentação

A integração com a RNDS é legal e formalmente estabelecida por meio de vários instrumentos, em particular, portarias. É bom que o estabelecimento de saúde, tanto gestor quanto integrador conheçam as exigências do Ministério da Saúde.

Dois destes instrumentos incluem a Política Nacional de Informação e Informática em Saúde (PNIIS) e a Estratégia de Saúde Digital para o Brasil (ESD28). Estes são basilares e foram discutidos ao apresentar o “Contexto” (pág. 25).

Naturalmente, muitos outros instrumentos poderiam ser citados além daqueles aqui fornecidos, contudo, prevaleceram os documentos mais recentes e próximos do escopo de interesse do integrador (Videoaula 4).

Videoaula 4: Regulamentação



Fonte: autoria própria.

Criação da RNDS

Está além do escopo apresentar a extensa história da gestação da RNDS, que culminou com sua criação formal pela Portaria GM/MS nº 1.434, de 28 de maio de 2020 [1]. Conforme o Art. 254-A. A respeito dessa Portaria, a RNDS

“... consiste em uma plataforma nacional voltada à integração e à interoperabilidade de informações em saúde entre estabelecimentos de saúde públicos e privados e órgãos de gestão em saúde dos entes federativos, para garantir o acesso à informação em saúde necessário à continuidade do cuidado do cidadão.”

Essa Portaria ainda estabelece o local de divulgação dos padrões de interoperabilidade, dentre outros, a serem adotados no território nacional:

¹ <https://www.in.gov.br/en/web/dou/-/portaria-n-1.434-de-28-de-maio-de-2020-259143327>

- Padrões de interoperabilidade...
<https://servicos-datasus.saude.gov.br/>
- Terminologias, ontologias, ...
<https://rts.saude.gov.br/>
- Governança, gestão...
<https://saudedigital.saude.gov.br/>

Notificação obrigatória de COVID-19

A Portaria GM/MS nº 1.792, de 17 de julho de 2020 [2] estabelece a obrigatoriedade de notificação ao Ministério da Saúde de todos os resultados de testes diagnóstico para SARS-CoV-2 realizados por laboratórios da rede pública, rede privada, universitários e quaisquer outros, em todo território nacional.

Marco histórico

A notificação obrigatória de resultado de exame de SARS-CoV-2 resulta na necessária integração entre todos os laboratórios em território nacional que realizam exames de SARS-CoV-2 e a RNDS.

Em uma perspectiva de *software*, isso significa a obrigação de desenvolvimento de *software* de integração entre SIS de laboratórios no território nacional e a RNDS.

Se preferir uma metáfora, ordena a criação de uma ponte entre estabelecimentos de saúde e o Ministério da Saúde, um projeto de infraestrutura, que potencializa inúmeros benefícios decorrentes do tráfego viabilizado de informações em saúde entre os estabelecimentos de saúde.

Talvez seja oportuno ressaltar que informações em saúde vão fluir entre estabelecimentos de saúde, mas não será pela inviável construção de uma ponte entre quaisquer dois estabelecimentos de saúde no Brasil. Se considerarmos apenas 10 estabelecimentos, para ilustrar, teríamos que construir 45 pontes. A RNDS viabiliza a troca de informações em saúde, entre 10 estabelecimentos, por meio da construção de apenas 10 pontes.

Dito de outra forma, em vez de se construir uma ponte entre A e B, quaisquer que sejam estes estabelecimentos, o que é impraticável, A constrói uma ponte até a RNDS, B constrói uma ponte até a RNDS, e agora, como resultado, A poderá consumir informação em saúde que foi gerada por B e vice-versa.

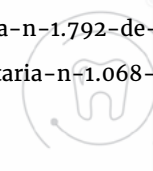
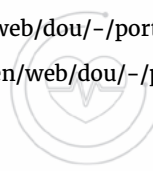
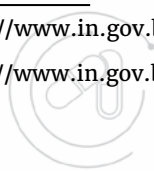
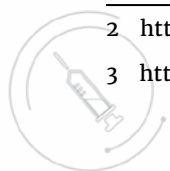
Não há oportunidade melhor, neste ponto, para destacar que a Portaria GM/MS nº 1.792, de 17 de julho de 2020, é um marco histórico, e que o objetivo do presente material é habilitar desenvolvedores a criarem pontes que conectam estabelecimentos à RNDS.

Resultado de exame laboratorial

Enquanto a Portaria comentada anteriormente estabelece a notificação compulsória, os detalhes da informação em saúde a ser trafegada pela ponte são definidos pela Portaria GM/MS nº 1.068, de 17 de novembro de 2020 [3] .

2 <https://www.in.gov.br/web/dou/-/portaria-n-1.792-de-17-de-julho-de-2020-267730859>

3 <https://www.in.gov.br/en/web/dou/-/portaria-n-1.068-de-17-de-novembro-de-2020-289283311>



Esta Portaria institui o modelo de informação de resultado de exame laboratorial para SARS-CoV-2. Também relaciona o modelo de informação com o modelo computacional. Ambos são produzidos para uma mesma informação em saúde a ser trocada, apenas são dirigidos a públicos distintos.

Tais modelos foram definidos em “Como registrar o que flui de/para a RNDS?” (pág. 38) e aqueles correspondentes para a notificação de exame para SARS-CoV-2 estão extensivamente documentados em “Entradas e Saídas” (pág. 82).

Registro de vacinação contra COVID-19

A Portaria GM/MS nº 69, de 14 de janeiro de 2021 [4] institui a obrigatoriedade do registro de vacinação contra a COVID-19 nos sistemas de informação do Ministério da Saúde. Essa Portaria orientou a definição do modelo de informação do Registro de Imunobiológico Administrado – Campanha (RIA-C).

Formalmente, esse é o segundo instrumento oficial que exige a plena operação da RNDS para que possa ser cumprido pelos estabelecimentos em questão.

Integração dos resultados de exames da COVID-19 com a RNDS

A Portaria GM/MS Nº 1.046, de 24 de maio de 2021, estabelece as regras para integração dos resultados de exames realizados para a detecção do SARS-CoV-2 por laboratórios da rede pública, rede privada, universitários e quaisquer outros, em todo território nacional na RNDS. Esses exames realizados, enviados para a RNDS, serão integrados de forma automática com o sistema de registro de notificações de casos de síndrome gripal suspeitos de COVID-19 (e-SUS Notifica). Exceto os laboratórios já integrados com a RNDS não são obrigados a realizar a notificação no e-SUS Notifica. Mais detalhes sobre as regras e procedimentos de como integrar com a RNDS, acesse <https://rnnds-guia.prod.saude.gov.br/>.

Registro de Prescrição de Medicamentos e Registro de Dispensação de Medicamentos

A Portaria SAES/MS nº 50, de 9 de fevereiro de 2022, institui os Modelos de Informação para Registro de Prescrição e de Dispensação de Medicamentos. Essa Portaria fornece uma introdução sobre a temática, objetivos, escopo, termos e definições relacionados ao assunto, assim como o cenário de uso e, por último, os elementos que compõem os modelos de informação para Registro de Prescrição de Medicamentos (RPM) e para o Registro de Dispensação de Medicamentos (RDM).

Registro de Atendimento Clínico (RAC)

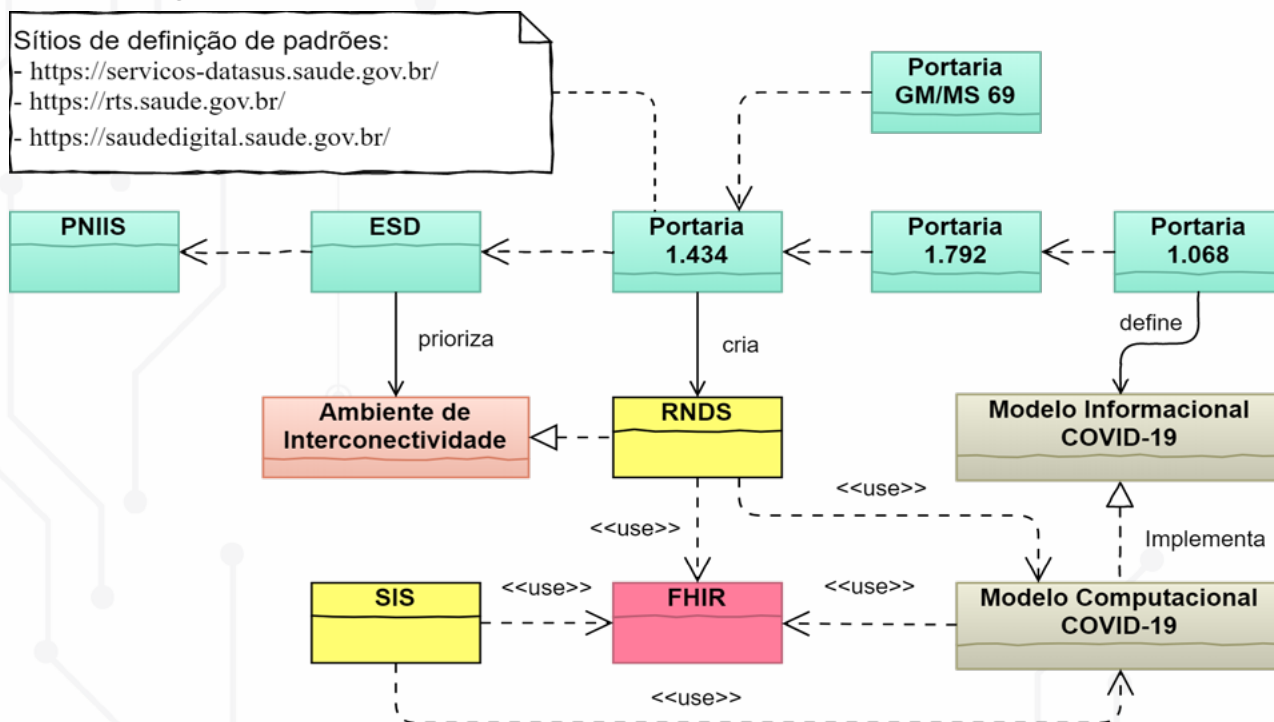
A Portaria SAES/MS Nº 234, de 18 de julho de 2022², institui o Modelo de Informação Registro de Atendimento Clínico (RAC). Essa Portaria apresenta uma introdução sobre o registro de atendimento clínico, objetivos deste modelo de informação, escopo, conjunto mínimo de dados, contexto que deve ser utilizado. Esse documento é primordial para apoiar nas trocas de informações de um atendimento clínico entre os diversos níveis de atenção; melhorar a qualidade da atenção em saúde e a segurança para o indivíduo com informações qualificadas; reduzir custos, dentre outros.

4 <https://www.in.gov.br/en/web/dou/-/portaria-gm/ms-n-69-de-14-de-janeiro-de-2021-299306102>

Relações relevantes entre Portarias

Documentos, Portarias, modelos, conceitos, o padrão FHIR e outros estão relacionados. Esta seção esclarece algumas das relações relevantes entre esses instrumentos, conforme a na Figura 10.

Figura 10. Relações entre as Portarias no contexto da informação em saúde



Fonte: autoria própria.

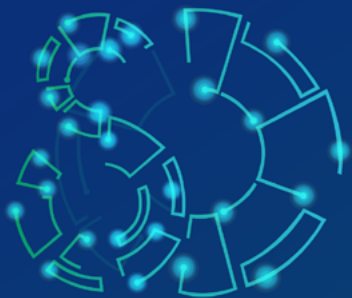
A política nacional (PNIIS) é o documento seminal que ordena todo o resto.

A ESD28 foi criada observando-se a PNIIS, ou seja, a ESD28 depende da PNIIS. A Portaria GM/MS nº 1.434, de 28 de maio de 2020 cria formalmente a RNDS, que é o instrumento de implementação do Ambiente de Interconectividade, definido pela ESD28, ao mesmo tempo em que é uma das prioridades da ESD28.

Uma vez criada a RNDS (Portaria GM/MS nº 1.434, de 28 de maio de 2020), a Portaria GM/MS nº 1.792, de 17 de julho de 2020 estabelece a obrigatoriedade de notificação de exames da COVID-19 por meio da RNDS. A Portaria GM/MS nº 1.068, de 17 de novembro de 2020, baseada no exame obrigatório definido pela Portaria GM/MS nº 1.792, de 17 de julho de 2020, define o modelo informacional para registro de resultado de exame de COVID-19. O modelo computacional, instrumento técnico, implementa o modelo informacional correspondente conforme o padrão FHIR. Observe que a comunicação entre um SIS e a RNDS dá-se por meio do FHIR observando-se o modelo computacional em questão.

Em tempo, FHIR e outros padrões são identificados formalmente pelos sítios oficiais definidos pela Portaria GM/MS nº 1.434, de 28 de maio de 2020. Daí a dependência da Portaria GM/MS nº 1.792, de 17 de julho de 2020, comentada acima e da Portaria nº 69, que institui a obrigatoriedade do registro da vacinação contra a COVID-19 nos sistemas de informação do Ministério da Saúde, leia-se, integração com a RNDS, para essa finalidade. ^[1]

1 <https://www.in.gov.br/en/web/dou/-/portaria-saes/ms-n-50-de-9-de-fevereiro-de-2022-379573004>



EDUCAÇÃO E CAPACITAÇÃO
DE RECURSOS HUMANOS
EM **SAÚDE DIGITAL**

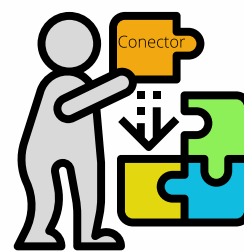
Integração com a Rede Nacional de Dados em Saúde - RNDS

Unidade 4 **Integrador**

Fábio Nogueira de Lucena
Plínio de Sá Leitão Júnior



Integrador



Papel

O integrador desempenha a função de criar o *software* que conecta um SIS à RNDS. Por simplicidade, este *software* é denominado de Conector, veja “Conector” (pág. 48).

A função do Integrador também pode ser realizada por um grupo de pessoas ou uma empresa que produz *software* (Videoaula 5).

Videoaula 5: Integrador



Fonte: autoria própria.

Habilidades

Quem é integrador deve possuir habilidades técnicas no desenvolvimento e manutenção de *software*. Essa formação pode ser obtida em um curso tecnológico de programação ou um curso superior em Engenharia de *Software*, dentre outros.

Tecnicamente é preciso fluência em linguagem de programação. O texto faz referências a duas delas, as linguagens Java e JavaScript. Noções de projeto (*design*) de *software*, orientação a objetos e a *Unified Modeling Language* (UML) são úteis para a compreensão de vários diagramas apresentados. Também é desejável desenvoltura na manipulação de documentos em JSON, facilidade para instalação de programas e uso de aplicativos, via linha de comandos, e integração de *software* via REST API.



Insumos

A função do integrador no contexto da RNDS depende dos seguintes insumos: (a) Guia de Integração (o material deste curso inclui o conteúdo desse Guia); (b) o SIS da instituição em questão e (c) a descrição detalhada dos serviços para acesso à RNDS (*Capability Statement* – FHIR). Informações pertinentes sobre todos esses itens estão espalhadas pelo presente material.

Macroatividades

Invariavelmente o integrador terá que:

1. Analisar a funcionalidade necessária para a integração. Isso inclui, dentre outras, identificar os dados a serem enviados, criar o documento JSON correspondente e enviá-lo. Observe que essas atividades dependem do SIS em questão.
2. Realizar o projeto (*design*) do Conector (*software* de integração).
3. Implementar o conector.

Todas essas atividades são consideradas em detalhes ao apresentar um modelo de “Conector” (pág. 48) e, de certa forma, todo este material contribui com tais atividades.

Figura 11. Função do integrador responsável pela criação do *software* que se conecta à Rede Nacional de Dados em Saúde

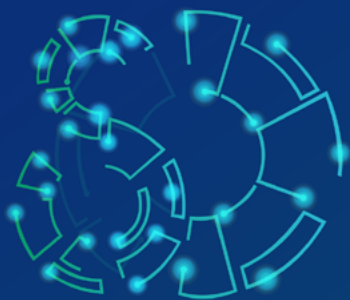
A FUNÇÃO DO INTEGRADOR É CRIAR O SOFTWARE QUE SE CONECTE À RNDS

Exige habilidades técnicas de **DESIGN & CONSTRUÇÃO**



Fonte: autoria própria.





EDUCAÇÃO E CAPACITAÇÃO
DE RECURSOS HUMANOS
EM **SAÚDE DIGITAL**

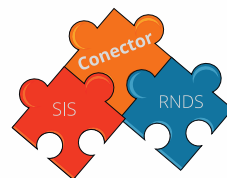
Integração com a Rede Nacional de Dados em Saúde - RNDS

Unidade 5 Software de Integração (Conector)

Fábio Nogueira de Lucena
Plínio de Sá Leitão Júnior



Conector



Conector é o nome aqui dado à solução tecnológica que acessa a RNDS. É esse software que precisa ser desenvolvido pelo “Integrador” (pág. 45) e, de fato, o que implementa a troca de informação em saúde com a RNDS. A troca pode ser observada na perspectiva dos dados, “Entradas e Saídas” (pág. 82) e da tecnologia empregada para envio/recebimento propriamente dito, ou seja, a “FHIR API” (pág. 100).

Ao interagir com a RNDS, um estabelecimento de saúde indiretamente viabiliza a interação com todos os demais estabelecimentos. Por exemplo, no contexto de um laboratório, faz parte das funções do *software* Conector enviar o resultado de um exame laboratorial para a RNDS, o que torna possível o acesso a esse resultado por outros estabelecimentos de saúde que estejam conectados à RNDS. Ou seja, em vez de se construir inúmeras conexões entre os estabelecimentos de saúde, é construída, uma única conexão entre cada estabelecimento e a RNDS.

As seções seguintes apresentam questões típicas da integração com a RNDS por meio de um Conector de referência, abstrato o suficiente para que possa servir de orientação para vários estabelecimentos de saúde interessados em detalhes técnicos da integração com a RNDS. Ou seja, um Conector “genérico” é empregado para discussão útil ao desenvolvimento de um Conector específico, compatível com o cenário do estabelecimento em questão (Videoaula 6).

Videoaula 6: Conector



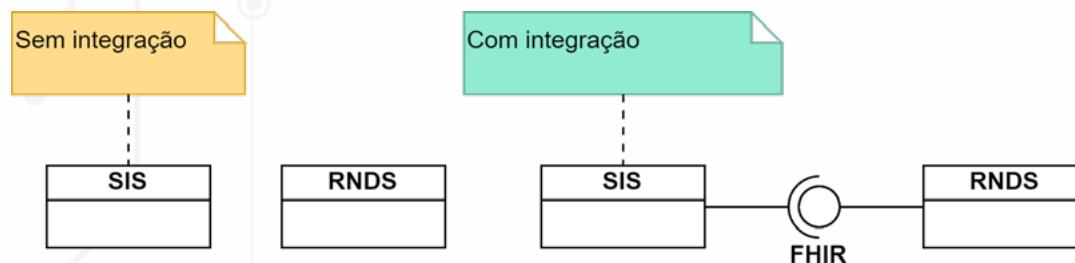
Fonte: autoria própria.



Sistema de Informação em Saúde (SIS)

Um estabelecimento de saúde usa, em geral, um SIS para auxiliar na gestão das suas demandas, usuários, profissionais de saúde, procedimentos e outros. Tal sistema pode estar integrado à RNDS ou não.

Figura 12. Sistema de informação em saúde



Fonte: autoria própria.

Adicionalmente, pode-se representar abstratamente qualquer SIS por dois componentes: (a) um Banco de Dados, no qual informações de saúde e outras administrativas são armazenadas; e (b) o *software* propriamente dito do SIS.

Essa organização abstrata de um SIS em apenas dois componentes auxilia na identificação do que é denominado de Conector, ou Conector de Referência, ou ainda, Modelo de Conector.

Esta caracterização é feita abaixo por meio dos dois cenários ilustrados acima.

Sem integração com a RNDS

Quando não há interoperabilidade do SIS com a RNDS, não há *software* Conector e, conseqüentemente, não há integração com a RNDS. Nesse cenário, as informações em saúde produzidas no estabelecimento de saúde podem ser registradas e geridas por um SIS, mas para uso “exclusivo” do estabelecimento, pois permanecem “confinadas” nas fronteiras desse estabelecimento.

Naturalmente, a informação pode cruzar as fronteiras do estabelecimento, mas geralmente será impressa em papel, que se deteriora com o tempo, se perde, exige transporte físico, ...

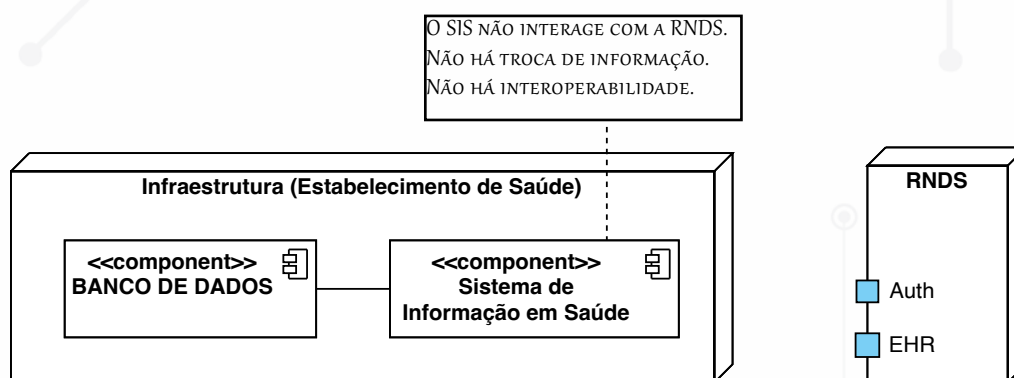


Figura 13. Modelo de Sistema de Informação em Saúde não conectado à Rede Nacional de Dados em Saúde

Estabelecimentos que empregam um SIS **sem** conexão com a RNDS não contribuem nem se beneficiam de informações em saúde do usuário que assiste.



Modelo de SIS (não conectado à RNDS)



Fonte: autoria própria.

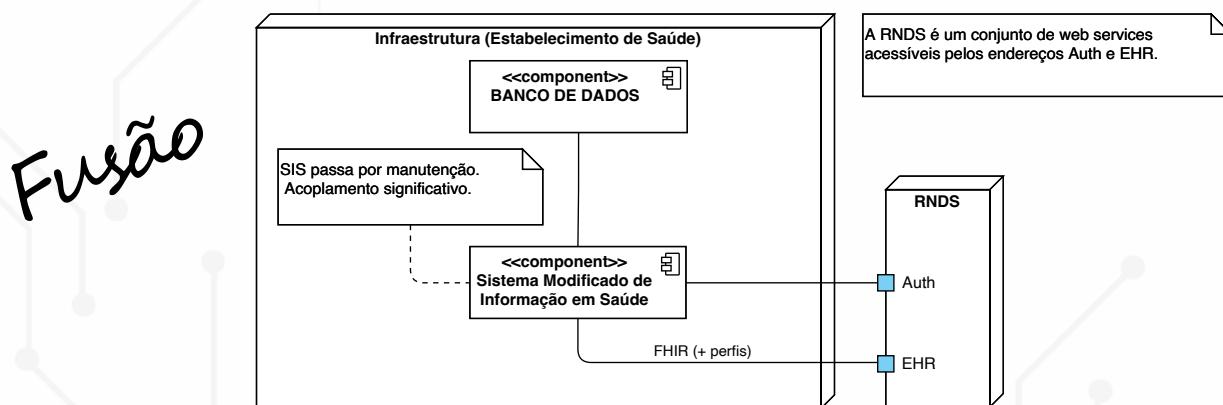


Com integração com a RNDS

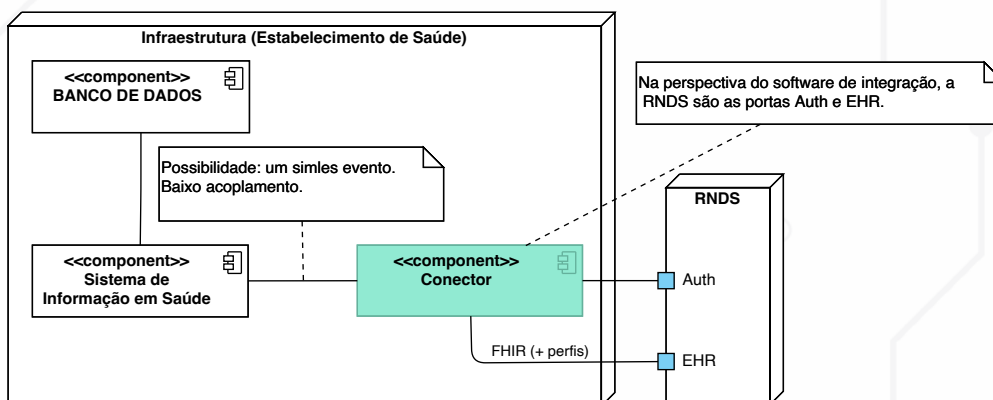
Para iniciar essa relação do Conector com o SIS, pode-se assumir várias formas, mas seguindo duas estratégias: fusão ou separação. A seguir ambas são exemplificadas na Figura 14.

