

Sistema e-SUS Atenção Básica

Manual de Exportação

Este documento apresenta o modelo de integração do Sistema e-SUS AB com outros sistemas informatizados que estruturam o processo de trabalho das equipes de atenção básica em municípios com sistemas próprios.

Como eixo central tratamos das bibliotecas de comunicação Apache Thrift, das APIs do e-SUS AB e ainda do uso de XML como alternativa para efetivar a comunicação entre os sistemas.

API Thrift e-SUS AB Versão 2.0

XML Schema e-SUS AB Versão 2.0

Capítulo 1. Introdução à Estratégia e-SUS AB

1.1 Sistema com Coleta de Dados Simplificada

1.1.1 Cadastro da Atenção Básica

1.1.2 Fichas de Atendimento

1.1.3 Sistema de software e-SUS AB com CDS

1.2 Sistema com Prontuário Eletrônico do Cidadão

1.2.2 Sistema de Software e-SUS AB com PEC

1.3 Sistema de Informação em Saúde para a Atenção Básica

1.3.1 Fluxo de Transmissão de Dados para o SISAB

Capítulo 3. Modelo de Integração

3.1 Apache Thrift

3.1.1 Arquitetura do Thrift

3.2 APIs Thrift do Sistema e-SUS AB

3.2.1 Gerando um Arquivo para o Sistema e-SUS AB

3.3 Suporte XML do Sistema e-SUS AB

Capítulo 4. API Cidadão

4.1 Estrutura da API Cidadão

4.2 Exemplo de uso da API Thrift Cidadão

Capítulo 5. API RAS

5.1 Estrutura da API RAS

5.3 Entidades do Dicionário de Dados

5.4 Esquema Geral de um Arquivo RAS

[5.5 Exemplos de uso da API Thrift RAS](#)

[5.3 Exemplo de uso do RAS exportado via XML](#)

[Conclusão](#)

[Referências Bibliográficas](#)

[Anexo](#)

[Modelo de Troca de Informação do e-SUS AB](#)

[2.1 Troca de informação entre o sistema com CDS e com PEC](#)

[2.2 Coleta e envio das informações para o SISAB](#)

[2.3 Compartilhamento hierárquico dos dados do Sistema e-SUS AB](#)

Capítulo 1. Introdução à Estratégia e-SUS AB

A Estratégia e-SUS Atenção Básica é apoiada, essencialmente, por dois sistemas: i) Sistema de Informação em Saúde para a Atenção Básica (**SISAB**), o sistema de informação nacional; e ii) Sistema **e-SUS Atenção Básica**, composto por sistemas de software que instrumentalizam o processo de trabalho nas unidades básicas de saúde (UBS). Como estratégia, é fundamental que o e-SUS Atenção Básica garanta um processo amplo e padronizado de troca de informações entre sistemas em vários níveis de atenção e no próprio nível da Atenção Básica (AB).

Da mesma forma, para apoiar a **efetivação do cuidado continuado**, a Estratégia e-SUS AB precisa de uma estrutura que atue sobre um registro longitudinal dos eventos de saúde de um cidadão independentemente do sistema de software que as equipes de saúde utilizem para fazer a gestão local do serviço de saúde. Portanto, faz-se necessário uma base de Registro Eletrônico de Saúde (RES) que dialogue minimamente com as necessidades das Redes de Atenção à Saúde organizadas de forma intermunicipal e interestadual.

O Departamento de Atenção Básica (DAB) da Secretaria de Atenção à Saúde (SAS) do Ministério da Saúde (MS), com o objetivo de incentivar o uso de tecnologias de informação e comunicação em saúde na AB, vem apoiando o desenvolvimento de tecnologias que atendam às necessidades de gestão da AB, em especial, nos processos de gestão do cuidado e gestão por resultados.

Característica importante desse processo é a inversão do objetivo principal na construção das ferramentas e instrumentos que apoiam os processos de gestão. Esse projeto tem como foco as necessidades locais e da esfera municipal. Nessa perspectiva, entende-se que a melhoria e a qualificação do processo de trabalho das equipes de saúde da AB integradas às Redes de Atenção à Saúde devem trazer resultados significativos e promissores na gestão estadual e federal da AB.

Nesse contexto, a Estratégia e-SUS AB busca, por meio dos Sistemas e-SUS AB, implementar essas tecnologias para tornar mais fácil o processo de trabalho das equipes de saúde e de gestão, reduzindo o tempo gasto com a burocracia do uso e a alimentação dos sistemas de informação em saúde que fazem interface com a AB. A estratégia busca, ainda,

garantir que o desenvolvimento das soluções avancem na adoção de padrões internacionais da área de informática em saúde, ampliando, com isso, a interoperabilidade entre os sistemas gerenciais da saúde e de outras áreas no município.

O Sistema e-SUS AB é composto por dois sub-sistemas, pelo Sistema com Coleta Simplificada de Dados (CDS) e o Sistema com Prontuário Eletrônico do Cidadão (PEC), atendendo aos cenários de informatização distintos, como já foi apresentado em outros documentos.

Nas próximas seções, será apresentado, resumidamente, como funciona cada software do Sistema e-SUS AB, além de outros detalhes que podem ajudar os usuários dos sistemas na compreensão do processo de integração dos dados.

1.1 Sistema com Coleta de Dados Simplificada

O Sistema com Coleta de Dados Simplificada (CDS) foi formulado para atender às equipes de AB lotadas em UBS que ainda não possuem condições de infraestrutura tecnológica de informática para a utilização do sistema e-SUS AB com PEC (ver seção 2.3). O CDS tem como característica ser um sistema de transição, para viabilizar a implantação de um Sistema com PEC no tempo apropriado.

Na versão atual (versão 2.0), o Sistema com CDS utiliza **dez fichas** para o registro das informações:

1. Cadastro Individual
2. Cadastro Domiciliar
3. Ficha de Atendimento Individual
4. Ficha de Atendimento Odontológico
5. Ficha de Atividade Coletiva
6. Ficha de Procedimentos
7. Ficha de Visita Domiciliar
8. Marcadores de Consumo Alimentar
9. Ficha de Avaliação de Elegibilidade (exclusivo para equipes de AD)