Na primeira, o SIS passa por uma manutenção na qual o código do Conector é “mesclado” ao código existente. Por mais isolado que esteja esse código, há acoplamento considerável entre SIS e Conector. Na segunda, um componente específico reúne e isola o código do Conector e do SIS.

Figura 14. Modelo de Sistema de Informação em Saúde conectado à Rede Nacional de Dados em Saúde



Separação



Fonte: autoria própria.

Qual estratégia?

Cabe ao integrador considerar o contexto e decidir.

NÃO é recomendação!



A integração com a RNDS não impõe restrições ao projeto (*design*) do SIS, nem às tecnologias empregadas pelo estabelecimento de saúde

O foco é a **formação do integrador**.

O desenvolvimento do Conector cabe ao estabelecimento de saúde.

A linguagem de programação é escolha do estabelecimento de saúde.

A plataforma .Net, Java ou outra, é opção do estabelecimento de saúde.

A manutenção do SIS cabe ao estabelecimento de saúde.

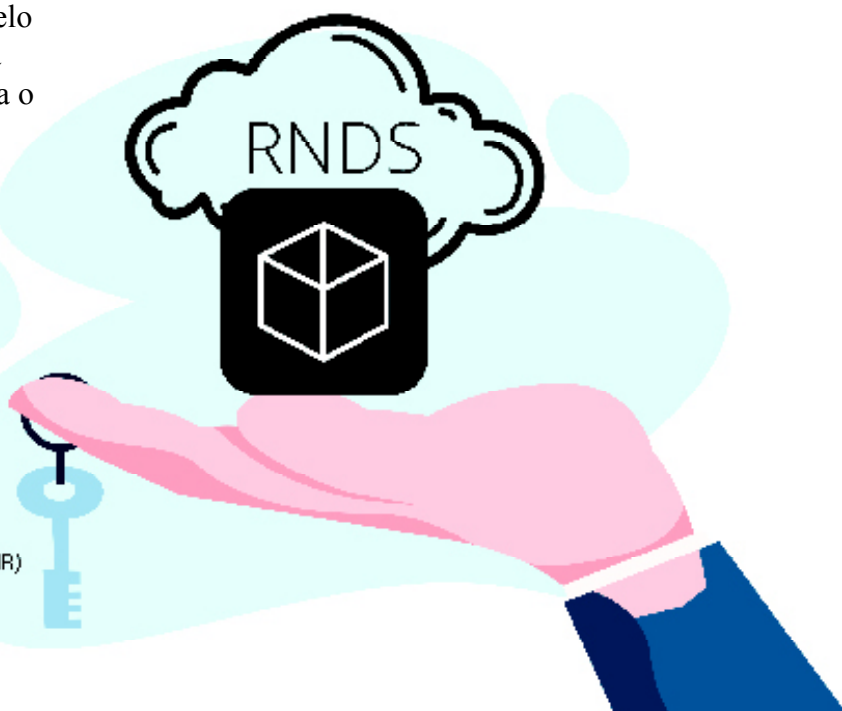
A RNDS não sugere nem recomenda tecnologia específica usada pelo SIS.

Contrato

A RNDS especifica a informação em saúde que flui e como fazê-la fluir entre SIS e os *web services* que oferece para tal.

O registro da especificação é feito por meio de Portarias, que empregam o modelo de informação e o modelo computacional. O primeiro é uma descrição textual, o segundo adota o padrão FHIR.

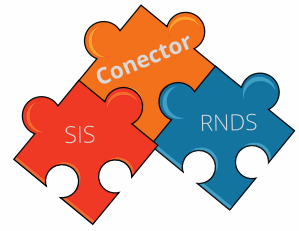
Contrato
Modelo de informação
Modelo computacional (FHIR)



Modelo de Conector

Tendo em vista as especificidades de cada estabelecimento de saúde, não é factível definir uma análise e um design de um Conector adequado a todos eles.

Objetivo: oferecer uma orientação para integradores. Estratégia: familiarizar o integrador com questões naturais da integração com a RNDS.



Escopo

O Conector, como qualquer outro *software*, visa a atender alguma demanda. Para o esforço de familiarização a demanda selecionada (escopo) é a Portaria GM/MS nº 1.792, de 17 de julho de 2020. Essa Portaria, legalmente, determina a obrigatoriedade de notificação de resultados de exame da SARS-CoV-2. Convém observar que a notificação de laudo de COVID-19 é a primeira necessidade de troca de informação contemplada pela RNDS.

Ao longo do tempo, outras necessidades serão incluídas, como a notificação do Sumário de Alta (SA) e do Registro de Atendimento Clínico (RAC), por exemplo. Cada necessidade é detalhada tanto pelo modelo de informação quanto pelo modelo computacional, o que viabiliza a integração com a RNDS para a necessidade em questão.

Naturalmente, à medida que novas necessidades de integração forem implementadas pela RNDS, mais informações em saúde e outros tipos de estabelecimentos de saúde estarão envolvidos.

A notificação de laudos de COVID-19 é uma necessidade específica, mas permanece relevante para outras demandas.

Figura 15. Possível percurso de adoção da Rede Nacional de Dados em Saúde

Possível percurso de adoção da RNDS (fevereiro/2021)



Resultado de Exame Laboratorial

Portaria 1.792, de 17/07/2020



Fonte: autoria própria.



Requisitos

Os requisitos são apresentados em duas perspectivas: uma de fluxo de dados e outra de casos de uso.

Fluxo de dados

A notificação de exame de SARS-CoV-2 é ilustrada na Figura 16, na perspectiva de processos e do fluxo de informações entre eles. Os coloridos são “comuns”, enquanto os demais são específicos por SIS.

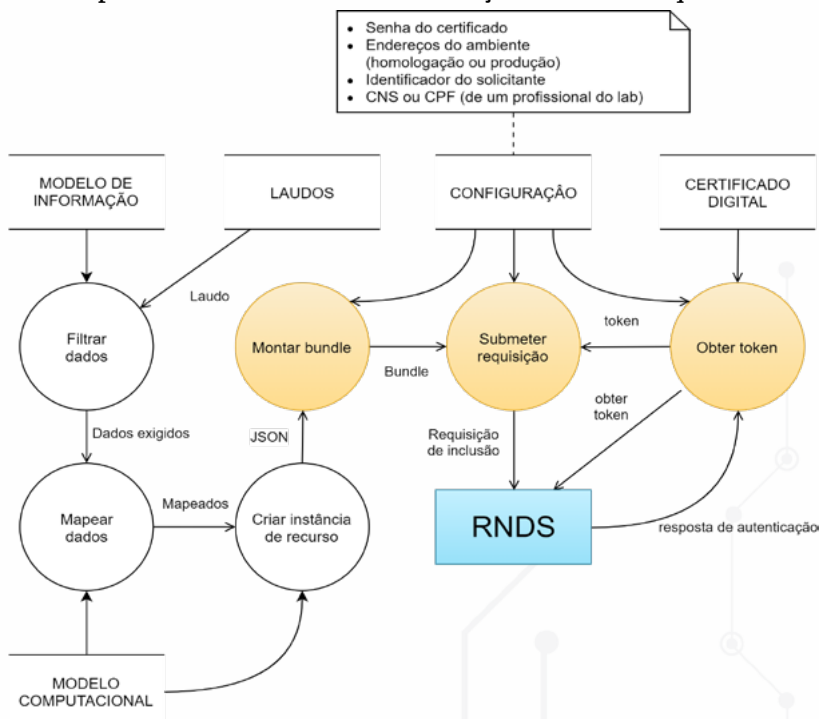
Cada uma dessas funções é comentada abaixo. Dependendo do SIS, contudo, algumas delas podem não ser necessárias. Por exemplo, se um dado SIS já guarda informações sobre cada resultado de exame em documento XML próprio, então não será necessário coletar informações dispersas, filtrar e talvez tampouco efetuar algum mapeamento.

Filtrar dados. Seleciona os dados de um resultado de exame contendo o que é necessário para o envio. Consome informações registradas no depósito de dados do laboratório.

Mapear dados. Realiza a conversão e/ou mapeamento, se for o caso, entre o que é recuperado (em “Filtrar dados”) e o formato exigido pela RNDS. Por exemplo, a data “01/01/21” é mapeada para “01/01/2021”.

Criar instância de recurso. Os dados já filtrados e mapeados empregam uma estrutura de dados que precisa ser representada em JSON, em conformidade com o recurso FHIR (*resource*) pertinente.

Figura 16. Notificação de exame de COVID-19 na perspectiva de processos e do fluxo de informações entre os requisitos



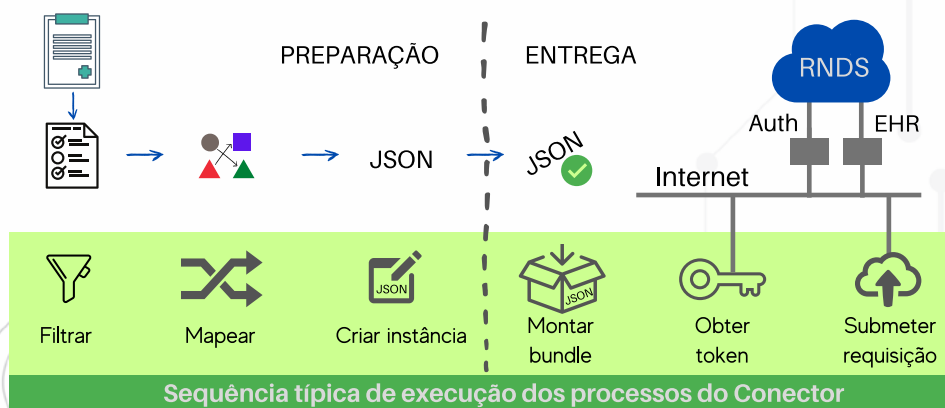
Fonte: autoria própria.

Montar bundle. Empacota os recursos pertinentes na representação JSON de um Bundle (um dos recursos FHIR). Quando executada, essa função também pode realizar a verificação da representação resultante. Ou seja, assegurar que foi criada conforme esperado pela RNDS.

Obter token. Obtém token do web service de segurança da RNDS para acesso aos demais serviços, inclusive aquele pelo qual a notificação de exame é realizada.

Submeter requisição. Notifica o resultado de um exame à RNDS. Essa é a função que, de fato, realiza o envio desejado, faz uso de um web service distinto daquele de segurança, empregado para obter o token de acesso.

Figura 17. Sequência típica de execução dos processos do Conector.

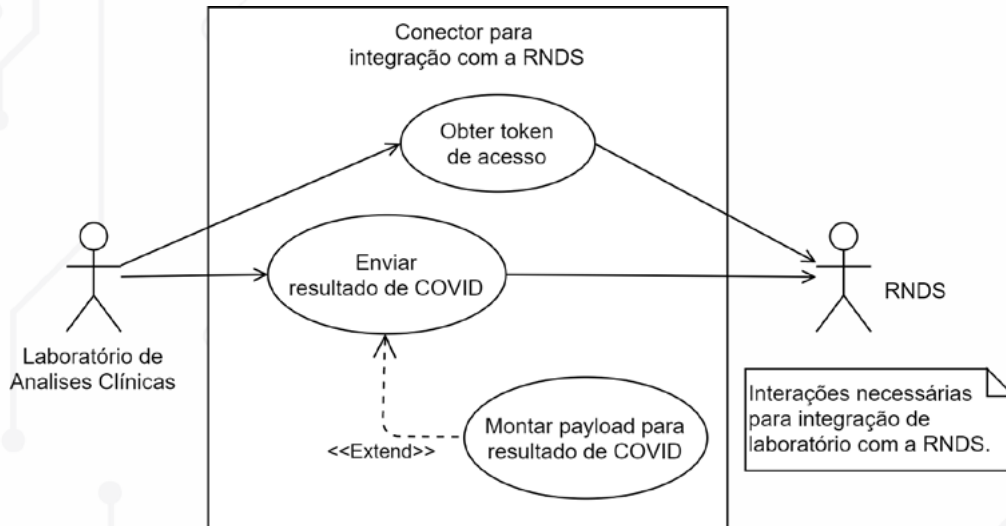


Fonte: autoria própria.

Casos de uso

As interações entre um estabelecimento de saúde e a RNDS podem ser registradas por meio de casos de uso. Aqueles relevantes para a notificação de exame de COVID-19 são: (a) Obter *token* de acesso; (b) Montar *payload* para resultado de COVID e (c) Enviar resultado de COVID.

Figura 18. Interações entre um estabelecimento de saúde e a Rede Nacional de Dados em Saúde registradas por meio de casos de uso



Fonte: autoria própria.

Uma integração pode demandar outras atividades a serem contempladas pelo Conector. Por exemplo, responder se um determinado resultado foi submetido satisfatoriamente ou não, e recuperar a resposta da RNDS para uma dada notificação entregue, dentre outras.

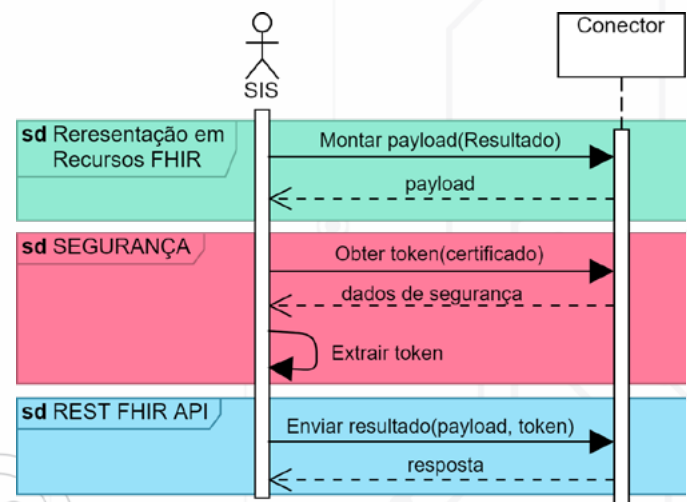
Convém esclarecer que outras atividades podem ser necessárias, mas não são específicas da integração com a RNDS. Em consequência, apenas os casos de uso citados acima são comentados.

Nota

Compreender estes casos de uso significa compreender o que é relevante para qualquer integração com a RNDS, e não apenas para a notificação de resultado de COVID-19. Eles representam a segurança e uma necessidade típica de interoperabilidade em saúde.

Figura 19. Interações com a Rede Nacional de Dados em Saúde

A análise da interação com a RNDS revela um diálogo similar àquele modelado no diagrama de sequência da UML. Observe que não se trata de *design*.



Fonte: autoria própria.

Caso de Uso: Obter token de acesso (Figura 18)

A obtenção de um *token* de acesso é obrigatória e visa contemplar aspectos de segurança. O *token* é obtido por meio de *web service* específico (*Auth*), e o *token* recuperado será exigido em todas as requisições endereçadas aos demais *web services* oferecidos pela RNDS.

Essa atividade é realizada por meio de uma requisição *https*, método *GET*, endereçada ao *web service* de autenticação. Esse serviço é identificado por *Auth*. Na documentação dos ambientes são fornecidos detalhes, inclusive o endereço onde está disponível. Essa requisição depende do certificado digital associado ao estabelecimento de saúde em questão.

Observe que o certificado digital é associado ao estabelecimento de saúde pelo gestor, durante a solicitação de acesso à RNDS (os passos do credenciamento à RNDS fornecem detalhes).

Uma requisição satisfatória deverá retornar o código *HTTP 200*, e o corpo da resposta segue o formato ilustrado abaixo. O principal valor é aquele para a chave *access_token*, uma “longa” sequência de caracteres que, no exemplo abaixo, foi substituída por texto explicativo.

```
{
  "access_token": "token (longa sequência de caracteres)",
  "scope": "read write",
  "token_type": "jwt",
  "expires_in": 1800000
}
```

JSON

A resposta acima indica que o *token* tem validade de 30min, ou 1.800.000ms. Nesse período de validade, a expectativa é que o Conector reutilize o valor de *access_token* para enviar e receber informações.

Noutras palavras, o Conector deve realizar uma requisição para obter o *token* de acesso, guardar o valor recebido e empregá-lo até que seja substituído cerca de 30 minutos depois, quando perde a validade. Não é preciso manter um *token* válido se não há requisição a ser feita.



Caso de Uso: Montar payload... (Figura 18)

Um resultado de COVID-19 deve ser estruturado conforme os perfis definidos por modelos computacionais baseados no FHIR. Isso assegura que a informação enviada será compreendida pela RNDS e, naturalmente, por outro estabelecimento de saúde ao consultar essa mesma informação recuperada da RNDS. Este caso de uso descreve o que é necessário para criar essa estrutura, denominada de *payload* porque esse termo designa o que é enviado/recebido de um *web service*.

A montagem é realizada por quatro processos identificados anteriormente: (a) filtrar dados; (b) mapear dados; (c) criar instância e (d) montar *bundle*. Esses processos, executados nesta ordem, devem produzir um recurso FHIR, um *bundle*, cuja representação JSON é o *payload* a ser submetido.

Esse caso de uso, portanto, reúne processos que visam a converter um resultado de exame laboratorial, do formato empregado pelo estabelecimento de saúde em questão, para a versão equivalente no formato esperado pela RNDS.

A filtragem dos dados exige conhecer o formato de dados empregado pelo SIS do estabelecimento de saúde em questão (entrada) e o que deve ser enviado para a RNDS (saída), o que depende do modelo de informação.

O processo de mapeamento realiza alguma transformação necessária entre formatos de dados.

A criação de instância de recurso exige o conhecimento do recurso FHIR empregado e, em particular, de eventuais perfis que estabelecem restrições a serem observadas.

Por último, o *bundle* empacota os recursos criados no processo anterior, já serializados em JSON, em JSON resultante, o *payload* do resultado de COVID-19 a ser enviado.



Caso de Uso: Enviar resultado de COVID-19 (Figura 18)

Cria uma requisição *https* e a submete à RNDS.

O caso de uso anterior produz, no formato esperado, o conteúdo a ser enviado. Outras duas informações são necessárias para a criação da requisição: a URL e *headers*.

A URL segue o formato <https://{{endereço}}/api/fhir/r4/Bundle>.

Observe que `{{endereço}}` deve ser substituído pelo endereço do *web service* oferecido pela RNDS, conforme o ambiente empregado. Por exemplo, para acesso ao ambiente de homologação o endereço é `ehr-auth-hmg.saude.gov.br`. [Consulte ambiente para detalhes](#).

Dois *headers* devem ser fornecidos: *X-AuthORIZATION-Server* e *Authorization*. O primeiro é definido pela concatenação de “*Bearer*” com o valor do *token* de acesso. O caso de uso Obter *token* de acesso fornece detalhes. O segundo é o Cartão Nacional de Saúde (CNS), o número do CNS, do profissional de saúde em nome do qual a requisição é feita. Necessariamente deve ser um profissional de saúde lotado no estabelecimento de saúde em questão.

De posse do *payload*, URL e *headers*, a requisição pode ser submetida. A resposta de código HTTP 200 indica que a requisição foi executada satisfatoriamente. Nesse caso, dentre os *headers* retornados está *Location*, que informa o identificador único atribuído pela RNDS ao resultado do exame submetido.

Nota

O Postman é uma ferramenta empregada para experimentar *web services* e há uma *collection* (configuração) pronta com requisições para acesso à RNDS .
<https://documenter.getpostman.com/view/215332/TVewY47S>

Sequência clássica para acesso a qualquer serviço oferecido pela RNDS

1. Obter o *token* de acesso, possivelmente de cache. Consultar caso de uso Obter *token* de acesso.
2. Montar valor para *X-AuthORIZATION-Server*. Concatenar “*Bearer*” (observe o espaço em branco) com o valor do *token* de acesso (passo anterior).
3. Definir valor para *Authorization*. Recuperar o CNS do profissional de saúde do estabelecimento em questão, em nome do qual a requisição será submetida.
4. Montar a requisição. Isto exige o endereço e o *path* para a montagem da URL correspondente. Também é possível a existência de parâmetros. A sugestão aqui é consultar URLs prontas para as requisições oferecidas pela RNDS. Veja a nota acima.
5. Submeter a requisição.



Projeto (design)

Os requisitos atribuídos ao Conector podem ser realizados de várias maneiras. A opção adotada é alocar as funções do Conector a um microsserviço acionado por evento que registra a produção de um resultado de COVID-19.

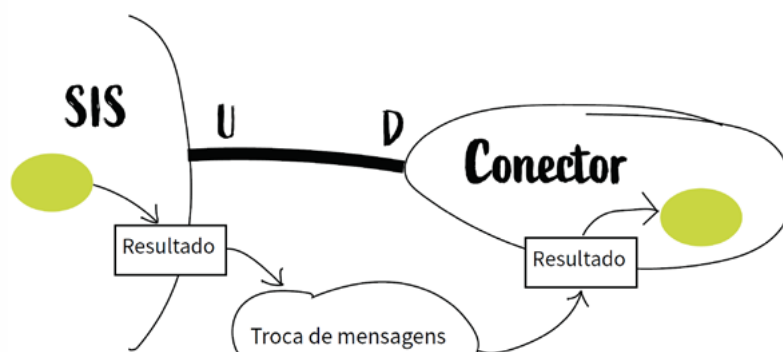
A ausência de um cenário real e concreto admite essa escolha direta, sem outras considerações, sem restrições, que existem no contexto de um estabelecimento de saúde. De fato, essa escolha pode não ser adequada, assim como outras feitas adiantes, por outro lado, não invalida o aprendizado.

Nesta proposta, o SIS é modificado para gerar um evento para cada laudo de COVID-19 produzido. O microsserviço (Conector), ao recebê-lo, encarrega-se de notificar o Ministério da Saúde por meio da RNDS.

A proposta emprega o projeto dirigido por domínio [5] e *context mapping* conforme [6]. Seguramente são referências úteis para compreensão dos fundamentos subjacentes.

Na Figura 20, são incluídos dois *bounded contexts*, um no SIS e outro no Conector, cuja relação é de fornecedor (*Upstream*) para consumidor (*Downstream*). O que significa a existência de um acordo entre SIS e Conector, principalmente quanto ao formato de dados de um evento baseado no modelo de informação a ser considerado. Um evento Resultado (laudo) é sinalizado pelo SIS. O microsserviço (Conector), ao recebê-lo, executa processos comentados anteriormente com a finalidade de entregar para a RNDS, o *payload* correspondente ao resultado registrado no evento.

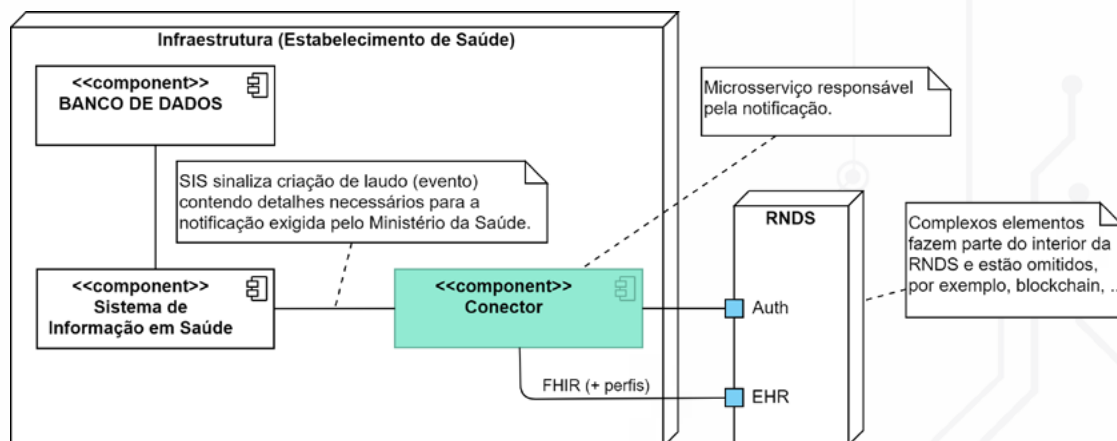
Figura 20. *Bounded contexts* e relação entre o sistema de informação em saúde (*Upstream*) e o Conector (*Downstream*)



Fonte: autoria própria.

Essa decisão estratégica pode ser representada por um diagrama de componentes e, dado que a funcionalidade necessária já está alocada, a questão agora é a organização desse componente

Figura 21. Decisão estratégica representada por diagrama de componentes

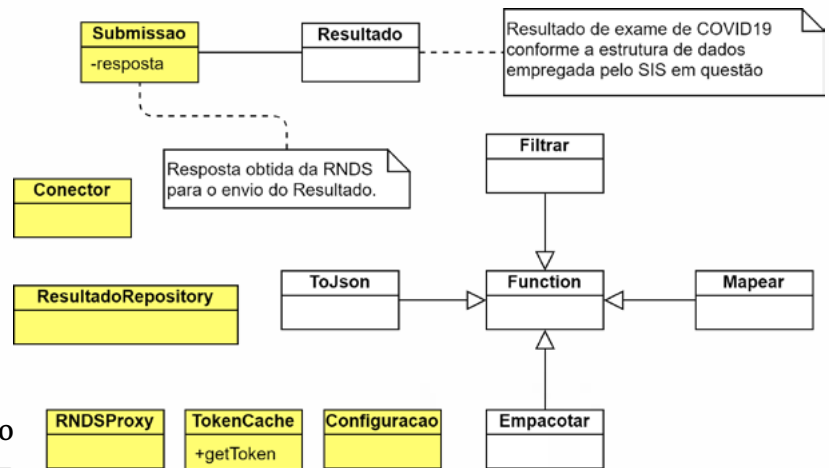


Fonte: autoria própria.

Esboço inicial

Em um primeiro esforço de leitura criteriosa dos casos de uso, foram identificadas algumas classes (Figura 22). Aquelas não coloridas realizam as funções do negócio, enquanto as demais oferecem o suporte necessário.

Figura 22. Esboço inicial de classes de acordo com os casos de uso identificados



A classe Conector encapsula a conexão do SIS com o microserviço. Não confunde com a conexão do componente Conector a RNDS atribuída à classe RNDSPProxy.

Dessa forma, a implementação correspondente, por exemplo, se uma requisição https ou o envio de uma mensagem por meio de ActiveMQ (*message broker*) ou SNS (AWS), fica encapsulada nessa classe. Em consequência, o cliente (SIS) executa uma simples mensagem: `Conector.notifique(resultado)`.

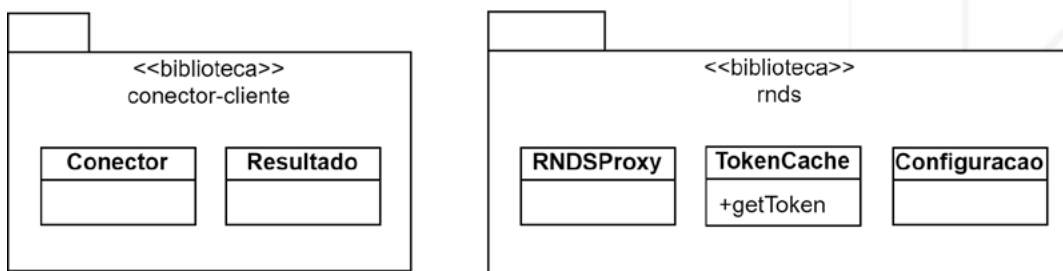
Esse microserviço deve possuir o seu próprio repositório onde são registradas as requisições recebidas para serem notificadas (eventos) e as respostas obtidas das submissões à RNDS. A classe Submissao reúne o que é submetido e a resposta obtida. A resposta inclui o identificador único, atribuído pela RNDS, ao resultado de exame. Talvez um nome melhor para o repositório seja *ConectorRepository*, em vez de *ResultadoRepository*.

O envio propriamente dito de informação para a RNDS está encapsulada na classe *RNDSPProxy*. Esse envio depende do *token* de acesso cuja gestão pode estar encapsulada na classe *TokenCache*. Naturalmente, estas funções dependem de valores que variam ao longo do tempo e são obtidos, nessa proposta, pela classe *Configuracao*. Por exemplo, o certificado digital e a senha correspondente são algumas das informações obtidas por meio dessa classe.

As classes comentadas no parágrafo anterior implementam funções de acesso à RNDS e, naturalmente, são candidatas naturais para serem reutilizadas. Em consequência, uma biblioteca deve acomodar estas funções. A biblioteca RNDS é uma implementação desta decisão (Figura 23).

Dado que as classes Conector e Resultado são empregadas para sinalizar a ocorrência de um evento (produção de um resultado de exame), o código que as emprega não é executado pelo microserviço, mas pelo cliente do microserviço, a opção é disponibilizá-las em uma outra biblioteca. Essa, ao contrário da anterior, deve ser utilizada pela parte do SIS a ser adaptada.

Figura 23. Bibliotecas com classes que promovem a reutilização de código



Fonte: autoria própria.



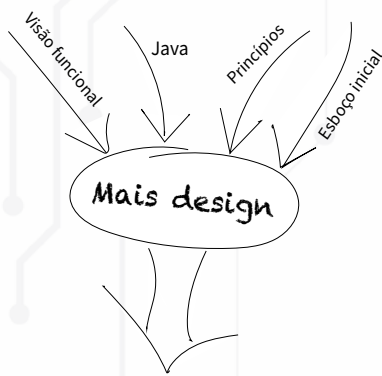
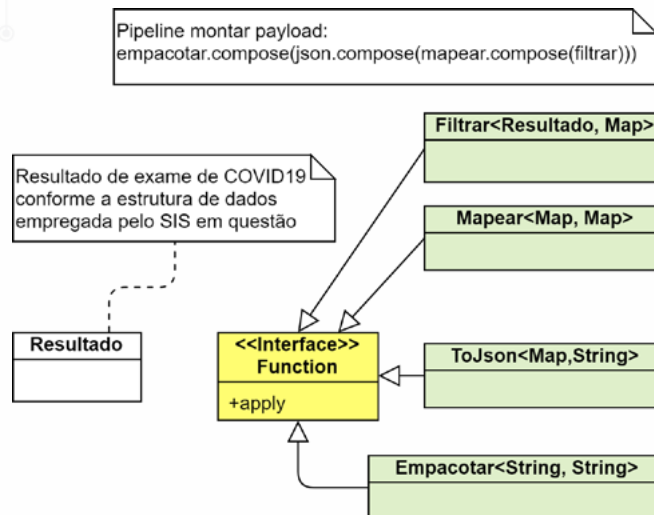


Figura 24. Esboço resultante



Fonte: autoria própria.

Esboço resultante

Após refinamento tem-se resultado com clara influência da visão funcional e de Java (o alerta foi feito ao anunciar a “ausência de um cenário real” que definiria restrições). Sem restrições, essa é uma opção. Além disso, essa interface funcional não é exclusividade de Java, C# e outras linguagens podem implementá-la.

Ao retirar as classes (Figura 22) alocadas às bibliotecas anteriormente identificadas (Figura 23), tem-se a interface fornecida por Java (*Function*) e cinco classes (Figura 24). A opção de criar uma classe por função promove a coesão e a independência entre elas. A combinação destas funções é exibida em uma nota na figura, na qual ocorre a transformação de um *Resultado* (recebido pelo evento) na representação JSON de um *Bundle* (recurso FHIR), esperado pela RNDS.

A partir de uma instância de *Resultado*, a classe *Filtrar* seleciona o que é relevante e deposita em um dicionário, por exemplo, a chave “cns” mantém o CNS do usuário em questão.

A classe *Mapear* realiza eventuais mapeamentos entre códigos empregados pelo laboratório e aqueles esperados pela RNDS. Por exemplo, talvez o laboratório use “sange” em vez do código “SGHEM” esperado pela RNDS. As classes *Tojson* e *Empacotar* estão ligadas à representação de recursos FHIR a ser enviada para a RNDS.

Representação de recursos FHIR

As informações de interesse registradas no modelo de informação precisam ser representadas em formato esperado pela RNDS (modelo computacional baseado no FHIR). O modelo computacional emprega recursos FHIR para o registro de informações em saúde e, para a transferência, essas informações precisam ser representadas em XML ou JSON, conforme esperado pela RNDS.

Além de XML e JSON, o FHIR também admite uma terceira representação, a RDF, que é serializada no formato *Turtle*. Ou seja, é fácil encontrar esses termos na documentação do FHIR. Contudo, no momento em que esse texto foi escrito, os formatos XML e JSON eram os únicos aceitos pelo Ambiente de Produção da RNDS, conforme os comandos abaixo revelam.

```
d:\tmp> curl -o cs.json https://go-ehr-services.saude.gov.br/api/fhir/r4/metadata
d:\tmp> fhirpath -f cs.json -e format
[
  "application/fhir+xml",
  "application/fhir+json"
]
```



ToJson (classe)

A classe ToJson cria a representação JSON para um recurso FHIR. Por exemplo, um resultado de exame laboratorial de SARS-CoV-2 inclui a amostra biológica utilizada pelo exame. Essa amostra é definida pelo recurso FHIR *Specimen*. Em particular, há uma adaptação nacional para esse recurso, o que é chamado de perfil, neste caso, o perfil Amostra Biológica. A representação JSON para uma amostra de “sangue” a ser gerada pela classe ToJson é fornecida abaixo:

```
{
  "fullUrl": "urn:uuid:transient-2",
  "resource": {
    "resourceType": "Specimen",
    "meta": {
      "profile": [
        "http://www.saude.gov.br/fhir/r4/StructureDefinition/BRAmostraBiologica-1.0"
      ]
    },
    "type": {
      "coding": [
        {
          "system": "http://www.saude.gov.br/fhir/r4/CodeSystem/BRTipoAmostraGAL",
          "code": "SGHEM"
        }
      ]
    }
  }
}
```

Como representar recurso FHIR no formato JSON?

A representação no formato JSON de um dado recurso pode ser produzida de várias formas. A classe ToJson, para amostra biológica pode, por exemplo, usar o método Java **String.format()**. Seguramente uma das formas mais simples.

A classe *StringSubstituter* da biblioteca *Apache Commons Text* oferece uma alternativa mais sofisticada. Define-se um *template* contendo variáveis a serem substituídas e, quando requisitado, os valores fornecidos são devidamente substituídos produzindo uma sequência de caracteres resultante.

Também pode-se empregar bibliotecas como Jackson e Gson, dentre outras, para construir o JSON. Serialização e *marshalling* são sinônimos comumente empregados neste contexto e facilitam a busca por alternativas.

Nas opções apresentadas o processo de serialização é explícito. Contudo, usando a biblioteca HAPI FHIR, por meio de métodos de “alto nível” (*fluent interface*), instâncias de recursos FHIR podem ser construídas e serializadas em JSON ou XML, conforme ilustrado pelo programa abaixo.



```

package com.github.kyriosdata.rnds;

import org.hl7.fhir.r4.model.Patient;

import ca.uhn.fhir.context.FhirContext;
import ca.uhn.fhir.parser.IParser;

public class Serializacao {

    public static void main(String[] args) {

        // Cria um recurso FHIR (Patient)
        Patient paciente = new Patient();
        paciente.addName().setFamily("da Silva").addGiven("João");

        FhirContext ctx = FhirContext.forR4();

        // Para XML use ctx.newXmlParser()
        IParser parser = ctx.newJsonParser().setPrettyPrint(true);
        String json = parser.encodeResourceToString(paciente);

        System.out.println(json);
    }
}

```

Empacotar (classe)

A classe Empacotar recebe os recursos serializados em JSON e os agrupa em um *Bundle*, que possui seus próprios atributos. O resultado abaixo omite o valor para *identifier* e os elementos do vetor *entry*, por simplicidade.

```

{
  "resourceType" : "Bundle",
  "type" : "document",
  "timestamp" : "2020-03-20T00:00:00-03:00",
  "meta": {
    "lastUpdated" : "2020-03-20T00:00:00-03:00"
  },
  "identifier" : { ... omitido ... },
  "entry": [
    { ... Resultado de Exame Laboratorial ... },
    { ... Diagnóstico em Laboratório Clínico ... },
    { ... Amostra Biológica ... }
  ]
}

```



Implementação

O *design* fornece uma orientação para o código, mas persistem naturalmente opções de implementação, algumas delas são comentadas abaixo. Em vez de apresentar instruções em uma dada linguagem de programação, detalhes técnicos são comentados.

CONECTOR-CLIENTE (AWS)

A classe Conector deve encapsular a comunicação do SIS com o Conector e, em um cenário onde a nuvem da empresa contratada é empregada, o microsserviço Conector pode ser implementado por uma *Lambda Function* exposta por meio do serviço *API Gateway*. Neste cenário, a implementação do método `Conector.notificar(Resultado)` terá que fazer requisições `https` para o *endpoint* exposto pelo serviço *API Gateway*, que serão redirecionadas para a *Lambda Function* correspondente.

CONECTOR-CLIENTE (ARQUIVO)

O laboratório que usa infraestrutura própria e cujo SIS oferece recurso para exportar um laudo em documento XML, pode não demandar manutenção. O diretório em que tais documentos são depositados pode ser monitorado e, a cada novo arquivo, o método `Conector.notificar(Resultado)` deve iniciar o caso de uso `Enviar resultado de COVID-19`, após a montagem do *payload*.

A opção acima é uma estratégia amplamente empregada. A biblioteca *Chokidar* é uma evidência.

RNDS (BIBLIOTECA)

As funções atribuídas a essa biblioteca são bem específicas e podem ser encapsuladas, independentemente do cenário do estabelecimento de saúde em questão. É o que é feito por meio da *HAPI FHIR API*, dentre outras opções como a biblioteca *RNDS*.

Considerações feitas, o caso de uso `Obter token` está implementado, enquanto enviar resultado de exame, apenas parcialmente, em JavaScript por meio do projeto *RNDS* (projeto `open source`). Observe que o envio propriamente dito está implementado, enquanto as demais funções, filtrar, mapear e outras não estão implementa-

das, pois tem um contexto de dependência com o estabelecimento de saúde.

FUNÇÕES PRINCIPAIS

As estratégias adotadas nas bibliotecas acima dificilmente podem ser replicadas aqui na implementação das funções ditas principais, como filtrar e mapear, por exemplo.

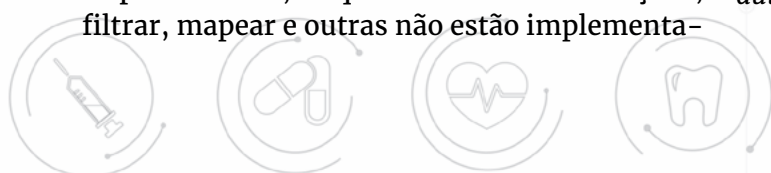
Quando em cenário anterior foi dito que um SIS hipotético é capaz de exportar um documento XML pertinente a um resultado de exame, tem-se uma função útil e que viabiliza o acréscimo do Conector sem necessidade de alteração do SIS. Contudo, o esquema empregado pelo SIS, provavelmente, é diferente daquele estabelecido pelo modelo computacional e também pelo informacional correspondente. Em consequência, não há como implementar uma função filtrar de forma genérica, mas específica e caso por caso.

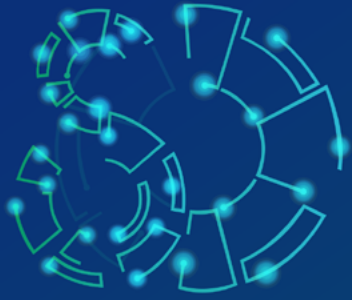
Apesar de não ser viável uma implementação que possa ser reutilizada, é possível indicar ferramentas úteis aos desenvolvedores, por exemplo, que realizam operações sobre documentos XML. Observe que, mesmo neste exemplo, a sugestão pode não se aplicar, pois um formato binário próprio pode ser empregado pelo SIS para exportar um resultado de exame, o que torna a indicação à XML irrelevante.

JAVA E JAVASCRIPT

Em <https://github.com/kyriosdata/rnds> encontram-se os projetos `rnds-java` e `rnds-js`, ambos ilustram como obter o *token* de acesso, respectivamente nas linguagens Java e JavaScript.

Aos interessados, muita informação pode ser encontrada na internet para o assunto “*ssl client authentication*”.





EDUCAÇÃO E CAPACITAÇÃO
DE RECURSOS HUMANOS
EM **SAÚDE DIGITAL**

Integração com a Rede Nacional de Dados em Saúde - RNDS

Unidade 6 Ambiente de desenvolvimento

Fábio Nogueira de Lucena
Plínio de Sá Leitão Júnior



Ambiente de Desenvolvimento



A criação e manutenção de *software*, responsabilidades do Integrador, são atividades extensivamente cobertas pela Engenharia de *Software*. Não é viável reapresentar aqui o que é relevante para cumprir esta responsabilidade, por outro lado, as especificidades da integração com a RNDS, via padrão FHIR, não é assunto comumente coberto nessa área, mas é nesta Unidade.

As tecnologias consideradas úteis à criação do Conector e manutenção de *software*, neste caso um SIS, são apresentadas e acompanhadas de explicações pertinentes ao possível uso neste contexto.

A ordem de exposição das tecnologias vai dos elementos mais básicos para aqueles derivados e específicos (Videoaula 7).

Videoaula 7: Ambiente de desenvolvimento



Fonte: autoria própria.

Obter token

No repositório [RNDS](#), nos diretórios *libs/rnds-java* e *libs/rnds-js*, encontram-se dois projetos contendo implementações em Java e JavaScript, respectivamente, para a obtenção do *token* de acesso.



Recurso FHIR no formato JSON

Na seção “Como representar recurso FHIR no formato JSON?” (pág. <?>) foi ilustrado como um recurso FHIR pode ser criado e serializado em Java em menos de 20 linhas de código usando a biblioteca HAPI FHIR.

Em JavaScript, Smart on FHIR JavaScript (<https://github.com/smart-on-fhir/client-js>) é uma opção para facilitar tal serialização. Objetos em JavaScript terão que ser construídos conforme a especificação do recurso em questão, definindo propriedade por propriedade. A serialização do objeto resultante é por meio de `JSON.stringify(recurso)`.

Em .Net a biblioteca oficial de implementação do HL7 FHIR é a [Firely .Net SDK](#). Por meio do uso desta biblioteca tem-se código semelhante àquele de Java já citado acima. Convém observar que não necessariamente essa é uma escolha do Integrador. Por fim, JavaScript, Java e C# talvez sejam as opções mais comuns, mas não são as únicas.

Validar recurso FHIR

A validação visa assegurar que recursos criados e/ou recebidos são válidos. Vários [critérios](#) podem ser considerados na validação de recursos FHIR, como a presença de elementos obrigatórios, a cardinalidade de coleções e outros.

Além das exigências predefinidas pelo FHIR, também é possível validar a conformidade em relação a perfis (que introduzem restrições para atender necessidades locais não contempladas pelo FHIR). De fato, a RNDS estabelece dezenas de perfis (*profiles*), *ValueSets*, *CodeSystems* e *Extensions*.

Adaptação nacional (definições)

Naturalmente, todas as definições nacionais devem ser do conhecimento de integradores e serem empregadas nas validações realizadas. Todas elas podem ser baixadas do [Portal de Serviços](#) da RNDS ou ainda do serviço Simplifier.Net, onde as definições da RNDS também estão [publicadas](#).



Recurso válido (e outro não)

Abaixo segue uma Amostra Biológica em conformidade com as restrições nacionais. Por curiosidade, “SGHEM” é o código para “sangue”.

```
{
  "resourceType": "Specimen",
  "meta": {
    "profile": [
      "http://www.saude.gov.br/fhir/r4/StructureDefinition/BRAmostraBiologica-1.0"
    ]
  },
  "type": {
    "coding": [
      {
        "system": "http://www.saude.gov.br/fhir/r4/CodeSystem/BRTipoAmostraGAL",
        "code": "SGHEM"
      }
    ]
  }
}
```

Um laboratório, ao construir uma Amostra Biológica, deve observar a restrição de uso de vários elementos do recurso *Specimen*, dentre eles, *receivedTime* e *status*, além de outros. Ou seja, uma Amostra Biológica, em conformidade com os perfis nacionais, não deve conter *receivedTime* nem *status* (entre os restritos). A amostra acima satisfaz essas e outras restrições.

Uma Amostra Biológica com o elemento *status*, por outro lado, conforme ilustrada abaixo é, portanto, uma Amostra Biológica que não está em conformidade com as especificidades nacionais.

```
{
  "resourceType": "Specimen",
  "meta": {
    "profile": [
      "http://www.saude.gov.br/fhir/r4/StructureDefinition/BRAmostraBiologica-1.0"
    ]
  },
  "status": "available",
  "type": {
    "coding": [
      {
        "system": "http://www.saude.gov.br/fhir/r4/CodeSystem/BRTipoAmostraGAL",
        "code": "SGHEM"
      }
    ]
  }
}
```



De fato, ao se tentar validar o recurso fornecido acima, o processo indica a presença do elemento *status*, definido para o recurso *Specimen* conforme originalmente estabelecido pelo FHIR, mas excluído pela RNDS. A mensagem retornada pelo aplicativo gráfico, comentado na sequência, claramente registra esta não conformidade:

```
Element 'Specimen.status': max allowed = 0, but found 1
```

Como validar um recurso?

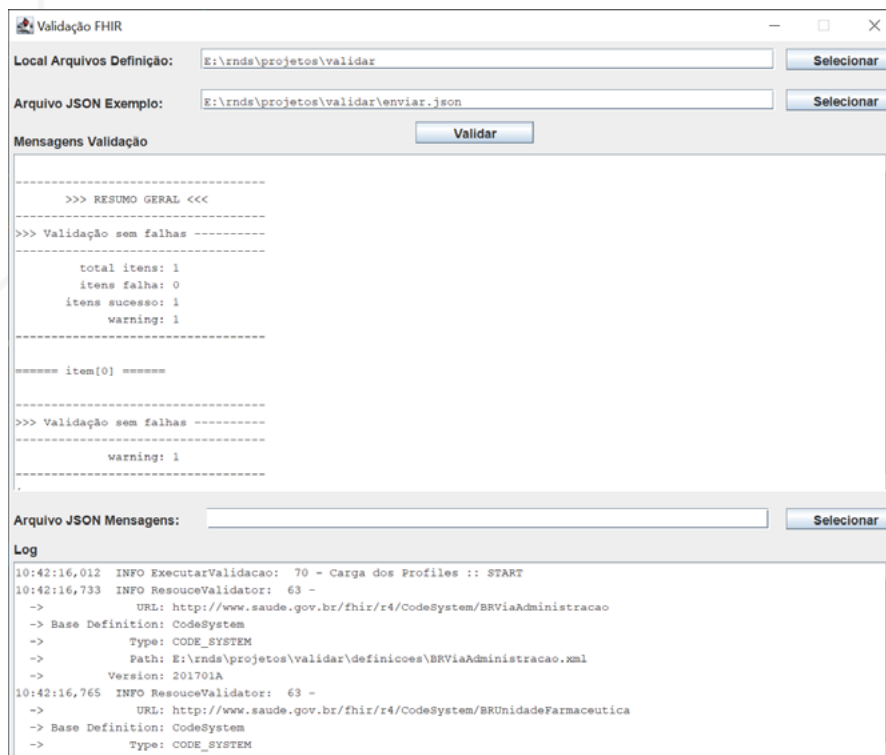
É possível validar manualmente um recurso, sem apoio de um *software*. Contudo, não é uma alternativa prática. Felizmente, há várias opções, que podem ser utilizadas conforme a necessidade.

Interface gráfica

Um aplicativo gráfico é disponibilizado pelo [Portal de Serviços](#) da RNDS. Abaixo (Figura 25) segue a tela desse aplicativo após a validação de um recurso.

No primeiro campo, forneça o diretório contendo as definições nacionais (veja como obtê-las acima), depois indique a representação do recurso a ser validado no formato JSON e, por fim, clique no botão **Validar** para que a validação seja executada.

Figura 25. Interface gráfica disponibilizada pelo Portal de Serviços da Rede Nacional de Dados em Saúde

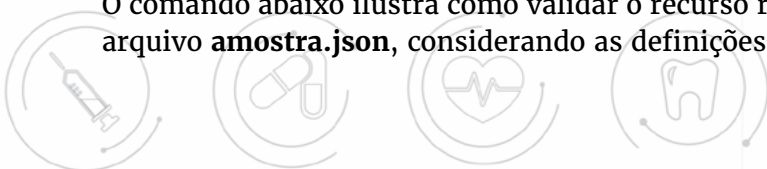


Fonte: autoria própria.

Linha de comandos

Uma alternativa é o aplicativo de linha de comandos `validator_cli`. Detalhes da instalação e opções disponíveis estão devidamente [documentadas](#).

O comando abaixo ilustra como validar o recurso representado no formato JSON e depositado no arquivo `amostra.json`, considerando as definições contidas no diretório `d:\definicoes`.



```
java -jar validador_cli.jar amostra-invalida.json -ig d:\definicoes -recurse
```

Na saída padrão será exibida a mensagem:

```
Specimen.status: max allowed = 0, but found 1
```

Há muitas opções, conforme a documentação indicada acima. Duas delas incluem `-output saida.json` para depositar a saída no arquivo indicado (nesse caso, `saida.json`) e `-version 4.0.1` para indicar a versão do FHIR a ser considerada no processo de validação. Se não fornecido, a versão mais recente é empregada. Observe que a versão empregada pela RNDS, neste momento, é a versão 4.0.1, aponte o seu navegador para <https://go-ehr-services.saude.gov.br/api/fhir/r4/metadata> e terá como resposta o recurso *CapabilityStatement* pertinente ao servidor FHIR da RNDS. A propriedade *fhirVersion* tem como valor **4.0.1**.

Nota. Quando é executado pela primeira vez, várias definições do FHIR serão baixadas o que torna esta operação mais lenta que as execuções posteriores.

Simplifier.Net (também valida)

Além de publicar perfis FHIR, como feito pela RNDS, você também pode executar validações diretamente no portal, pelo navegador. Essa é uma opção particularmente útil para ambientação com o FHIR e com os perfis nacionais.

É preciso se cadastrar no [Simplifier.Net](#) (processo gratuito).

Em sua conta, no canto superior direito, encontra-se a opção **SNIPPET**. Você pode empregar esse recurso para compartilhar recursos FHIR e para validar tais recursos. Ao clicar nessa opção abre-se uma tela como ilustrada na Figura 26.

Figura 26. Snippet para compartilhar e validar recursos FHIR

```
1 {
2   "resourceType": "Specimen",
3   "meta": {
4     "profile": [
5       "http://www.saude.gov.br/fhir/r4/StructureDefinition/BRAmostraBiologica-1.0"
6     ]
7   },
8   "status": "available",
9   "type": {
10    "coding": [
11      {
12        "system": "http://www.saude.gov.br/fhir/r4/CodeSystem/BRTipoAmostraGAL",
13        "code": "SGHEM"
14      }
15    ]
16  }
17 }
```

Fonte: autoria própria.

Observe que terá que indicar a versão, R4, e o escopo, nesse caso, as definições da RNDS, conforme ilustrado. Nesse exemplo, por simplicidade é experimentado o recurso *Specimen* e, em particular,

o perfil nacional Amostra Biológica. Observe que foi introduzido o elemento *status*, justamente para provocar um erro.

Pode-se então salvar o recurso e, na sequência, validar (canto superior direito), o que conduz ao resultado abaixo.

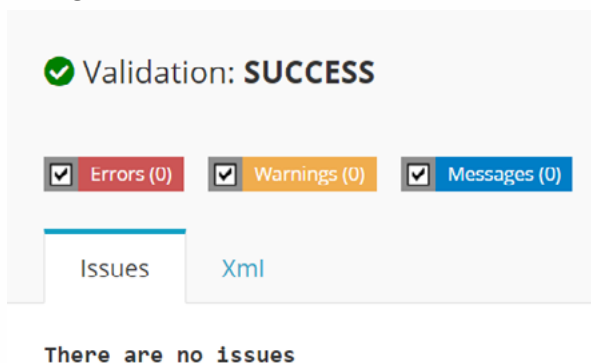
Figura 27. Salvar e validar recurso



Fonte: autoria própria.

Observe que pode ser retornado ao modo edição e, se o elemento não permitido no perfil (*status*) é removido, tudo realizado via navegador, seguido do pedido de nova validação, o resultado é este na imagem abaixo.

Figura 28. Resultado de validação de recurso



Fonte: autoria própria.

FHIR Resource Editor (FRED)

FRED é um [editor de recursos](#) oferecido via navegador. Você pode carregar um recurso, editá-lo e exportá-lo em JSON. Em geral, não se espera que um recurso FHIR seja criado dessa forma, contudo, como ferramenta de aprendizado, convém usá-lo para ampla compreensão da estrutura dos recursos.

Servidor FHIR

FHIR é uma especificação, um padrão, e computadores não exatamente executam especificações. A execução do FHIR depende de *software*. Há várias implementações do padrão FHIR e algumas são [open source](#), como a implementação de referência do FHIR, a [HAPI FHIR](#).

A integração com a RNDS significa que há uma implementação do FHIR oferecida pela RNDS, observando os perfis nacionais, por meio da qual estabelecimentos de saúde enviam e consultam



informações em saúde. A implementação do FHIR oferecida pela RNDS está disponível em dois ambientes, conforme “Os ambientes” (pág. 26).

Estabelecimentos de saúde não precisam disponibilizar uma implementação do FHIR para interagir com a RNDS, em vez disso, precisam criar um “Conector” (pág. 48).

Os ambientes disponibilizados pela RNDS são para homologação e produção, ou seja, não podem ser utilizados durante o desenvolvimento ou visando o aprendizado. Para tal, pode-se empregar um dos vários servidores amplamente disponíveis na *internet*, ou até disponibilizar um localmente.

Servidor FHIR (para experimentação)

Há [vários](#) servidores disponíveis para experimentação com o FHIR. Ou seja, são computadores que estão executando alguma implementação do FHIR e não cobram nada por isso. Esses servidores são empregados apenas para testes, o que é suficiente para desenvolvedores que queiram conhecer mais sobre o padrão.

Cada um desses servidores está apto a interagir com outro *software* por meio do FHIR. Dessa forma, sem exigências que são necessárias em um cenário real, até porque os dados disponibilizados não são dados reais, você pode submeter requisições e observar os recursos retornados no formato da sua escolha, em geral XML ou JSON, conforme visto anteriormente.

Desenvolvedores podem empregar esses servidores para se ambientar com o padrão FHIR. De fato, o acesso ao ambiente de homologação da RNDS pode ser precedido pelo contato com um desses servidores.

Antes que o responsável pelo laboratório obtenha um certificado digital, peça o credenciamento e aguarde pela homologação, desenvolvedores podem empregar um desses servidores para experimentação. Convém observar, contudo, que, nesses casos, tais servidores não estarão aderentes aos perfis estabelecidos pela RNDS. Ou seja, seguem o padrão FHIR, mas sem “obedecer” as especificidades impostas pelos perfis nacionais.

Em tempo, a implementação de referência do FHIR, Hapi FHIR, citada anteriormente, também possui um servidor para [testes](#).

Servidor FHIR (local)

Ainda há possibilidade de disponibilizar um servidor localmente.

Faça o *download* do aplicativo *Command Line Interface Tool* for HAPI FHIR. A versão 5.1.0 pode ser baixada [aqui](#).

Extraia o conteúdo do arquivo *.zip*. Isso pode ser feito com o utilitário [jar](#) que acompanha o JDK. Possivelmente com o comando:

```
jar xvf hapi-fhir-5.1.0-cli.zip
```

Coloque o servidor FHIR em execução com o comando abaixo. Consulte detalhes das opções [aqui](#).

```
hapi-fhir-cli run-server -v r4
```

Gerar informação em saúde (para teste)

Synthea™ é um gerador sintético de história médica de pacientes sintéticos. Vários recursos FHIR são contemplados por esse gerador. Ou seja, pode ser empregado para produção de informação em saúde a ser fornecida para um servidor FHIR de experimentação. Consulte detalhes no repositório do [Synthea](#).



FHIR

FHIR é um padrão para a troca de informação em saúde e já fizemos várias referências a ele neste documento. O acrônimo vem de *Fast Healthcare Interoperability Resources*.

Destacado no próprio nome, *resource*, ou **recurso**, é o elemento básico empregado para a interoperabilidade de SIS usando FHIR.

Um recurso representa algum tipo de entidade do cuidado em saúde. Por exemplo, o recurso [Patient](#) é empregado para dados demográficos ou outra informação administrativa acerca do indivíduo. Por outro lado, se o que se deseja trocar são medidas como pressão ou temperatura, por exemplo, então fará uso do recurso [Observation](#).

No momento em que este documento foi escrito, estavam definidos 145 tipos distintos de [recursos](#). Todos eles devidamente documentados.

Está além do escopo apresentar o FHIR em abrangência e profundidade. Os primeiros passos podem ser orientados por [aqui](#).

Forge (editor de perfis)

Os recursos do FHIR visam a contemplar um conjunto razoável de cenários, mas não é possível abarcar os usos específicos de todo o planeta. Felizmente, o FHIR permite “personalizações” por meio de perfis (*profiles*).

A definição de perfis é um esforço de modelagem de informação em saúde. Desenvolvedores terão que ter acesso aos perfis que personalizam os recursos a serem trocados. Afinal, toda a troca de dados deverá estar em conformidade com os perfis definidos. Consulte os [perfis](#) definidos pela RNDS para detalhes.

A edição de um perfil é melhor realizada com o apoio de um editor específico para essa finalidade. [Forge](#), disponível para Windows, é uma opção.

Convém ressaltar que para a integração com a RNDS, não será necessário definir perfis (*profiles*). Essa é uma atribuição da RNDS e os [perfis](#) já estão definidos. O acesso a eles, contudo, é necessário para verificar se as personalizações definidas são atendidas.

Simplifier.NET

[Simplifier.NET](#) é um serviço que permite publicar e consultar perfis (*profiles*). Convém ressaltar que perfis precisam ser conhecidos entre as partes que irão interagir e que ferramenta a *Forge*, por exemplo, apresentada na seção anterior, permite apenas criá-los.

As facilidades oferecidas pelo Simplifier.NET incluem a navegação entre recursos e definições pertinentes, o que é desejável para quem deseja consultar perfis.

Os [perfis](#) definidos pela RNDS podem ser consultados via esse serviço.

JSON (há ainda XML e RDF)

Quando se usa o FHIR para troca de dados, o que ocorre é a troca de recursos. Tais recursos podem trafegar representados em vários formatos, dentre eles, JSON. Este guia de integração faz uso de JSON.



O portal oficial define [JSON](#) (*JavaScript Object Notation*) como um formato leve para troca de dados. Ainda acrescenta que este formato é fácil para seres humanos escreverem e lerem.

[XML](#), à semelhança de JSON, é comumente empregado em *web services* (serviços oferecidos por meio da internet). Por outro lado, [RDF](#) está associado, em geral, a questões semânticas, por exemplo, quando se deseja realizar inferência sobre os dados.

FHIRPath

FHIRPath é um mecanismo para manusear documentos JSON contendo recursos FHIR. Trata-se de proposta similar à *JsonPath* e *XML Path*, por exemplo. Contudo, possui funções específicas para documentos contendo recursos FHIR, ao contrário de *JsonPath*. Adicionalmente, *FHIRPath* é usado pela *Clinical Quality Language* ([CQL](#)).

A [definição](#) de *FHIRPath* está disponível, inclusive em [detalhes](#). Adicionalmente, consultas baseadas em *FHIRPath* podem ser executadas por meio do portal [clinfhir](#). A implementação [fhirpath.js](#) (*JavaScript*) é empregada nos exemplos abaixo.

A instalação pode ser feita via comando `npm install -g fhirpath`. Essa instalação é global por simples conveniência.

Qual o tipo do recurso?

```
$ yarn fhirpath -- -f exemplo.json -e "resourceType"
```

Quando o *bundle* foi atualizado pela última vez?

```
$ yarn fhirpath -- -f exemplo.json -e "meta.lastUpdated"
```

Qual o identificador local (definido pelo laboratório) do *bundle*?

```
$ yarn fhirpath -- -f exemplo.json -e "identifier.value"
```

Quantos recursos estão reunidos no *Bundle*?

```
$ npm run fhirpath -- -f exemplo.json -e "entry.count()"
```

Quais os tipos dos recursos reunidos no *Bundle*?

```
$ yarn fhirpath -- -f exemplo.json -e "entry.resource.resourceType"
```

Bibliotecas

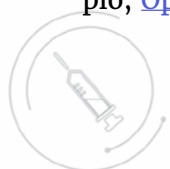
Estão disponíveis duas bibliotecas para ilustrar código que realiza a integração com a RNDS (submissão de requisições), uma para a linguagem Java ([rnds-java](#)) e outra para *JavaScript* ([rnds-js](#)).

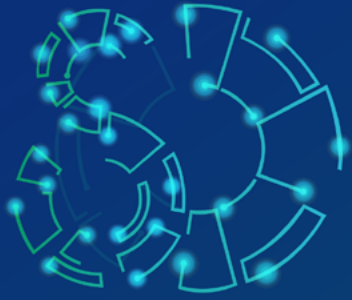
IMPORTANTE

Nem Java nem *JavaScript*, tampouco as bibliotecas são recomendações do Datasus, são apenas exemplos.

keytool

keytool é uma ferramenta usada via linha de comandos para gerenciar um *keystore*, ou base de dados contendo chaves, cadeias de certificados e certificados confiáveis (*trusted certificates*). A *keytool* é uma das várias ferramentas que acompanham as distribuições de Java (JDK), por exemplo, [OpenJDK](#).





EDUCAÇÃO E CAPACITAÇÃO
DE RECURSOS HUMANOS
EM **SAÚDE DIGITAL**

Integração com a Rede Nacional de Dados em Saúde - RNDS

Unidade 7 Processo de credenciamento

Fábio Nogueira de Lucena
Plínio de Sá Leitão Júnior



Processo de Credenciamento

O acesso à RNDS por meio de um SIS de um estabelecimento de saúde envolve atores, atividades, documentos, ordenados em uma sequência, tudo isto denominado de processo de credenciamento.

O processo de credenciamento de um estabelecimento de saúde junto à RNDS é realizado em duas fases. Na primeira o estabelecimento requisita acesso ao ambiente de homologação. Na segunda, após obtido acesso ao ambiente de homologação, o estabelecimento requisita acesso ao ambiente de produção. Quando este último acesso é concedido, o estabelecimento está autorizado a contribuir com informações em saúde no âmbito do território nacional.

Os atores e a interação entre eles, em um fluxo típico, representativo de vários cenários, são os tópicos seguintes. Variações são naturais, tendo em vista que muitas atividades são internas ao estabelecimento de saúde (Videoaula 8).

Videoaula 8: Processo de credenciamento



Fonte: autoria própria.

Público-alvo

O público-alvo são os estabelecimentos de saúde do país, o conjunto específico é definido conforme a necessidade de troca de informação.

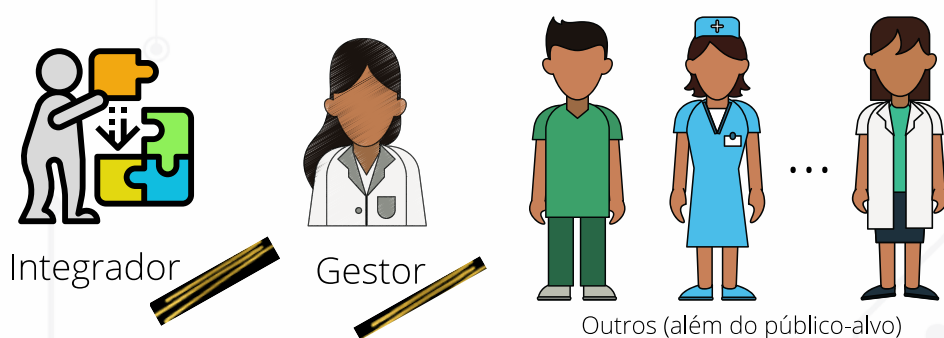


A primeira necessidade contemplada para a interoperabilidade via RNDS foi o envio de resultados de exame de SARS-CoV-2, o que cabe somente a estabelecimentos de saúde específicos (apenas os laboratórios de análises clínicas). Ao longo do tempo, todos os estabelecimentos de saúde estarão integrados à RNDS. Por exemplo, quando o Sumário de Alta (SA), Registro de Atendimento Clínico (RAC) e outras informações forem contempladas, novos estabelecimentos serão incluídos.

Profissionais realizam várias atividades no âmbito de estabelecimentos de saúde. A interoperabilidade com a RNDS introduz novas responsabilidades distribuídas entre dois destes profissionais: o gestor e o integrador. Em consequência, torna-se necessário estabelecer claramente quais são as novas responsabilidades e, naturalmente, o que se espera dos profissionais que as assumem.

Figura 29. Colaboradores nos estabelecimentos de saúde

Estabelecimento de saúde (colaboradores)



Fonte: autoria própria.

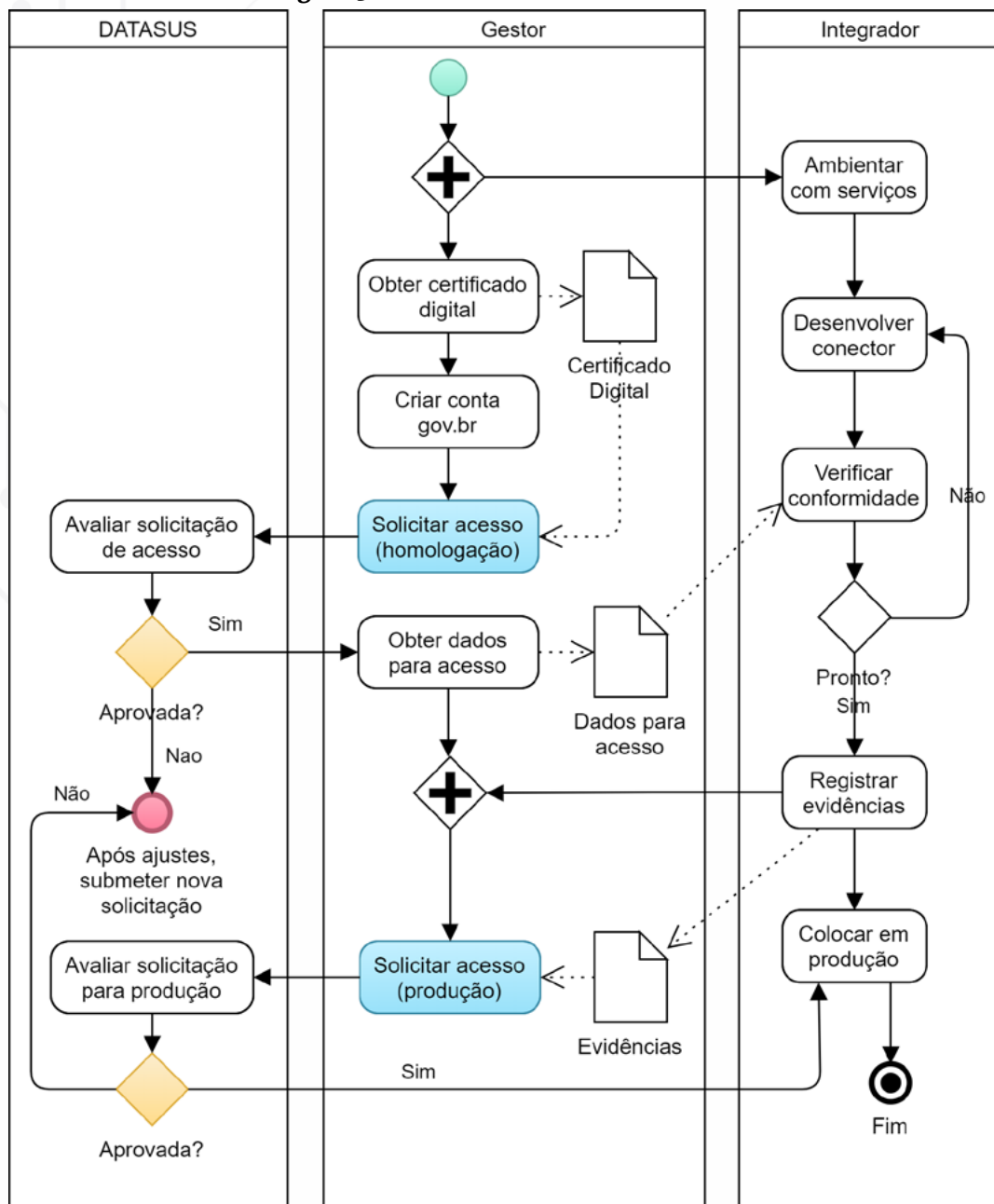
Apesar da informação em saúde variar, e com ela os estabelecimentos de saúde pertinentes, dois profissionais persistem, assim como suas atividades, para a integração com a RNDS: o gestor e o integrador. Antes de detalhar as atribuições deles, convém mostrar o mapa de atividades.



Passo a passo

Além do Gestor e do Integrador, já ressaltados anteriormente, o Datasus é o terceiro ator, dentre aqueles de maior destaque para o processo de credenciamento. Isso porque o Portal Gov.BR e a autoridade certificadora empregada para obter o certificado digital do estabelecimento também estão envolvidos.

Figura 30. Processo de credenciamento



Fonte: autoria própria.

Alguns desses passos (Figura 30), os principais, são comentados nas seções seguintes.

Obter o certificado digital

Certificado digital é o instrumento empregado para identificar uma pessoa (e-CPF) ou empresa (e-CNPJ) no mundo digital.

O estabelecimento de saúde é reconhecido pela RNDS por meio do e-CPF do seu gestor ou do seu próprio e-CNPJ. A decisão entre e-CPF ou e-CNPJ é realizada no momento da solicitação de acesso, que exige o acesso ao certificado digital a ser empregado nas interações do Conector com a RNDS.

Um certificado digital pode ser adquirido junto a uma autoridade certificadora. Existem várias autoridades certificadoras que emitem certificados legalmente válidos no território nacional, ou certificados [ICP-Brasil](#), exigidos pela RNDS. O [Serpro](#) e os [Correios](#) são dois exemplos.

O certificado digital a ser fornecido no credenciamento do estabelecimento de saúde, junto à RNDS, seja o e-CPF ou e-CNPJ, pode ser de dois tipos: A1 ou A3. O A3 é fornecido por meio de um *token (hardware)* ou *smart card* e pode ter uma duração (validade) maior do que o A1. Essas opções cabem ao estabelecimento de saúde decidir.

Seja e-CPF ou e-CNPJ, tipo A1 ou A3, o certificado deve ser mantido em segurança, assim como a senha empregada para acesso ao conteúdo do certificado. Tanto o certificado digital quanto a senha de acesso (código) ao conteúdo do certificado serão empregados pelo conector para obter acesso aos serviços oferecidos pela RNDS.

Convém esclarecer que o certificado digital do estabelecimento de saúde informado no processo de solicitação de acesso deve ser o mesmo empregado para autenticação. Ou seja, a RNDS irá identificar as requisições do estabelecimento de saúde por meio do certificado empregado no processo de solicitação de acesso.

Por fim, a existência de um certificado e de uma senha (código) de acesso exige a adoção de “boas práticas” de segurança da informação pelo estabelecimento de saúde.

A execução satisfatória dessa atribuição resulta em:

- Certificado digital do estabelecimento de saúde disponível.
- Senha de acesso ao conteúdo do certificado digital definida.

Criar conta gov.br

O acesso aos serviços digitais oferecidos pelo governo exige uma conta que qualquer cidadão pode criar pelo portal <https://acesso.gov.br>.

O gestor do estabelecimento de saúde deverá criar uma conta gov.br, caso não possua uma, pois essa é uma condição necessária para requisitar a solicitação de acesso à RNDS.

Solicitar acesso

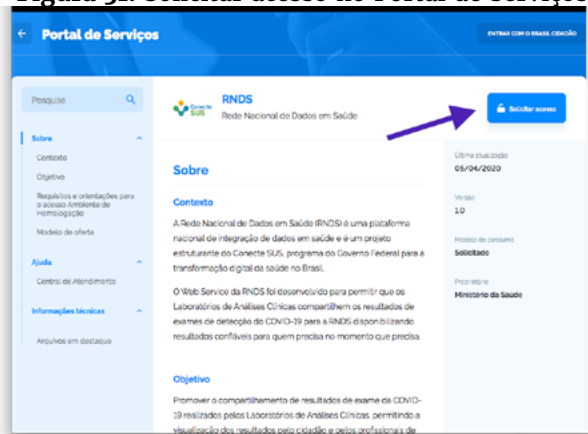
O Portal de Serviços, <https://servicos-datasus.saude.gov.br/>, oferece um catálogo de *web services* e APIs para integração com a RNDS e outros sistemas do Ministério da Saúde (MS). É por meio do portal de serviços que a solicitação de acesso à RNDS é feita.

Um recorte da página do Portal de Serviços é exibida na Figura 31. Observe o botão ENTRAR COM O BRASIL CIDADÃO (canto superior e lado direito) e os vários serviços oferecidos como REL, RIA e outros.

Para solicitar o acesso aos serviços da RNDS é necessário selecionar o ícone do serviço RNDS (opção destacada na Figura 31). Quando selecionado, a página destino é exibida abaixo, contendo vários tópicos como contexto, objetivo, e outros, dentre os quais o botão “Solicitar acesso”.

Quando o botão “Solicitar acesso” é pressionado, a página resultante é a exibida na Figura 32.

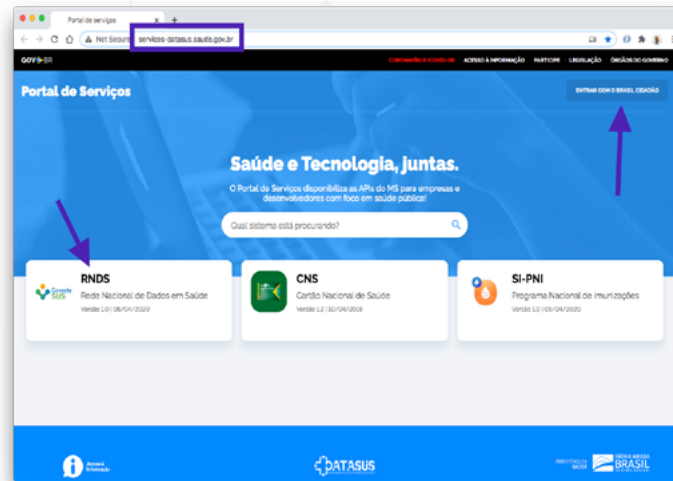
Figura 31. Solicitar acesso no Portal de Serviços



Fonte: autoria própria.



Figura 32. Página de destino após solicitar acesso no Portal de Serviços



Fonte: autoria própria.

Para entrar, é necessário que o gestor acesse sua conta gov.br.

A solicitação de acesso exige o fornecimento de várias informações, conforme Figura 33. As etapas incluem Responsável, Sistema, Operação e Finalização. Observe que as tarjas azuis na parte superior ocultam a identificação do usuário gov.br em questão.

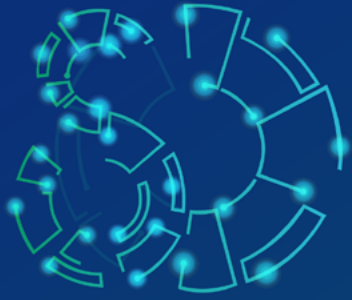
Figura 33. Origem da solicitação no Portal de Serviços



Fonte: autoria própria.

Conforme a Figura 33, a primeira informação a ser fornecida, da primeira etapa, indica se a solicitação é interna ao Ministério da Saúde ou não. Na ilustração, a opção “Outros” encontra-se selecionada, indicando que a solicitação não é interna, ou seja, parte de um SIS não gerido pelo Ministério da Saúde.





EDUCAÇÃO E CAPACITAÇÃO
DE RECURSOS HUMANOS
EM **SAÚDE DIGITAL**

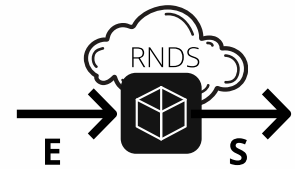
Integração com a Rede Nacional de Dados em Saúde - RNDS

Unidade 8 **Entradas e saídas**

Fábio Nogueira de Lucena
Plínio de Sá Leitão Júnior



Entradas e Saídas



Uma perspectiva da RNDS é definida pelo conjunto de dados de entrada e aquele de saída ou, de outra forma, as informações em saúde fornecidas e aquelas obtidas, respectivamente.

A primeira necessidade de troca de informação em saúde contemplada pela RNDS é o Resultado de Exame Laboratorial (REL). Compreender como um REL está organizado em termos de recursos FHIR e dos perfis correspondentes exercita as habilidades necessárias para compreender outros cenários de integração com a RNDS. Noutras palavras, quem compreende o REL está apto a compreender outras informações em saúde à luz do FHIR.

Nesse sentido, para que nenhum item fornecido/recebido não seja comentado, antes de tratar do REL, segue a identificação da capacidade do servidor FHIR e da informação de autenticação. Essas últimas, convém destacar, não são informações em saúde, mas técnicas e necessárias para a operação em conjunto com a RNDS (Videoaula 9).

Videoaula 9: Entradas e saídas



Fonte: autoria própria.

Capacidade (CapabilityStatement)

O conjunto das capacidades oferecidas por um servidor FHIR são declaradas em um recurso que não está associado à informação em saúde, o recurso [CapabilityStatement](#). Esse recurso declara informações técnicas.



As informações contidas neste recurso incluem a versão do padrão FHIR, a identificação e a versão do *software* em execução no servidor e outras, além das requisições disponíveis para cada um dos recursos. Ou seja, um *CapabilityStatement* é a resposta oferecida por um servidor FHIR para informar quem ele é e as funcionalidades que oferece.

Requisitando

As capacidades de um servidor FHIR (*CapabilityStatement*) podem ser recuperadas diretamente do servidor FHIR, sem exigência de segurança, inclusive no ambiente de produção da RNDS. É uma forma de se identificar para um *software* cliente em potencial.

Para ilustrar, siga o endereço <https://go-ehr-services.saude.gov.br/api/fhir/r4/metadata> e terá acesso à resposta oferecida pelo ambiente de produção da RNDS para o estado de Goiás no próprio navegador. Via linha de comandos, pode-se usar o comando abaixo (*curl*):

```
curl -o go.json https://go-ehr-services.saude.gov.br/api/fhir/r4/metadata
```

Esse comando deposita, no diretório em que é executado, o arquivo **go.json** contendo a representação, em formato JSON, da “carteira de identidade do servidor FHIR” (*CapabilityStatement*). Detalhes dessa requisição são fornecidos em “FHIR API” (pág. 100), aqui, a atenção está nas informações que trafegam para a RNDS e da RNDS.

Desmitificando

Esse recurso é a declaração de um servidor FHIR do que ele oferece. Detalhes, novamente, no portal do FHIR em [CapabilityStatement](#).

A descrição de um recurso no portal do FHIR conta com vários elementos de apoio. Inclui uma breve descrição do conteúdo, o escopo e uso esperado do recurso, o contexto e o conteúdo propriamente dito do recurso, ou seja, os itens de informação, à luz do FHIR, necessários para caracterizar, nesse caso, o que um servidor FHIR oferece.

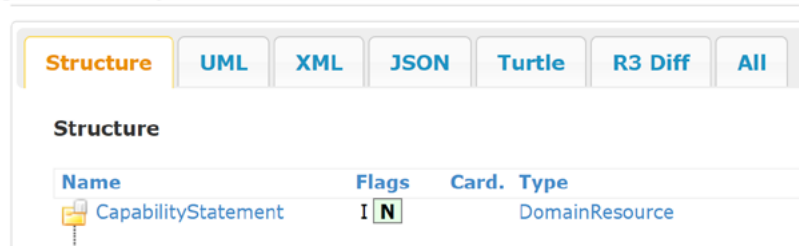
Essas não são as únicas informações contidas na página de descrição de um recurso. E tudo se repete para todos os recursos FHIR (mais de uma centena deles).




Formas de definição do conteúdo de um recurso FHIR

A documentação precisa de cada item de informação que caracteriza um recurso, fornecida por meio de várias “visões” distintas, conforme a ilustração abaixo, obtida de um recorte da definição do recurso [CapabilityStatement](#).

Figura 34. Recorte da definição do recurso *CapabilityStatement*



Name	Flags	Card.	Type
 CapabilityStatement	I N		DomainResource

Fonte: autoria própria.

Da esquerda para a direita, a estrutura (*Structure*) é a primeira delas, ela fornece uma perspectiva mais próxima de um modelo de informação. Provavelmente, é a mais consultada das alternativas. Na figura acima é possível identificar os atributos *Name*, *Flags*, *Card.* (cardinalidade) e *Type*. Observe que para o atributo *Name* o valor correspondente é *CapabilityStatement*.

A posterior é a UML, amplamente empregada na construção de modelos de *software*. Por si só já demonstra aproximação com o modelo computacional, o que, definitivamente, é o caso das opções seguintes, na ordem, XML, JSON e *Turtle*.

Estas três últimas são empregadas para “ilustrar e definir”, ao mesmo tempo, o que é necessário para registrar um recurso.

Formatos de registro de um recurso

Existem inúmeros mecanismos para serialização de dados estruturados, [Protocol Buffers](#) é um deles. Contudo, o padrão FHIR adotou outros três, a saber, XML, JSON e *Turtle*. Isso significa que um servidor FHIR recebe e fornece informações usando pelo menos um desses formatos.

Quando um laboratório notifica um Resultado de Exame Laboratorial (REL) ao Ministério da Saúde, via RNDS, isto significa que o resultado em questão, em conformidade com os recursos pertinentes, foi serializado em um destes três formatos (de fato, apenas XML ou JSON, dado que os servidores FHIR da RNDS não estão habilitados a empregar o formato *Turtle*, conforme visto adiante).

A serialização é o processo que produz, como resultado, um formato que pode ser empregado para transferência (caso da RNDS) ou até mesmo para o armazenamento da informação em questão.

No sentido inverso, o cenário é semelhante. Ou seja, informações em saúde podem ser armazenadas pela RNDS de várias formas e, mesmo que o formato empregado para armazenamento não seja conhecido, os servidores da RNDS irão entregar como resposta informações nos formatos XML ou JSON. JSON é empregado por padrão.

Observe que o formato *Turtle* não faz parte das opções. De fato, o recurso *CapabilityStatement* inclui, conforme ilustra o excerto abaixo, XML e JSON. Anteriormente foi ilustrado como pode ser recuperado o *CapabilityStatement* dos servidores FHIR da RNDS.

```
“format”: [ “application/fhir+xml”, “application/fhir+json” ]
```



Apenas para ilustrar, o comando abaixo recupera o *CapabilityStatement* em formato XML, em vez do formato JSON (valor *default*).

```
curl -o go.xml https://go-ehr-services.saude.gov.br/api/fhir/r4/metadata?_format=xml
```

Este comando, ao contrário do anterior, recupera um *CapabilityStatement* do servidor em questão no formato XML, daí o resultado ser registrado no arquivo **go.xml**.

Versão FHIR e outras informações

O excerto abaixo do *CapabilityStatement*, recuperado conforme ilustrado anteriormente, contém várias propriedades.

A data (*date*) em que o recurso foi recuperado, o *software* (*software*) que responde a requisição, ou seja, implementa o servidor FHIR em questão, neste caso na versão (*version*) **4.2.0**. A implementação (*implementation*) empregada pelo servidor é a [HAPI FHIR](#), uma implementação *open source* do FHIR. Outra propriedade relevante é a versão do FHIR oferecida por esta implementação, indicada pela propriedade *fhirVersion*, conforme abaixo, **4.0.1**.

```
“resourceType”: “CapabilityStatement”,
“status”: “active”,
“date”: “2021-02-18T08:23:05-03:00”,
“publisher”: “Not provided”,
“kind”: “instance”,
“software”: {
  “name”: “RNDS FHIR R4 PROD Server - GO”,
  “version”: “4.2.0”
},
“implementation”: {
  “description”: “HAPI FHIR”,
  “url”: “https://go-ehr-services.saude.gov.br/1.15/api/fhir/r4”
},
“fhirVersion”: “4.0.1”,
“format”: [
  “application/fhir+xml”,
  “application/fhir+json”
]
```

Este excerto também é útil para ressaltar outra propriedade empregada em todos os recursos FHIR, a propriedade *resourceType*. Conforme o nome indica, é por esta propriedade que é informado o recurso em questão, neste caso, *CapabilityStatement*.

Embora relevante para a compreensão de um recurso FHIR qualquer, acima é fornecido apenas um fragmento de um *CapabilityStatement*. Para este recurso em particular, demais elementos foram omitidos porque a explanação deles é melhor compreendida no contexto da “FHIR API” (pág. 100).

Autenticação

O serviço de autenticação, identificado por *Auth*, é o serviço por meio do qual o *token* exigido para acesso aos serviços da RNDS é recuperado. A utilização deste serviço é detalhada em “Casos de uso” (pág. 55), em particular, no caso de uso *Obter token de acesso*. Este caso de uso depende da identidade digital do estabelecimento de saúde (Certificado Digital) e, quando o serviço é acionado, a resposta é um JSON cuja propriedade *access_token* indica o *token* de acesso.



Informações em saúde

Informações em saúde incluem REL, Sumário de Alta (SA) e Registro de Atendimento Clínico (RAC), para citar apenas três exemplos. Ao longo do tempo o conjunto contemplado pela RNDS será estendido. Abaixo é detalhado o primeiro elemento desse conjunto, o REL.

Compreender a definição de um REL é um exercício útil não apenas a integradores de laboratórios de análises clínicas. Tal compreensão demanda a interpretação de estruturas similares que serão empregados por quaisquer outras necessidades de troca de informação.

Resultado de Exame Laboratorial

O objetivo do REL é promover o compartilhamento de resultado de exame laboratorial referente aos laudos de COVID-19, realizado em qualquer laboratório de análises clínicas, público ou privado, em todo o território nacional, permitindo a visualização desses resultados pelo profissional de saúde para garantir a continuidade do cuidado na atenção primária em saúde, bem como na atenção especializada ou internação. Adicionalmente, esses resultados também estarão acessíveis aos cidadãos.

Tanto o contexto quanto o modelo de informação, documentados nas seções seguintes, não são conteúdos criados como parte deste texto. Em vez disso, tais seções foram obtidas da [Portaria GM/MS nº 1.068, de 17 de novembro de 2020](#). Siga este *link* para obter a versão original do portal da Imprensa Nacional.

Convém mencionar que este resultado foi definido exclusivamente para o caso específico de exame de SARS-CoV-2, ou seja, qualquer outro exame não está contemplado pelos modelos de informação e computacional fornecidos.

Contexto

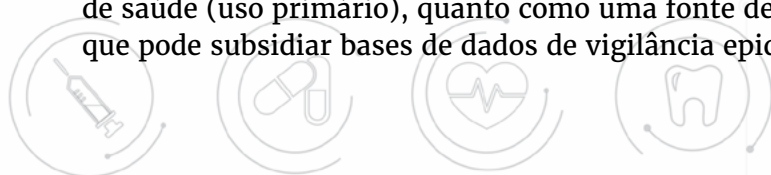
Em dezembro de 2019, na província de Wuhan (China), foi identificada uma nova cepa de coronavírus, reportada por causar sintomas de pneumonia em humanos e cuja origem permanece desconhecida até o momento.

Esse vírus, denominado SARS-CoV-2, mostrou-se ser altamente transmissível, e registros da sua presença foram crescentes pelo mundo até a declaração de pandemia mundial pela Organização Mundial da Saúde (OMS) em 11/03/2020.

COVID-19 é a denominação da doença causada pelo SARS-CoV-2, caracterizada por uma infecção respiratória que pode ser assintomática, com sintomas leves a moderados, mas também causar síndrome respiratória aguda grave e insuficiência renal em casos graves e críticos. Essa doença tem mobilizado autoridades mundiais na prevenção de sua transmissibilidade, na otimização da assistência à saúde e na melhoria da gestão e coordenação dos serviços, de modo a evitar um colapso dos sistemas de saúde e permitir acesso dos indivíduos ao cuidado assistencial.

Considerando a Portaria GM/MS nº 1.792, de 17 de julho de 2020 que altera a Portaria GM/MS nº 356, de 11 de março de 2020, para dispor sobre a obrigatoriedade de notificação ao Ministério da Saúde de todos os resultados de testes diagnóstico para SARS-CoV-2 realizados por laboratórios da rede pública, rede privada, universitários e quaisquer outros, em todo território nacional tornando obrigatório aos laboratórios clínicos enviarem os resultados dos exames para detecção da COVID-19 ao Ministério da Saúde, por meio da RNDS.

A RNDS vem ao encontro dessa questão, contribuindo tanto na continuidade do cuidado em saúde, pela disponibilização da informação em plataformas de acesso direto pelo cidadão e profissionais de saúde (uso primário), quanto como uma fonte de informação para uso secundário dos dados, que pode subsidiar bases de dados de vigilância epidemiológica do Ministério da Saúde e Secre-



tarias Estaduais e Municipais de Saúde, além de munir os gestores do Brasil com informações relevantes ao enfrentamento à pandemia.

Trata-se de uma plataforma nacional de integração de dados em saúde, sendo um projeto estruturante do ConecteSUS para a transformação digital da saúde no Brasil. Contém um repositório de documentos responsável por armazenar informações de saúde dos cidadãos, mantendo a privacidade, integridade e auditabilidade dos dados e promovendo a acessibilidade e interoperabilidade das informações de forma segura e controlada [4].

Modelo de informação

O modelo de informação é uma representação conceitual e canônica, na qual os elementos referentes a um documento específico são modelados em seções e blocos de dados, com seus respectivos tipos de dados a serem informados. Também são informadas as referências para o uso de recursos terminológicos.

Abaixo, segue a legenda que define as colunas da Tabela 3:

Coluna 1 - Nível: apresenta o nível do elemento no modelo de informação;

Coluna 2 - Ocorrência: descreve o número de vezes que o elemento deve/pode aparecer, onde:

- [0..1] - o elemento é opcional e, se ocorrer, aparece uma vez;
- [1..1] - o elemento é obrigatório e deve estar presente uma única vez;
- [0..N] - o elemento é opcional e pode ocorrer várias vezes;
- [1..N] - o elemento é obrigatório e pode ocorrer várias vezes;

Coluna 3 - Elemento: apresenta o elemento a ser informado;

Coluna 4 - Descrição/Regras: apresenta o conceito e ou a regra referente ao elemento;

Coluna 5 - Tipo de dado: descreve o tipo de dado a ser preenchido, e

Coluna 6 - Dado: apresenta o(s) dado(s) possível(is), conforme o tipo de dado descrito na coluna anterior.

Tabela 3. Modelo de informação do Resultado de Exame Laboratorial COVID-19

Nível	Ocorrência	Elemento	Descrição/regras	Tipo de dado	Dado
1	1..1	Laboratório			
2	1..1	Nome do laboratório	Nome do estabelecimento de saúde responsável pelo resultado do exame laboratorial.	Texto	
2	1..1	CNES	Número do Cadastro Nacional do Estabelecimentos de Saúde do laboratório responsável pelo resultado do exame laboratorial.		
2	0..1	CNPJ			
2	0..1	Responsável técnico			
3	1..1	Nome completo do profissional	Nome completo do responsável técnico pelo laboratório.	Texto	
3	1..1	Conselho de classe profissional			



Nível	Ocor- rência	Elemento	Descrição/regras	Tipo de dado	Dado
4	1..1	Tipo de conselho	Conselho de classe profissional do responsável técnico pelo laboratório.	Texto codificado	CRM, CRF, CRBM, CRBIO, CRQ
4	1..1	Unidade Federativa	Unidade Federativa do conselho de classe profissional do responsável técnico pelo laboratório.	Texto	
4	1..1	Número do registro	Número do registro no conselho de classe profissional do responsável técnico pelo laboratório.	Texto	
1	1..1	Identificação do indivíduo			
2	1..1	Nome completo	Nome completo do sujeito do exame laboratorial	Texto	
2	1..1	CNS	Número do Cartão Nacional de Saúde	Texto	
1	1..1	Resultado de exame laboratorial			
2	1..N	Nome do exame	Nome do exame a que foi submetida a amostra biológica. Terminologia externa LOINC.	Texto codificado	LOINC
3	0..1	Resultado qualitativo	Valor atribuído ao analito de acordo com o método de análise, de forma qualitativa.	Texto codificado	Detectável, Não detectável, Inconclusivo
3	0..1	Resultado quantitativo	Valor quantitativo do resultado do exame, expresso com unidades de medida	Quantidade	
3	1..1	Amostra	Amostra biológica, preparada ou não, que foi submetida ao exame laboratorial. Ex: “soro”, “plasma”, “sangue”. Terminologias externas FHIR v2-0487 e Tipo Amostra GAL.	Texto codificado	FHIR v2-0487 ou Tipo Amostra GAL
3	1..1	Método de análise	Método analítico utilizado para determinação do resultado do analito.	Texto	
3	1..1	Faixa de referência	Faixa de valores de resultado esperada para determinada população de indivíduos.	Texto	
3	1..1	Data e hora do resultado	Data hora na qual o resultado do exame laboratorial foi registrado.	Data/hora	
3	0..N	Nota	Narrativa adicional sobre o exame laboratorial.	Texto	

Fonte: Portaria GM/MS nº 1.068, de 17 de novembro de 2020.

Modelo computacional

Objetivo

Especificar os recursos FHIR (*resources*) utilizados no registro de um REL. Os recursos são identificados e detalhados por meio da representação JSON de um resultado. Ou seja, é para consumo de integradores (profissionais com habilidades em desenvolvimento de *software*). Gestores e outros profissionais não interessados em detalhes técnicos podem consultar o modelo de informação.



Orientação

O integrador deverá...

- Saber quais são as informações necessárias para montar um resultado de exame de SARS-CoV-2.
- Saber como as informações devem ser registrados no formato JSON compatível com o padrão FHIR e perfis nacionais.
- Ser capaz de montar um documento JSON para refletir o resultado de exame de COVID-19.

Material para consulta

A representação completa em formato JSON de um resultado de exame laboratorial pode ser obtida [aqui](#).

As partes de um resultado

O resultado de exame laboratorial, por exemplo, o resultado do exame de SARS-CoV-2, é definido por meio de um recurso [Composition](#), que referencia um recurso [Observation](#) que, por fim, faz uso de um terceiro recurso FHIR, o [Specimen](#).

Perfis

Todos esses três recursos que compõem um resultado não são usados exatamente conforme definidos, mas por “versões personalizadas para o contexto nacional”, ou perfis (*profiles*). As adaptações do FHIR para o contexto nacional, definidas pela RNDS, são:

- [Resultado de Exame Laboratorial](#) (perfil para [Composition](#)),
- [Diagnóstico em Laboratório Clínico](#) (perfil para [Observation](#)) e
- [Amostra Biológica](#) (perfil para [Specimen](#)).

Bundle (um quarto recurso para resultado de exame)

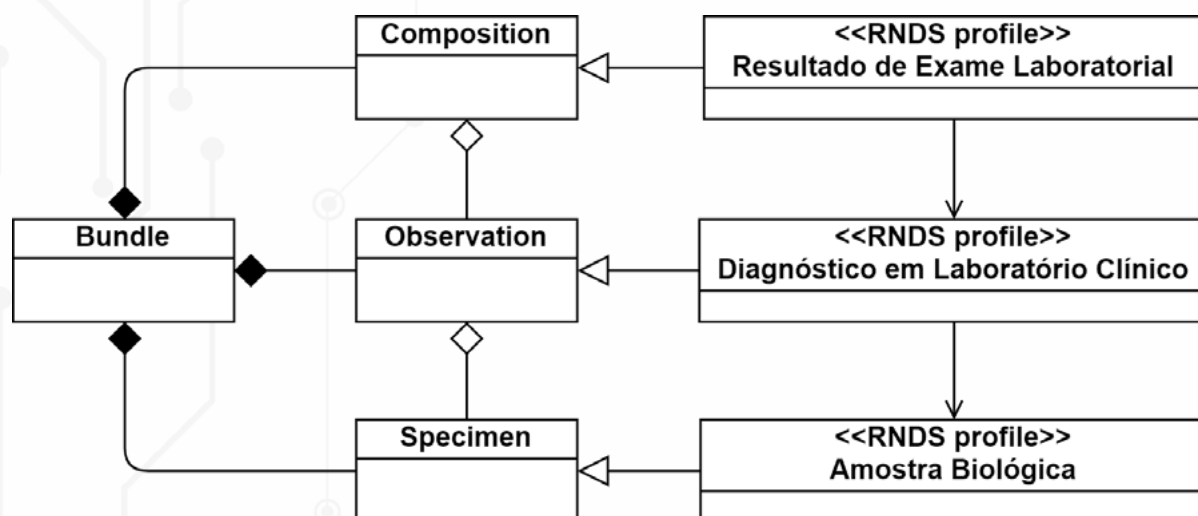
O FHIR faz uso extensivo do recurso [Bundle](#) com o propósito de reunir outros recursos. Esse uso resulta no quarto recurso necessário para compor um resultado de exame laboratorial, um recurso que agrupa os outros três comentados anteriormente.

O diagrama UML na Figura 35 esclarece que um *Bundle* é o “envelope” ou “pacote” no qual é depositado um resultado de exame laboratorial, que referencia um diagnóstico em laboratório clínico que, por sua vez, referencia a amostra biológica correspondente.

Todo recurso FHIR pode ser representado no formato JSON, conforme ilustrado no “esqueleto” de documento JSON abaixo. A propriedade *resourceType* é obrigatória e identifica o tipo de recurso. Para um recurso do tipo *Bundle* o valor desta propriedade é “*Bundle*”, conforme ilustrado.



Figura 35. Abstração para um *Bundle* (“envelope” ou “pacote”) no qual é depositado um resultado de exame laboratorial



Fonte: autoria própria.

```

{
  "resourceType": "Bundle",
  "type": "document",
  "timestamp": "2020-03-20T00:00:00-03:00",
  "meta": {
    "lastUpdated": "2020-03-20T00:00:00-03:00"
  },
  "identifier": { ... omitido ... },
  "entry": [
    { ... Resultado de Exame Laboratorial ... },
    { ... Diagnóstico em Laboratório Clínico ... },
    { ... Amostra Biológica ... }
  ]
}

```

A propriedade *type* indica o propósito do *Bundle*, no caso, trata-se de um documento (*document*). A propriedade *timestamp* indica o instante em que o *Bundle* foi criado, provavelmente o mesmo instante da última atualização, fornecida na propriedade *meta.lastUpdate*.

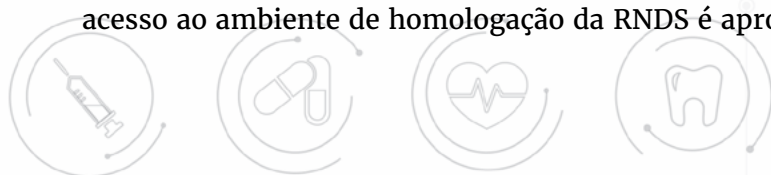
As outras duas propriedades do “esqueleto” são *identifier* e *entry*, justamente aquelas cujos valores foram omitidos, mas que são detalhadamente fornecidos nas seções seguintes.

Em particular, a propriedade *entry* é o motivo da existência de um *Bundle*. Exatamente por meio dessa propriedade são definidos os recursos FHIR que fazem parte do *Bundle*. Nesse caso, o [Resultado de Exame Laboratorial](#), o [Diagnóstico em Laboratório Clínico](#) e a [Amostra Biológica](#).

Quem desejar consultar o JSON completo, já “inflado” com os valores para essas propriedades, antes de percorrer as seções seguintes, pode obtê-lo [aqui](#).

Identificador do solicitante

O identificador do solicitante é um identificador fornecido pela RNDS quando a solicitação de acesso ao ambiente de homologação da RNDS é aprovada.



O identificador do solicitante é empregado na construção da identificação de uma requisição submetida para a RNDS, por exemplo, na composição do resultado de um exame laboratorial, conforme visto na seção seguinte.

Na Figura 36, é ilustrado o local onde o gestor pode localizar o identificador do solicitante no [portal de serviços](#). Tanto o número da solicitação de acesso quanto o identificador do solicitante foram ocultados nessa figura. O valor desejado, o identificador do solicitante, está ocultado pelo texto de fundo vermelho.

Figura 36. Local onde o gestor pode localizar o identificador do solicitante no Portal de Serviços



Convém ressaltar que não se trata do CNES do laboratório, mas de um identificador que será criado pela RNDS e atribuído ao laboratório.

Bundle (identifier)

O identificador (*identifier*) do *Bundle* é montado a partir de dois valores, definidos para as propriedades *system* e *value*. A propriedade *system* depende do identificador do solicitante, enquanto a propriedade *value* é definida pelo identificador único do resultado no escopo do laboratório em questão.

No trecho em formato JSON abaixo, esses dois valores são respectivamente representados por `{{lab-identificador}}` e `{{exame-id-lab}}`.

```
“identifier”: {  
  “system”: “http://www.saude.gov.br/fhir/r4/NamingSystem/BRRNDS-{{lab-identificador}}”,  
  “value”: “{{exame-id-lab}}”  
}
```

As chaves duplas são uma forma de referenciar variáveis na ferramenta *Postman*. Dessa forma, quando se usa tal ferramenta, pode-se definir valores para as variáveis **lab-identificador** e **exame-id-lab**, e a repercussão é o uso do valor correspondente onde quer que as variáveis sejam utilizadas.

O identificador do solicitante, representado acima por `{{lab-identificador}}`, é fornecido pela RNDS quando o pedido de solicitação de acesso à RNDS é aprovado. Consulte identificador do solicitante para detalhes.

O identificador do resultado de exame, por outro lado, é um identificador criado pelo laboratório para unicamente identificar o resultado em questão. Quaisquer dois resultados produzidos pelo laboratório devem, necessariamente, possuir identificadores distintos.



O laboratório pode optar por criar identificadores sequenciais, por exemplo, “1”, “2”, e assim por diante. Ou ainda, “2020-09-04-0001”, “2020-09-04-0002” e assim sucessivamente, para resultados produzidos em um determinado dia.

Também pode gerar um identificador universalmente único (*Universally Unique Identifier*) ou [UUID](#). Veja como podem ser gerados em [Java](#) e [JavaScript](#), por exemplo.

De posse tanto do identificador do solicitante, por exemplo, “99”, quanto do identificador de um resultado de exame a ser enviado para a RNDS, digamos “04/09/2020-cdYQj”, o trecho do JSON correspondente à propriedade *identifier* do *Bundle* a ser enviado para a RNDS, seria

```
“identifier”: {  
  “system”: “http://www.saude.gov.br/fhir/r4/NamingSystem/BRRNDS-99”,  
  “value”: “04/09/2020-cdYQj”  
}
```

Em consequência, o “esqueleto” JSON pode ser reescrito, considerando o preenchimento do *identifier*, conforme abaixo:

```
{  
  “resourceType”:”Bundle”,  
  “type”:”document”,  
  “timestamp”:”2020-03-20T00:00:00-03:00”,  
  “meta”: {  
    “lastUpdated”: “2020-03-20T00:00:00-03:00”  
  },  
  “identifier”: {  
    “system”: “http://www.saude.gov.br/fhir/r4/NamingSystem/BRRNDS-99”,  
    “value”: “04/09/2020-cdYQj”  
  },  
  “entry”:[  
    { ... Resultado de Exame Laboratorial ... },  
    { ... Diagnóstico em Laboratório Clínico ... },  
    { ... Amostra Biológica ... }  
  ]  
}
```

Bundle (entry)

Um *Bundle* é empregado para reunir recursos FHIR, e *entry*, destacada abaixo, é a propriedade onde os recursos devem ser fornecidos, observe que é um *array*. No caso em questão, este *array* deve possuir três entradas:

```
“entry”:[  
  { ... Resultado de Exame Laboratorial ... },  
  { ... Diagnóstico em Laboratório Clínico ... },  
  { ... Amostra Biológica ... }  
]
```

Essas três entradas, respectivamente, referem-se aos seguintes perfis definidos pela RNDS: [Resultado de Exame Laboratorial](#), [Diagnóstico em Laboratório Clínico](#) e [Amostra Biológica](#).



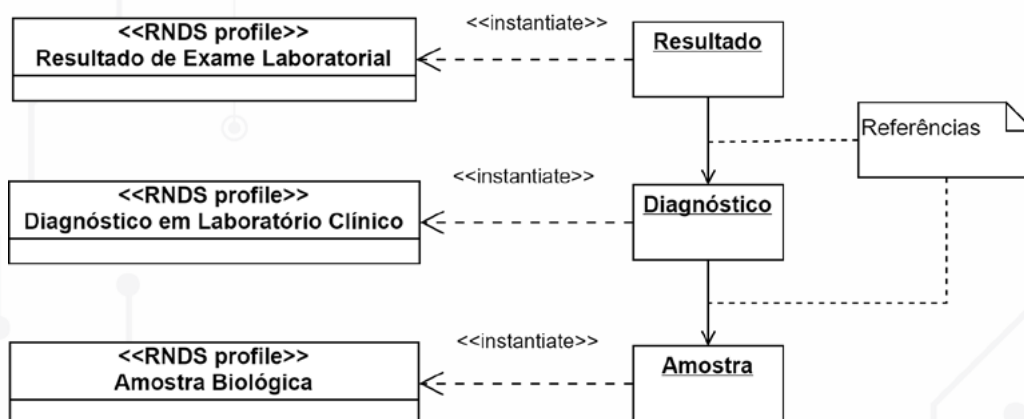
Observe que estes recursos são fornecidos em entradas próprias da propriedade *entry*, ou seja, não estão “aninhadas”, apesar da amostra biológica ser empregada pelo diagnóstico que, por sua vez, faz parte do resultado de exame laboratorial.

A “conexão” entre recursos é realizada por meio de referências, que são amplamente empregadas quando se usa o padrão FHIR.

Referências entre recursos

O FHIR faz uso extensivo do conceito de [referência](#) para permitir o registro de referências entre recursos. Por meio do emprego de referências é possível criar uma rede de informação em saúde.

Figura 37. Modelo de objetos UML para um resultado de exame laboratorial inclui três recursos, entre os quais há duas referências



Fonte: autoria própria.

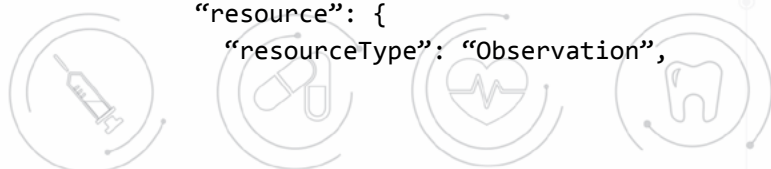
No contexto do REL, em vez de um diagnóstico em laboratório clínico (*Observation*) ser fornecido “embutido” em um resultado de exame laboratorial (*Composition*), esse último recurso referencia (aponta para) o recurso que é o diagnóstico (*Observation*).

Conforme ilustrado na Figura 37, o modelo de objetos UML para um resultado de exame laboratorial é formado por três recursos, entre os quais há duas referências. O recurso Resultado possui uma referência para Diagnóstico que, por sua vez, referencia o recurso Amostra.

A propriedade *entry* de um *Bundle* é fornecida abaixo para ilustrar o uso de referências entre recursos FHIR. Cada entrada do *array* possui o seu identificador único definido pela propriedade *fullUrl* (o recurso propriamente dito é apenas parcialmente fornecido), cada um com o seu próprio *resourceType*. A primeira entrada referencia a segunda entrada, dado que a propriedade *reference* indica o *fullUrl* da segunda entrada e, de forma similar, a segunda referencia a terceira.

```

“entry”: [
  {
    “fullUrl”: “urn:uuid:transient-0”,
    “resource”: {
      “resourceType”: “Composition”,
      ... “reference”: “urn:uuid:transient-1” ...
    },
  },
  {
    “fullUrl”: “urn:uuid:transient-1”,
    “resource”: {
      “resourceType”: “Observation”,
  
```



```

    ... "reference": "urn:uuid:transient-2" ...
  }
},
{
  "fullUrl": "urn:uuid:transient-2",
  "resource": {
    "resourceType": "Specimen"
    ... outras propriedades (mas não há 'reference')...
  }
}
]

```

Tendo em vista que os recursos que definem um resultado de exame laboratorial foram identificados, e que cada um deles é fornecido em entrada própria na propriedade *entry* (conforme ilustrado acima), e que a ligação entre eles é estabelecida por meio de referências, falta prosseguir e preencher cada um desses recursos. Novamente, o documento JSON completo está disponível [aqui](#).

Resultado de Exame Laboratorial (recurso)

Um resultado de exame no Brasil é definido pela RNDS por meio do perfil [Resultado de Exame Laboratorial](#). Este perfil é uma personalização do recurso [Composition](#). Conforme o perfil, um resultado é caracterizado por várias propriedades, cada uma delas comentada abaixo.

status. Identifica um dos valores do [Estado da Observação](#). São dois valores possíveis: “final” e “entered-in-error”. Nesse caso, o valor correto é “final”, que indica que o documento está concluído. A representação no formato JSON é fornecida abaixo:

```
“status”: “final”
```

type. Identifica o tipo do documento por meio da propriedade *coding*, que é um *array*, nesse caso, de uma entrada apenas e obrigatória. O objeto correspondente a tal entrada possui duas propriedades, *system* e *code*. A primeira define o conjunto de valores possíveis, nesse caso, o [Tipo de Documento](#). A segunda, um dos valores possíveis. Dentre eles há “REL”, que representa “Resultado de Exame(s) Laboratorial(is)”. Em consequência, esta propriedade é definida conforme abaixo:

```

“type”: {
  “coding”: [
    {
      “system”: “http://www.saude.gov.br/fhir/r4/CodeSystem/BRTipoDocumento”,
      “code”: “REL”
    }
  ]
}

```

subject. O indivíduo ao qual está associado o resultado de exame. A identificação é fornecida pelo objeto *identifier*, que possui duas propriedades, *system* e *value*. A primeira, *system*, possui um valor fixo. A segunda, *value*, é utilizada para fornecer o CNS (Cartão Nacional de Saúde) do indivíduo. Abaixo segue a representação desta propriedade na qual, em vez do CNS de um indivíduo, é fornecida a sequência “{{indivíduo-cns}}”.

```

“subject”: {
  “identifier”: {
    “system”: “http://www.saude.gov.br/fhir/r4/StructureDefinition/BRIndividuo-1.0”,
    “value”: “{{indivíduo-cns}}”
  }
}

```



date. Data e hora em que o documento foi gerado, por exemplo:

```
“date” : “2020-03-20T00:00:00-03:00”
```

author. Identifica a pessoa física (profissional liberal) ou a pessoa jurídica responsável por gerar o documento. A estrutura desse objeto é similar àquela de *subject*, fornecida acima. A sugestão é fornecer o estabelecimento de saúde e, em consequência, a definição correspondente é fornecida abaixo, onde a sequência “`{{lab-cnes}}`” deve ser substituída pelo CNES do estabelecimento em questão.

```
“author”:[
  {
    “identifier”:{
      “system”:"http://www.saude.gov.br/fhir/r4/StructureDefinition/BREstabelecimentoSaude-1.0",
      “value”:"{{lab-cnes}}”
    }
  }
]
```

title. O título do documento é o valor fixo “Resultado de Exame Laboratorial”.

```
“title”: “Resultado de Exame Laboratorial”
```

section. Define as seções empregadas pelo resultado (documento). Neste caso há uma única seção na qual é registrado o Diagnóstico em Laboratório Clínico. Ou seja, a única seção é um recurso FHIR, um *Observation* e, para ser ainda mais preciso, o perfil definido pela RNDS para registrar o diagnóstico de um laboratório clínico. A indicação da entrada do *Bundle* que contém o diagnóstico é fornecida abaixo:

```
“section”: [
  {
    “entry”: [
      {
        “reference”: “urn:uuid:transient-1”
      }
    ]
  }
]
```

Diagnóstico em Laboratório Clínico (recurso)

O perfil Diagnóstico em Laboratório Clínico detalha um exame ou teste realizado em laboratório com finalidade diagnóstica ou investigativa. Esse perfil é uma personalização do recurso *Observation*. As propriedades são definidas abaixo.

status. Define o Estado da Observação. São dois valores possíveis: “final” e “entered-in-error”. Nesse caso, o valor correto é “final”, para indicar que o diagnóstico está concluído. A representação JSON correspondente é fornecida a abaixo:

```
“status”: “final”
```

category. Classifica o exame ou teste utilizando um Subgrupo da Tabela SUS. Se o diagnóstico é por teste rápido, então o código correspondente é “0214”. Ou seja, a propriedade *category* para “Diagnóstico por teste rápido” é definida conforme a seguir:



```

“category”: [
  {
    “coding”: [
      {
        “system”: “http://www.saude.gov.br/fhir/r4/CodeSystem/BRSubgrupoTabelaSUS”,
        “code”: “0214”
      }
    ]
  }
]

```

code. Identifica o exame ou teste. Os valores são obtidos de códigos de Nome do Exame, que é formada pela união dos valores fornecidos na tabela Exames LOINC e códigos da tabela Exames do GAL. O trecho abaixo ilustra um exame identificado pelo código LOINC correspondente, nesse caso, “94507-1”, que designa “SARS Coronavírus 2, [dosagem de] anticorpo IgG em soro ou plasma por imunoenensaio”.

```

“code”: {
  “coding”: [
    {
      “system”: “http://www.saude.gov.br/fhir/r4/CodeSystem/BRNomeExameLOINC”,
      “code”: “94507-1”
    }
  ]
}

```

subject. Identifica o indivíduo associado ao exame ou teste. O valor é o mesmo daquele fornecido anteriormente, para o Diagnóstico em Laboratório Clínico. Em consequência, o trecho JSON correspondente, também com o mesmo propósito de não citar explicitamente um indivíduo, substitui o código CNS do indivíduo por “{{individuo-cns}}”.

```

“subject”: {
  “identifier”: {
    “system”: “http://www.saude.gov.br/fhir/r4/StructureDefinition/BRIndividuo-1.0”,
    “value”: “{{individuo-cns}}”
  }
}

```

issued. Data/hora em que o resultado foi liberado. Esse instante pode ser diferente daquele em que o resultado é produzido e também diferente do instante em que o *Bundle* foi produzido. Um exemplo é fornecido abaixo:

```

“issued”: “2020-09-10T10:49:10-03:00”

```

performer. Identifica o profissional e/ou estabelecimento de saúde responsável pelo resultado do exame. Abaixo, a sequência {{individuo-cns}} deve ser substituída pelo CNS do profissional responsável pelo resultado (laudo).

```

“performer”: [
  {
    “identifier”: {
      “system”: “http://www.saude.gov.br/fhir/r4/StructureDefinition/BRPessoaJuridicaProfissionalLiberal-1.0”,
      “value”: “{{individuo-cns}}”
    }
  }
]

```



O resultado propriamente dito de um exame é fornecido por apenas uma de duas propriedades possíveis. Se o resultado é quantitativo, então a propriedade que registra o resultado é *valueQuantity*. Se o resultado, por outro lado, é qualitativo, então a propriedade empregada é *valueCodeableConcept*.

valueQuantity. Use essa propriedade quando o valor do resultado é quantitativo (*Quantity*).

valueCodeableConcept. Use essa propriedade quando o valor do resultado é qualitativo (*CodeableConcept*).

No trecho JSON abaixo o resultado é qualitativo e o código obtido da tabela Resultado qualitativo do Exame, na qual 1 representa “Detectável”, 2 representa “Não detectável” e 3 representa “Inconclusivo”.

```
“valueCodeableConcept”: {
  “coding”: [
    {
      “system”: “http://www.saude.gov.br/fhir/r4/CodeSystem/BRResultadoQualitativoExame”,
      “code”: “3”
    }
  ]
}
```

interpretation. Interpretação qualitativa de um resultado quantitativo. Propriedade opcional e particularmente útil quando se deseja esclarecer o resultado quantitativo do exame.

note. Comentários sobre os resultados dos exames. Propriedade também opcional. Várias anotações, conforme ilustrado abaixo, podem ser fornecidas.

```
“note”: [ { “text”: “anotação 1” }, { “text”: “anotação 2” } ]
```

method. O método empregado na realização do exame. Esse objeto possui uma única propriedade, *text*. Abaixo segue um trecho JSON correspondente:

```
“method”: {
  “text”: “Imunocromatográfico”
}
```

referenceRange. De forma análoga à *method* (acima), para o perfil em questão, essa propriedade (*array*) é definida por um único objeto de uma única propriedade, *text*, conforme ilustra o trecho abaixo.

```
“referenceRange”: [
  {
    “text”: “(1) Detectável=presença de anticorpos; (2) Não detectável=ausência de anticorpos”
  }
]
```

specimen. Estabelece referência para o recurso *Specimen*. Este recurso identifica a amostra empregada na realização do exame. Neste caso específico, é uma referência para o perfil Amostra Biológica.

Em tempo, a propriedade *specimen* é a última do segundo recurso fornecido no *Bundle* e que, em particular, referencia o terceiro recurso, conforme ilustrado a seguir.



```

“specimen”: {
  “reference”: “urn:uuid:transient-2”
}

```

Amostra Biológica (recurso)

[Amostra Biológica](#) é um perfil de [Specimen](#). Esse perfil identifica a amostra de origem humana ou animal usada em investigações biológicas/laboratoriais para fins diagnósticos.

Nesse perfil, a única propriedade é *type*, que identifica o [Tipo de Amostra de Exame](#). Este tipo pode vir de duas tabelas (*code systems*) distintas. Uma delas é o [Tipo de Amostra Biológica](#) que, dentre os seus códigos se encontra “SGHEM” para designar “sangue”. Ou seja, se a amostra biológica empregada pelo laboratório foi sangue, então o trecho JSON correspondente é fornecido abaixo:

```

“type”: {
  “coding”: [
    {
      “system”: “http://www.saude.gov.br/fhir/r4/CodeSystem/BRTipoAmostraGAL”,
      “code”: “SGHEM”
    }
  ]
}

```

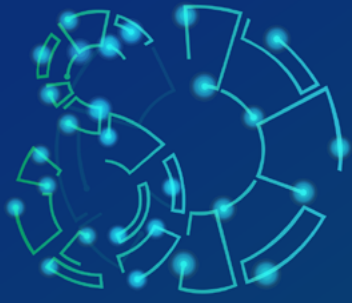
Embora a única propriedade do perfil seja *type*, ilustrado acima, todo recurso tem a propriedade *meta*, assim como *resourceType*. Tendo em vista que o recurso em questão é uma das entradas de um *Bundle*, toda a entrada é exibida abaixo:

```

{
  “fullUrl”: “urn:uuid:transient-2”,
  “resource”: {
    “resourceType”: “Specimen”,
    “meta”: {
      “profile”: [
        “http://www.saude.gov.br/fhir/r4/StructureDefinition/BRAmostraBiologica-1.0”
      ]
    },
    “type”: {
      “coding”: [
        {
          “system”: “http://www.saude.gov.br/fhir/r4/CodeSystem/BRTipoAmostraGAL”,
          “code”: “SGHEM”
        }
      ]
    }
  }
}

```





EDUCAÇÃO E CAPACITAÇÃO
DE RECURSOS HUMANOS
EM **SAÚDE DIGITAL**

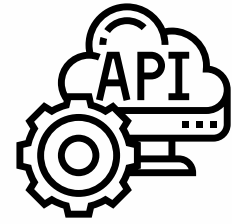
Integração com a Rede Nacional de Dados em Saúde - RNDS

Unidade 9 **FHIR API** (conexão)

Fábio Nogueira de Lucena
Plínio de Sá Leitão Júnior



FHIR API



Um dos principais componentes do padrão FHIR é a [RESTful API FHIR](http://hl7.org/fhir/http.html), uma API (*Application Programming Interface*) bem-definida que ordena a interação entre dois sistemas de *software* para a troca de informação em saúde. São vários os detalhes e o que segue destaca itens importantes, não é viável repetir e comentar todos eles e, dado que são fundamentais, convém reiterar o portal com a documentação oficial:

<http://hl7.org/fhir/http.html>

RESTful API é um estilo empregado para integração entre sistemas de *software*. Há extenso uso desse estilo, não é uma exclusividade da saúde, o que é evidenciado pelo conjunto significativo de serviços disponíveis que o emprega (veja uma lista deles [aqui](#)).

Nesta Unidade, o objetivo é conhecer detalhes da *RESTful API FHIR* pela compreensão dela no contexto de uso da RNDS. Isto é feito pela apresentação dos elementos que compõem uma requisição dirigida a um servidor FHIR e pela experiência de envio de requisições via ferramenta *Postman*. *Postman* não é uma ferramenta obrigatória para a integração com a RNDS, é uma ferramenta usada para conhecer e compreender a interação com um servidor FHIR (Videoaula 10).

Videoaula 10: FHIR API



Fonte: autoria própria.



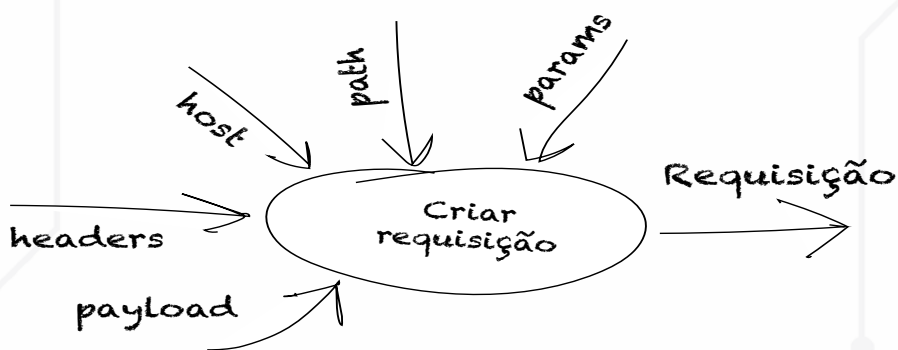
Convém informar que as requisições disponíveis estão devidamente [documentadas](#) e acessíveis pelos *web services* oferecidos pelos [ambientes](#) da RNDS. A experimentação com o ambiente de homologação da RNDS é desejável, ele foi feito para que “últimos detalhes” possam ser verificados e no contexto mais próximo possível do ambiente de homologação. Contudo, a experimentação também pode ser realizada em um servidor de teste ou até um servidor local, conforme comentado em “Servidor FHIR” (pág. 71).

Requisição *https* para servidor FHIR

A conexão com a RNDS ocorre por meio de requisições *https* em conformidade com o padrão FHIR. As requisições devem partir da infraestrutura empregada pelo estabelecimento de saúde e atingir os *web services* da RNDS.

A montagem de uma requisição é função de vários insumos, conforme a Figura 38, e cada um deles é documentado na sequência. Convém mencionar que o *payload* foi tratado em “Entradas e Saídas” (pág. 82). Os demais, *host*, *path*, *params* e *headers*, são componentes do protocolo HTTP cujo uso está bem definido para a conformidade com o padrão FHIR.

Figura 38. Componentes do protocolo HTTP cujo uso está bem definido para a conformidade com o padrão FHIR.



Fonte: autoria própria.



Host

O *host* identifica o servidor FHIR para o qual a requisição será enviada. Pode ser um dos endereços dos ambientes de homologação ou produção da RNDS, ou um servidor FHIR empregado para testes, pode até mesmo ser uma instância em execução no seu computador local.

Em geral é definido pelo domínio (*domain name*) empregado pelo servidor FHIR. Por exemplo, **hapi.fhir.org** é o *host* do servidor de testes oferecido pela HAPI FHIR, enquanto no ambiente de produção para estabelecimentos de saúde do Paraná o *host* é **pr-ehr-services.saude.gov.br**.

Em vez do domínio pode ser empregado o real endereço, ou seja, o valor IP correspondente, por exemplo 35.229.94.143 para **hapi.fhir.org**, no momento em que este texto foi escrito.

Path

O *path* depende de um prefixo e do recurso FHIR pertinente à requisição. Cada servidor FHIR pode fazer uso de um prefixo. No caso da RNDS o prefixo é **api/fhir/r4** enquanto na instância de teste oferecida pela implementação HAPI FHIR o prefixo é **baseR4**. Ou seja, esse prefixo vai variar conforme a instância usada. Observe que, no caso da RNDS, fica evidente, do prefixo, a indicação de uma API (**api**), pertinente ao FHIR (**fhir**) e, por fim, a versão 4 (**r4**).

Também faz parte do *path* o recurso FHIR pertinente à requisição. Por exemplo, para localizar um estabelecimento de saúde é necessária uma requisição para o recurso *Organization* e, em consequência, para a RNDS, o *path* resultante é **api/fhir/r4/Organization**. Por outro lado, se a busca é por um usuário (paciente), então o recurso é *Patient* e o *path* será **api/fhir/r4/Patient**.

Params

Em uma consulta são naturais os parâmetros, por exemplo, o parâmetro *_format* pode ser empregado para indicar o formato esperado na resposta do servidor FHIR, conforme ilustrado abaixo:

```
curl -o go.xml https://go-ehr-services.saude.gov.br/api/fhir/r4/metadata?_format=xml
```

O comando acima usa o parâmetro *_format* para indicar que a resposta deve ser fornecida no formato XML, em vez do padrão JSON. Há um conjunto significativo de parâmetros, especificamente quando o assunto são as buscas em recursos.

Apresentar e comentar todos os parâmetros aqui é inviável, além de redundante. A sugestão é consultar a URL abaixo:

<https://www.hl7.org/fhir/http.html>

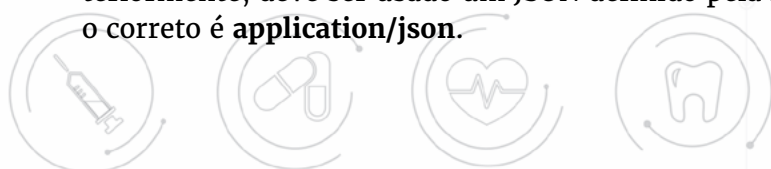
Headers (requisição)

Dois *headers* destacam-se dos demais por serem obrigatórias para as requisições dirigidas ao endereço *EHR*, são aqueles de segurança, documentados em “Segurança (*headers*)” (pág. 105).

O formato do conteúdo enviado (*Content-Type*), ou seja, requisições POST e PUT, variam. Se é enviado um recurso FHIR, então o valor deve ser **application/fhir+json** (para o formato JSON) ou **application/fhir+xml** (para o formato XML).

A RNDS não opera com o formato *Turtle*, conforme “Representação de recursos Fhir” (pág. 61), contudo, apenas para ilustrar, o formato seria **application/fhir+turtle**.

Observe que nem todas as requisições usam o conteúdo **application/fhir+json** ou **application/fhir+xml**, por exemplo, para recuperar o *token* de um contexto de atendimento, comentado posteriormente, deve ser usado um JSON definido pela RNDS, não é um recurso FHIR e, nesse caso, o correto é **application/json**.



Headers (resposta)

Acima foram identificados os *headers* empregados na composição de requisições. As respostas, contudo, também são acompanhadas de vários *headers*. Um deles é *Location*. Observe o sufixo deste *header* em uma resposta da RNDS para a submissão de Resultado de Exame Laboratorial (REL):

```
.../api/fhir/r4/Bundle/b0362dc0-bd1b-49da-8502-e1cbc1e44e69-r3x7
```

O valor completo não foi fornecido apenas para ressaltar o que interessa, em particular, o identificador criado pela RNDS para o resultado submetido. Esse identificador é o que segue após *Bundle/*.

Este *header* indica para o estabelecimento de saúde que a RNDS fará uso do valor retornado para identificar unicamente o resultado submetido, a saber:

```
b0362dc0-bd1b-49da-8502-e1cbc1e44e69-r3x7
```

O estabelecimento de saúde deve associar esse identificador fornecido pela RNDS e o identificador que faz uso internamente e guardá-los, para eventual consulta posterior.

A intenção aqui, contudo, não é fornecer orientações sobre o que o estabelecimento de saúde deve ou não fazer com informação em saúde assim como as respostas oferecidas pela RNDS. Em vez disso, a intenção é ilustrar que em uma resposta, não só o corpo da resposta, mas também por meio de *headers*, pode ser retornada informação relevante. Nesse caso, para o *header* denominado *location*, é retornado identificador único atribuído pela RNDS ao resultado de exame submetido.

Como me ambientar com esses detalhes?

Sim, o integrador precisa conhecer as requisições em todos os seus detalhes. Até porque estará criando código para submetê-las aos servidores da RNDS.

A estratégia sugerida para se familiarizar com as requisições a serem submetidas à RNDS é: (a) obtenha o *Postman*; (b) obtenha o *script* de configuração (já preparado para efetuar requisições para a RNDS); (c) configure o *Postman* com o *script* e outros detalhes e, por fim, (d) execute as requisições e seja criterioso ao observar o que está sendo enviado e o que é recebido como resposta. As seções seguintes comentam os passos desta sugestão.

Atenção

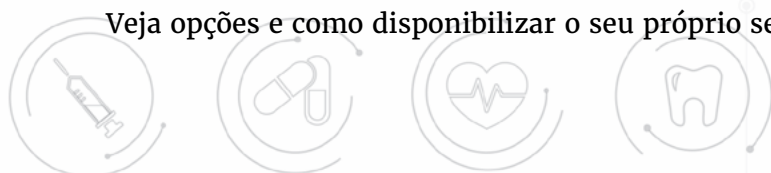
O ambiente de produção não deve ser empregado com a finalidade de experimentar a integração com os *web services* da RNDS.

Embora desejável o acesso ao ambiente de homologação para validar o que está sendo produzido, esse acesso pode não estar disponível, o que não impede a prática, outros servidores podem ser empregados, conforme “Servidor FHIR” (pág. 71).

Recomendação

*Empregue um servidor FHIR
publicamente disponível
para experimentação.*

Veja opções e como disponibilizar o seu próprio servidor em “Servidor FHIR” (pág. 71).



O que devo saber sobre a API?

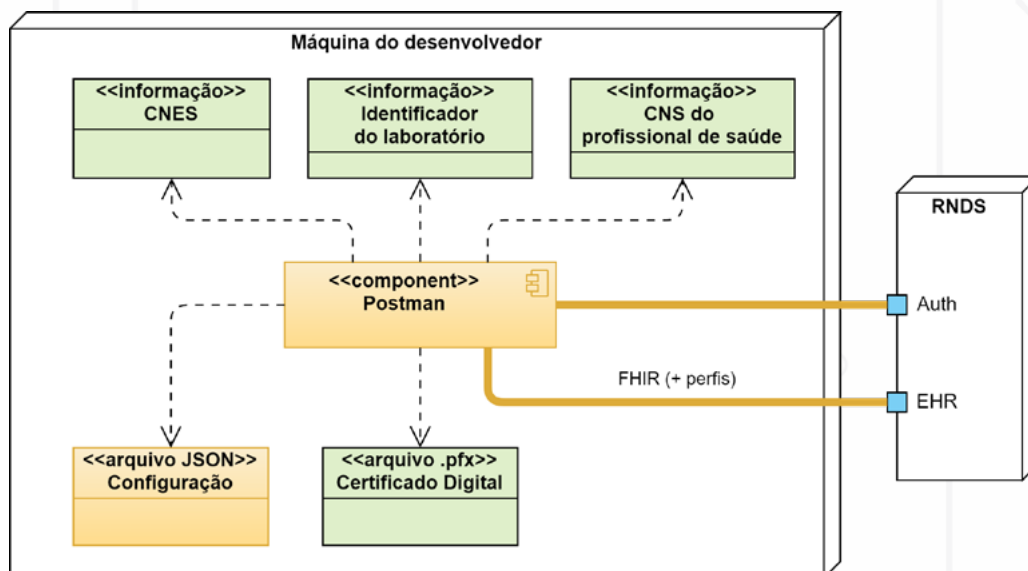
- As requisições disponíveis.
- Os *headers* necessários e como montar os valores correspondentes.
- A estrutura do *payload* para cada requisição, visto em “Entradas e Saídas” (pág. 82).
- Submeter as requisições, por exemplo, por meio da ferramenta Postman. Nesse caso, inclui a configuração dessa ferramenta com o uso do certificado digital, dentre outras.
- Interpretar as respostas.
- Experimentar valores diferentes para o *payload*, *headers* e parâmetros.
- O fluxo típico da interação com a RNDS usando HTTPS.
- Empregar o certificado digital para obter *token* desse acesso.

Caso siga, compreenda e tenha praticado cada um desses itens, terá adquirido parte relevante da habilidade necessária para construir o Conector.

Pré-requisitos

Conforme a Figura 39, a interação com a RNDS depende de várias informações e um arquivo, destacados em verde. Naturalmente, o Postman e seu arquivo de configuração não são necessários. Esses apenas viabilizam a conexão com a RNDS, conforme ilustrado.

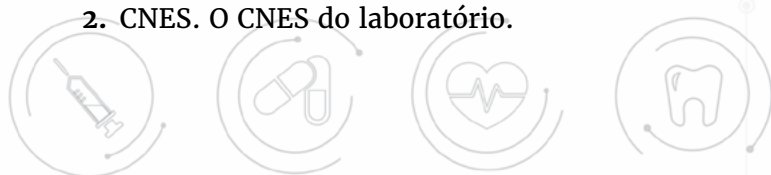
Figura 39. Pré-requisitos para interação com a Rede Nacional de Dados em Saúde



Fonte: autoria própria.

Cada uma das informações e arquivos são comentados abaixo:

1. Identificador do solicitante fornecido pela RNDS quando a solicitação de acesso é aprovada. Essa aprovação, além de definir o identificador do solicitante, também autoriza o acesso ao ambiente de homologação.
2. CNES. O CNES do laboratório.



3. CNS de um profissional de saúde lotado no estabelecimento de saúde cujo CNES é fornecido acima. O CNS indica em nome do quem as requisições ao ambiente de homologação serão feitas.
4. Certificado digital. O arquivo correspondente deve estar disponível, é um arquivo com a extensão .pfx, aqui será referenciado por certificado.pfx. Também será necessária a senha para acesso ao conteúdo do certificado.
5. Baixe o arquivo JSON de [configuração](#) do *Postman*, contendo exemplos de requisições prontas. Além das requisições este arquivo também contém a documentação de cada uma delas. A [documentação](#) das requisições também encontra-se amplamente disponível.

Segurança (headers)

Toda requisição ao endereço EHR (veja ambientes), faz uso de dois *headers* obrigatórios, aqui definidos como *headers* de segurança:

- **X-Authorization-Server:** esse é *header* por meio do qual o valor do *token* de acesso é fornecido. O valor deste *header* é definido pela concatenação de *Bearer* com o valor do *token*. Em consequência, para o *token* de valor “*token*”, o valor do *header* seria *Bearer token*. Convém ressaltar que o *token* de acesso é uma sequência de mais de 2000 caracteres e, portanto, bem mais extensa que o simples valor “*token*”.
- **Authorization:** neste *header* é identificado o profissional de saúde, lotado no estabelecimento de saúde em questão, em nome do qual a requisição é feita. O valor fornecido deve ser o CNS desse profissional.

Passos

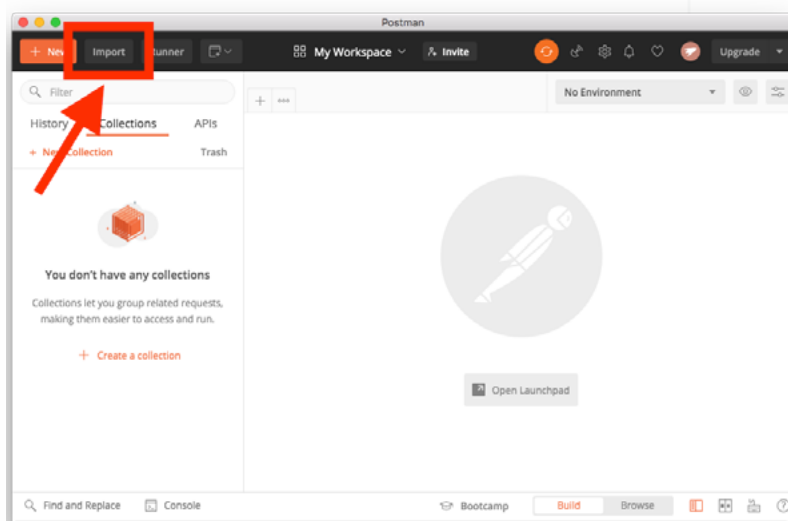
De posse das informações e dos arquivos, veja seção anterior, pode-se configurar o *Postman*: (a) importar *collection*; (b) configurar certificado digital e (c) configurar variáveis.

Importar collection

Veja o vídeo acerca de como importar [aqui](#).

Ao abrir o *Postman* você verá uma tela similar àquela na Figura 40, exceto que não terá o destaque para o botão *Import*, empregado para “importar” o arquivo baixado anteriormente:

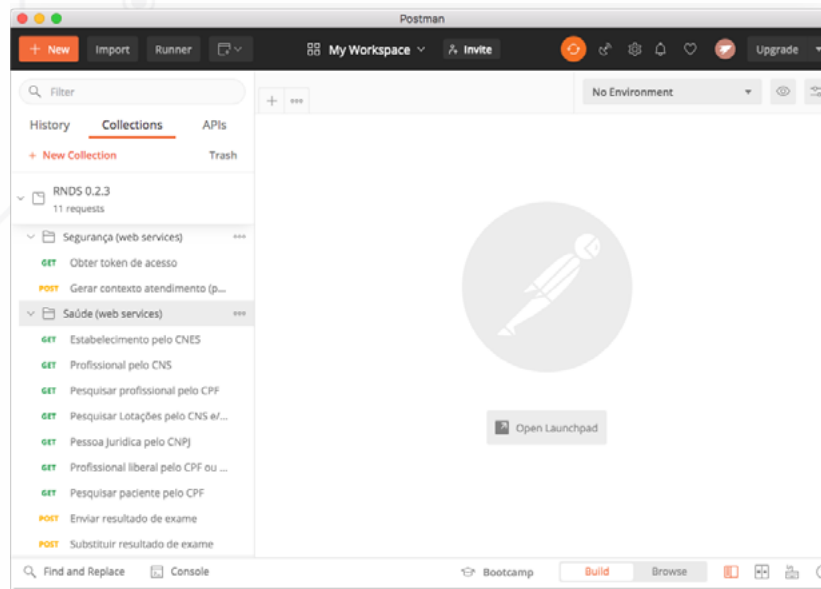
Figura 40. Configuração do importar collection



Fonte: autoria própria.

Após importado, estarão disponíveis várias requisições, agrupadas visando facilitar a localização, conforme a Figura 41. Nesse ponto, pode-se selecionar uma das requisições (barra lateral esquerda) e, no canto superior direito estará disponível o botão *Send*, que permite submeter aquela selecionada. Contudo, qualquer uma das requisições deve falhar, afinal, não foi configurado o certificado digital a ser empregado nem outras configurações necessárias. Observe que se estiver empregando um servidor de teste a requisição não irá falhar pela segurança (que não é exigida).

Figura 41. Requisições disponíveis



Fonte: autoria própria.

Configurando o uso do certificado digital

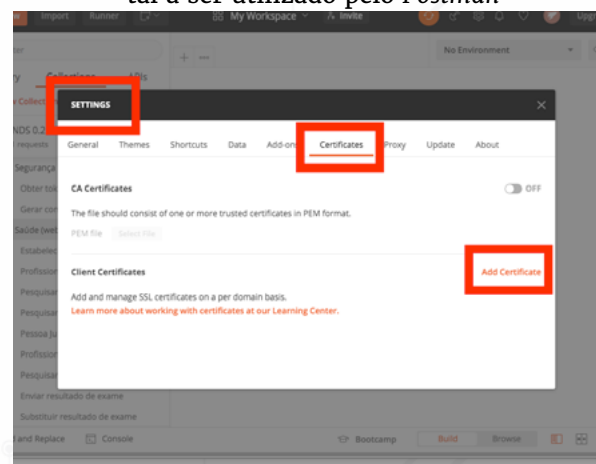
O *Postman* precisa ser configurado para usar o certificado digital do estabelecimento de saúde em questão, quando uma requisição for direcionada ao endereço empregado para autenticação do ambiente de homologação.

Essa configuração é exigida para a correta execução do serviço denominado “Obter token de acesso”. E o resultado desta requisição é necessário para a execução de todos os demais serviços. Em tempo, este é o único serviço que usa diretamente o certificado digital.

O serviço “Obter token de acesso” produz como resultado (retorno) o *token* de acesso. Tal *token* é exigido por todos os demais serviços. Ou seja, primeiro se obtém o *token* de acesso, que tem validade por 30 minutos, e depois ele é reutilizado, neste período, em todas as demais requisições. Transcorridos os 30 minutos, será necessária uma nova requisição ao serviço “Obter token de acesso”, para que um novo *token*, válido pelos próximos 30 minutos, possa ser reutilizado.

A indicação do certificado digital a ser utilizado pelo *Postman* é realizada da seguinte forma. Selecione *File* (opção do menu), na

Figura 42. Indicação do certificado digital a ser utilizado pelo *Postman*



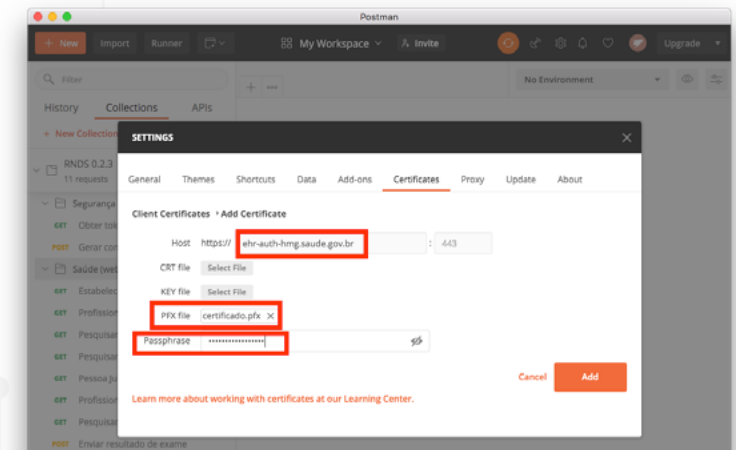
Fonte: autoria própria.



sequência a opção *Settings* e, por fim, abre-se a janela da Figura 42, na qual a aba *Certificates* deve ser selecionada e, por último, *Add Certificate*.

Quando *Add Certificate* é pressionado, abre-se tela similar à Figura 43. Observe que nenhum valor estará preenchido, ao contrário da tela exibida abaixo, na qual as três informações exigidas já estão fornecidas: (a) o domínio para o qual o certificado será utilizado pelo *Postman*, ou seja, o endereço Auth do ambiente de homologação (*ehr-auth-hmg.saude.gov.br*); (b) o arquivo *.pfx* contendo o certificado digital do estabelecimento de saúde e, por último, (c) a senha empregada para se ter acesso ao conteúdo do certificado.

Figura 43. Configurar Postman para usar o certificado



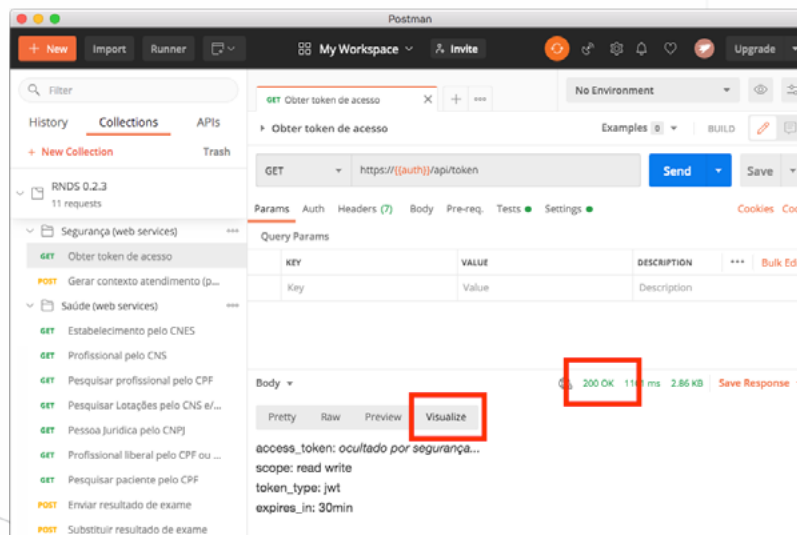
Fonte: autoria própria.

Ao clicar no botão *Add*, o *Postman* estará configurado para usar o certificado, para acesso ao endereço indicado e, para o uso, empregará a senha fornecida.

Após a configuração do certificado, quando se requisita “Obter token de acesso” (Figura 44), que está disponível exatamente no endereço fornecido na configuração acima, o certificado e a senha serão utilizados pelo *Postman* para submeter a requisição em questão. Agora, o resultado esperado é 200 OK. Observe que, logo abaixo, uma visualização (*visualize*) alternativa do retorno oferecido pela RNDS é exibida, na qual o *access_token* é ocultado. As demais informações não são sigilosas. Em particular, observe que o *token* tem uma validade de 30 minutos, ou seja, a intenção é que seja reutilizado neste período, conforme mencionado anteriormente.

As demais requisições dependem de outras configurações. Mais um passo e todas elas estarão funcionando.

Figura 44. Requisição para submissão do serviço “Obter token de acesso”



Fonte: autoria própria.

Configurar variáveis

A configuração do *Postman* para fazer uso do certificado digital viabiliza a execução da requisição “Obter token de acesso”. As demais, contudo, além do *token* retornado por esta requisição, dependem de outros valores depositados em variáveis. Abaixo seguem comentários para o conjunto das variáveis empregadas por uma das versões da coleção de requisições disponibilizada para uso do *Postman*. Observe que é esperado mudanças constantes ao longo do tempo.

São várias variáveis. Localize elas conforme ilustrado na Figura 45 e defina os valores correspondentes conforme o cenário em questão. Embora aqui seja empregada a configuração necessária para acesso ao Ambiente de Homologação, é possível que você empregue um outro servidor FHIR para teste.

As variáveis *uf*, *path*, *auth* e *ehr* são específicas para a confecção de requisições, em particular, a URL de cada requisição, o que é extensivamente coberto na “FHIR API” (pág. 100). Observe que algumas variáveis são empregadas apenas para oferecer um conjunto coerente e apto para verificação (testes).

Demais variáveis têm seus valores gerados pelo próprio *Postman* durante a execução das requisições. Por exemplo, a variável *access_token* é definida pela execução do serviço “Obter token de acesso” e, como anteriormente informado, o valor desta variável é empregado na composição do *header* de nome *X-Authorization-Server* empregado na composição de todas as demais requisições.

Variáveis específicas por estabelecimento de saúde (assim como o certificado digital):

- **lab-identificador:** identificador do laboratório fornecido pela RNDS quando o credenciamento é homologado. Observe que este identificador não é o CNES. Observe que o responsável pelo laboratório deverá acompanhar o pedido de credenciamento e, quando homologado, esse identificador estará disponível por meio do portal de serviços (o mesmo empregado para pedir o credenciamento). Veja identificador do solicitante para detalhes.
- **lab-cnes:** o código CNES do laboratório cujo credenciamento foi solicitado por meio do portal de serviços da RNDS e também aprovado. Naturalmente, o certificado digital empregado para configurar o *Postman* deve ser do laboratório em questão.
- **individuo-cns:** conforme o próprio nome indica, é o CNS de um indivíduo, em particular, o CNS do profissional de saúde em nome do qual requisições serão feitas. Ou seja, esse CNS deve estar associado ao laboratório em questão (CNES fornecido na variável acima). Esse valor será enviado para a RNDS por meio do *header* de nome *Authorization* em todos os contatos com a RNDS. A exceção é o serviço “Obter token de acesso”, que não faz uso desse *header*. Adicionalmente a esse uso, com o propósito de evitar a definição de outra variável, esse valor também é reutilizado para outras finalidades, por exemplo, para identificar o paciente de um exame.

Figura 46. Variável *auth* indica o endereço do *web service* empregado para autenticação

Figura 45. Variáveis específicas para esta coleta e requisição



VARIABLE	INITIAL VALUE
<input checked="" type="checkbox"/> <i>uf</i>	go
<input checked="" type="checkbox"/> <i>path</i>	api/fhir/r4
<input checked="" type="checkbox"/> <i>auth</i>	ehr-auth-hmg.saude.gov...
<input checked="" type="checkbox"/> <i>ehr</i>	ehr-services.hmg.saude...
<input checked="" type="checkbox"/> <i>individuo-cns</i>	00112233445566
<input checked="" type="checkbox"/> <i>lab-cnes</i>	1234567
<input checked="" type="checkbox"/> <i>identificador-solicitante</i>	99
<input checked="" type="checkbox"/> <i>ufg-cnpj</i>	01567601000143
<input checked="" type="checkbox"/> <i>access_token</i>	sequenciaDeCaracteres...
<input checked="" type="checkbox"/> <i>exame-id-lab</i>	8fead9e3-9b53-4189-b...
<input checked="" type="checkbox"/> <i>exame-id-rnds</i>	274206df-661c-475d-b...
<input checked="" type="checkbox"/> <i>individuo-cpf</i>	

Fonte: autoria própria.



Fonte: autoria própria.

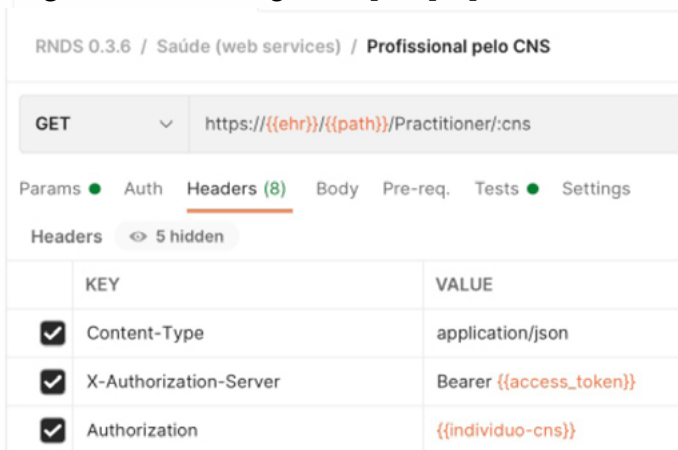
Outras variáveis

A variável *auth* deve indicar o endereço do *web service* empregado para autenticação. Esse valor é empregado na requisição “Obter token de acesso”, conforme ilustrado abaixo, como parte da URL correspondente (destaque na cor laranja)

Outra variável amplamente empregada é *ehr*. Nessa variável é esperado o endereço para envio das requisições de todos os demais serviços (*web services*), enquanto o valor da variável *auth* é empregado apenas para a requisição do serviço “Obter token de acesso”. Além dessa variável, várias outras são empregadas na Figura 47. Para a URL, além de *ehr* é empregada a variável **path**, enquanto dois *headers*, *X-Authorization-Server* e *Authorization*, empregam respectivamente as variáveis *access_token* e *individuo-cns*.

Variáveis geradas pelo próprio *Postman*:

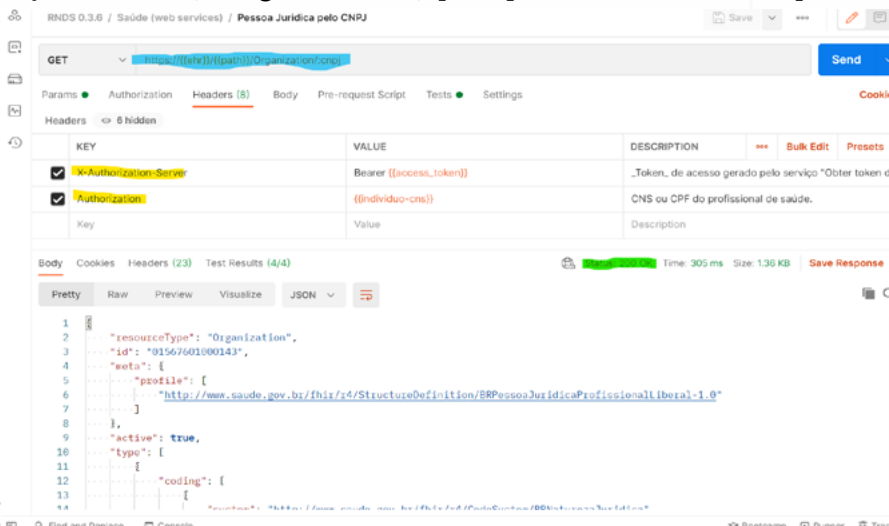
Figura 47. Variáveis geradas pelo próprio *Postman*



Fonte: autoria própria.

- **access_token**: gerada a partir da resposta para a requisição “Obter token de acesso”. Se executada de forma satisfatória, deposita nessa variável o token de acesso a ser consultado por todas as demais requisições.
- **exame-id-lab**: identificador gerado de forma aleatória para um resultado de exame laboratorial. Esse valor é gerado pela requisição “Enviar resultado de exame” e utilizado pela requisição “Substituir resultado de exame”.

Figura 48. Requisição após a submissão com destaque para a URL, os dois *headers* de segurança e o retorno, código HTTP 200, que representa sucesso e a resposta em JSON



Fonte: autoria própria.

- **exam-id-rnds**: identificador gerado pela própria RNDS para um resultado de exame submetido de forma satisfatória (requisição “Enviar resultado de exame”). Esse identificador único, gerado pela RNDS, é depositado nesta variável e, à semelhança de `exam-id-lab`, também é empregado pela requisição “Substituir resultado de exame”.

Conforme mencionado anteriormente, consulte a [documentação](#) da configuração do *Postman* para demais detalhes.

Experimentalizar requisições

A execução de requisições é feita com a seleção da requisição a ser executada e, em seguida, ao clicar no botão *Send*. A requisição será submetida e o retorno exibido. A sugestão é experimentar mudanças nos parâmetros das requisições, no *payload* de um resultado de exame, remover um *header*, alterar o valor de um *header* e observar os resultados. Dessa forma, será possível adquirir fluência na interação com a RNDS.

Abaixo, segue a ilustração de uma requisição após a submissão. Estão destacadas a URL, os dois *headers* de segurança e o retorno, código HTTP 200, que representa sucesso e a resposta em JSON.

Código que submete requisições

O *Postman* novamente ajuda aqui. Para cada requisição o *Postman* oferece várias opções para a correspondente execução em outros contextos, por exemplo, via linha de comandos como *curl* e *wget*, além de código em Java, C#, *JavaScript*, *Swift* e outras, perfazendo dezenas de alternativas.

Apenas por curiosidade, abaixo segue como executar uma das requisições oferecidas pela RNDS em *Ocaml*, sugerida pelo *Postman*:

```
open Lwt
open Cohttp
open Cohttp_lwt_unix

let reqBody =
  let uri = Uri.of_string "https://ehr-services.hmg.saude.gov.br/api/fhir/r4/Organization/01567601000143" in
  let headers = Header.init ()
    |> fun h -> Header.add h "Content-Type" "application/json"
    |> fun h -> Header.add h "X-Authorization-Server" "Bearer tokenAqui"
    |> fun h -> Header.add h "Authorization" "00112233445566"
  in
  Client.call ~headers `GET uri >>= fun (_resp, body) ->
  body |> Cohttp_lwt.Body.to_string >|= fun body -> body

let () =
  let respBody = Lwt_main.run reqBody in
  print_endline (respBody)
```

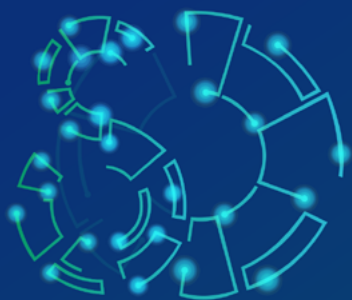
A exceção não adequadamente contemplada pelo *Postman* é a obtenção do token de acesso. Contudo, essa necessidade está devidamente desenvolvida, tanto para Java quanto *JavaScript*, e [documentada](#).

Cumprimentos feitos

Os “primeiros contatos” com um servidor FHIR, possivelmente do ambiente de homologação da RNDS foram estabelecidos. Seguramente, após exercitar as várias requisições, a ambientação necessária tanto com as informações exigidas para envio, quanto aquelas retornadas, e a estrutura da requisição, tem-se seguramente um passo concluído.

Essas atividades promovem a formação no desenvolvimento de parte significativa do código que interage com a RNDS. Embora, talvez, seja a parte mais “visível”, esse código tem outras funções, conforme a Unidade “Conector” (pág. 48).





EDUCAÇÃO E CAPACITAÇÃO
DE RECURSOS HUMANOS
EM **SAÚDE DIGITAL**

Integração com a Rede Nacional de Dados em Saúde - RNDS

Unidade 10 **Encerramento**

Fábio Nogueira de Lucena
Plínio de Sá Leitão Júnior
Renata Dutra Braga
Silvana de Lima Vieira dos Santos



Unidade de Encerramento

Chegamos ao final da cobertura de uma sequência considerável de tecnologias e informações pertinentes à desejada interoperabilidade em saúde no território nacional. Ao longo do caminho, foram fornecidas referências onde detalhes adicionais podem ser obtidos.

O contexto da integração, as resoluções correntes, profissionais envolvidos na interoperabilidade em saúde, inclusive a construção inicial do perfil de um integrador, o conhecimento, as ferramentas e o ambiente de trabalho deste desenvolvedor são elementos com foco no estabelecimento de saúde e, especificamente, no integrador.

O outro lado de uma integração, o lado do Ministério da Saúde, por meio do Datasus e, especificamente, a RNDS, também foi coberto. O processo de credenciamento, o detalhamento dos modelos de informação e computacional e, por fim, como fazer com que informações em saúde, em formato compatível com tais modelos, cheguem até o ambiente de produção da RNDS, por meio de uma *RESTful API*, foram apresentados. Em tempo, o modelo computacional e a API são diretamente decorrentes do padrão FHIR.

Você está orientado! É um passo necessário para que possa efetivamente conduzir ou participar de projetos de integração e, enfim, contribuir com a interoperabilidade em saúde no Brasil.



Referências

- [1] *Health Informatics on FHIR: How HL7's New API is Transforming Healthcare*, Mark L. Braunstein, Springer, 2018.
- [2] *Principles of Health Interoperability: FHIR, HL7 and SNOMED CT*, Tim Benson e Grahame Grieve, Fourth Edition, Springer, 2021.
- [3] BRASIL. MINISTÉRIO DA SAÚDE. Portaria GM/MS Nº 1.768, de 30 de julho de 2021. Altera o Anexo XLII da Portaria de Consolidação GM/MS Nº 2, de 28 de setembro de 2017, para dispor sobre a Política Nacional de Informação e Informática em Saúde (PNIIS). Acesso em 10 jul. 2022. Disponível em: <https://www.in.gov.br/en/web/dou/-/portaria-gm/ms-n-1.768-de-30-de-julho-de-2021-335472332>.
- [4] Comitê Gestor da Estratégia de Saúde Digital. Resolução CIT nº 46, de 29 de agosto de 2019. Disponível em: <https://saudedigital.saude.gov.br/governanca-e-gestao-da-esd/>. Acesso em 14 fev. 2021.
- [5] *Domain-Driven Design: Tackling Complexity in the Heart of Software*, Eric Evans, Addison-Wesley, 2003.
- [6] *Domain-Driven Design Distilled*, Vaughn Vernon, Addison-Wesley, 2016.



Glossário

Este *ebook* tem muitas palavras em inglês, por se tratar de termos técnicos.

Ambiente - a RNDS disponibiliza *web services* distribuídos em dois grupos, conforme o tipo de uso esperado. O ambiente de homologação reúne *web services* com foco na experimentação do Conector, enquanto o ambiente de produção está restrito àqueles cuja homologação foi aprovada.

API - sigla para *Application Programming Interface*. Uma API é o meio empregado por um *software* para se comunicar com outro *software*.

Auth - nome dado, por simples conveniência, ao endereço do *web service* da RNDS que atende pelas requisições de obtenção de token de acesso. *Auth* é prefixo de *Authentication*. Veja também EHR.

CapabilityStatement - recurso FHIR que registra a funcionalidade oferecida por um servidor FHIR.

CNES - sigla de Cadastro Nacional de Estabelecimentos de Saúde.

Caso de uso - sequência de interações entre, em geral, um usuário humano e um sistema de *software*. Também é útil para descrever interação entre sistemas de *software*, como empregado no presente material.

Certificado digital - identificação no mundo digital. No contexto do integrador, um certificado digital é empregado para identificar um estabelecimento de saúde junto à RNDS.

Conector - denominação do código produzido pelo integrador para interação com os *web services* da RNDS.

Context mapping - diagrama que identifica bounded-contexts e o relacionamento entre eles. Bounded-context é assunto pertinente a DDD.

Credenciamento - nome dado ao processo pelo qual um estabelecimento de saúde requisita e obtém acesso ao ambiente de produção da RNDS. O credenciamento é necessário para a integração com a RNDS.

DDD - sigla para *Domain-Driven Design*.

Domain-Driven Design - estratégia de projeto (*design*) de *software* na qual entidades são relacionadas formando um modelo de domínio, que aproxima o código do negócio em questão.

EHR - nome dado, por simples conveniência, ao endereço do *web service* da RNDS que atende requisições de troca de informação em saúde. É por este endereço que, de fato, a interoperabilidade em saúde ocorre. EHR é sigla de *Electronic Health Record*. Veja também *Auth*.

Engenharia de software - é a área do conhecimento pertinente ao desenvolvimento de *software*. Vários termos deste Glossário são pertinentes à engenharia de *software*. O leitor poderá obter definições formais destes termos, dentre outros, por meio do *Software Engineering Body of Knowledge* (SWEBOK), disponível gratuitamente em <https://www.computer.org/education/bodies-of-knowledge/software-engineering>.

Estabelecimento de saúde - usa-se esta denominação alinhada com o Cadastro Nacional de Estabelecimentos de Saúde (CNES) e a definição correspondente da Portaria 2.022 de 17 de agosto de 2017: “é o espaço físico delimitado e permanente onde são realizadas ações e serviços de saúde humana sob responsabilidade técnica.”



Estratégia de Saúde Digital para o Brasil – orienta ações no período de 2020 a 2028 pertinentes à saúde digital. Está alinhada à Política Nacional de Informação e Informática em Saúde (PNIIS).

FHIR – sigla para *Fast Healthcare Interoperability Resources*. FHIR é um padrão para troca de dados em saúde. Consulte <http://hl7.org/FHIR> para detalhes. A RNDS adotou o padrão FHIR para troca de informação em saúde, ou seja, o Conector, a ser produzido pelo integrador, deve fazer uso deste padrão para interagir com a RNDS.

Gestor – indivíduo que responde formalmente por um estabelecimento de saúde.

Header – elemento do protocolo HTTP por meio do qual informação adicional é fornecida em uma requisição ou resposta. Um ou mais *headers* podem acompanhar uma requisição ou resposta. Por exemplo, o *header* identificado por *Content-Type* é empregado tanto em requisições quanto respostas para indicar o formato empregado no conteúdo da requisição ou da resposta.

HL7 – *Health Level Seven International* ou HL7 é uma instituição sem fins lucrativos fundada em 1987, com foco na produção de padrões para troca, integração, compartilhamento e recuperação de informação em saúde. Consulte <http://HL7.org> para detalhes.

Implementação – construção propriamente dita de código. Neste contexto, o código do Conector, responsável pela integração de um SIS à RNDS.

Informação em saúde – o que se deseja transferir entre estabelecimentos de saúde e a RNDS. A identificação desta informação visa atender uma necessidade da saúde. Relatório de Exame Laboratorial e Sumário de Alta são exemplos.

Integrador – denominação do desenvolvedor de *software* que produz o código que interage com a RNDS. É o foco de atenção deste material.

Interoperabilidade – é empregada como a capacidade de um SIS trabalhar em cooperação com outros sistemas, neste caso, materializada pelo uso dos *web services* oferecidos pelos ambientes da RNDS.

JSON – sigla para *JavaScript Object Notation*. JSON é um formato para troca de dados. Veja também XML.

Laboratório – um exemplo de estabelecimento de saúde.

Modelo computacional – modelo produzido a partir de um modelo de informação considerando recursos FHIR e eventuais perfis. Esse modelo fornece detalhes suficientes para o seu uso por um *software*.

Modelo de informação – modelo que identifica itens de informação que caracterizam determinada necessidade de troca de informação. Por exemplo, o modelo de informação do Resultado de Exame Laboratorial (REL), identifica todos os itens de dados que são empregados na definição de um REL. O modelo de informação é produzido para consumo por seres humanos. Veja também “modelo computacional”.

Paciente – termo evitado neste material em favor de usuário, um termo que chama a atenção para uma posição mais ativa do indivíduo em relação à sua própria saúde.

Padrão – termo empregado neste material com acepção de *software standard*, ou seja, uma norma ou requisito técnico usado por desenvolvedores de *software* trabalhando, neste contexto, visando a interoperabilidade entre diferentes programas, aqueles criados por integradores e aqueles criados pelo Datasus.

Perfil – do inglês *profile*, designa uma especialização ou particularização de um recurso FHIR com o propósito de adequá-lo a um uso local específico. Convém observar que recursos FHIR foram modelados para cobrir um conjunto significativo de cenários e que restrições adicionais são naturais e esperadas para contemplar especificidades locais.

Política Nacional de Informação e Informática em Saúde (PNIIS) – documento norteador da saúde para questões pertinentes à informação e informática em saúde. A versão publicada em 2016 está em revisão e, inclusive, já passou por Consulta Pública.

Projeto – a descrição de como requisitos serão realizados por meio da definição da estrutura do *software* em questão.

Recurso (FHIR) – ou *resource* (no FHIR) é o instrumento empregado pelo FHIR para a troca e/ou armazenamento de dados relacionados à saúde. Também pode ser visto como um formato de dados para necessidades comuns no domínio da saúde. Por exemplo, o FHIR define recursos como *Patient, Practitioner, Medication* e *Procedure*, dentre mais de uma centena de outros, típicos para o registro de informação em saúde.

Requisito – característica ou funcionalidade desejada em um *software*.

RNDS – acrônimo de Rede Nacional de Dados em Saúde. Trata-se de uma plataforma nacional de interoperabilidade de dados em saúde.

Servidor FHIR – implementação do padrão FHIR.

Sistema de Informação em Saúde (SIS) – sistema de informação em uso por um estabelecimento de saúde. A manutenção do SIS é necessária para integrar o estabelecimento de saúde em questão à RNDS.

Token (de acesso) – do inglês *access token*, é uma estratégia de autenticação para permitir uma aplicação acessar uma API. Esse *token* é uma credencial de acesso que identifica um usuário. Neste contexto, a aplicação é o que denominamos de Conector, enquanto a API é definida pela RESTful API FHIR, oferecida pelos *web services* disponibilizados nos ambientes da RNDS e, por fim, o usuário identificado pelo *token* é o estabelecimento de saúde. A RNDS faz uso de JWT (JSON Web Tokens). Consulte <https://jwt.io> para detalhes.

UML – sigla para *Unified Modeling Language*. UML é, provavelmente, a linguagem mais amplamente utilizada para o registro de modelos de *software*. Consulte detalhes em <http://uml.org/>.

Usuário – termo preferido neste texto para designar quem é assistido pela saúde, geralmente denominado de paciente.

X-Authorization-Server – *header* por meio do qual o *token* de acesso, obtido do endereço *Auth*, é fornecido em uma requisição dirigida ao endereço EHR.

XML – sigla para *eXtensible Markup Language*. É uma linguagem de “marcação” como HTML projetada para transportar e também armazenar dados. Atualmente, muitos sistemas de gerenciamento de banco de dados ditos NoSQL permitem o uso dos formatos XML e JSON.



Minibiografias

Organizadores

Fábio Nogueira de Lucena é graduado em Ciência da Computação (UFG), mestre e doutor em Ciência da Computação (UNICAMP), especialista em Informática em Saúde (UNIFESP), Project Management Professional (PMI) e Certified Software Development Professional (IEEE), além de possuir outras certificações da indústria de software. É professor titular do curso de Engenharia de Software do Instituto de Informática da UFG. Github: <https://github.com/kyriosdata>

E-mail: kyriosdata@ufg.br

Plínio de Sá Leitão Júnior é Engenheiro Eletricista, com mestrado e doutorado pela Universidade Estadual de Campinas (UNICAMP) na área de Engenharia de Software, e Especialização em Informática em Saúde pela Universidade Federal de São Paulo (UNIFESP). É Professor Associado no Instituto de Informática da Universidade Federal de Goiás (UFG), com atuação na graduação e na pós-graduação. Desenvolve pesquisas nos temas Teste de Software, Banco de Dados, Inteligência Computacional e Persistência de Registros Clínicos.

E-mail: plinio.sa.leitao.junior@ufg.br

Renata Dutra Braga é professora adjunta do Instituto de Informática da Universidade Federal de Goiás (UFG). É mestre e doutora em Ciências da Saúde pela Faculdade de Medicina da UFG, pós-graduada em Informática em Saúde (UNIFESP) e em Qualidade e Gestão de Software (PUC-GO) e é graduada em Sistemas de Informação (UniEvangélica). É atualmente vice-coordenadora da Comissão de Governança da Informação em Saúde (CGIS-UFG). Ensina, pesquisa, orienta e desenvolve projetos de extensão na área de saúde digital, com interesse, principalmente em modelagem de processos de negócios, engenharia de requisitos, modelos de informação, terminologias clínicas e padrões para a troca da informação em saúde.

E-mail: renatadbraga@ufg.br

PROGRAMA
EDUCACIONAL
EM **SAÚDE
DIGITAL**
DA UNIVERSIDADE
FEDERAL DE GOIÁS



SOBRE O E-BOOK

Tipografia: Montserrat

Publicação: Cegraf UFG

Câmpus Samambaia, Goiânia -
Goiás. Brasil. CEP 74690-900

Fone: (62) 3521-1358

<https://cegraf.ufg.br>
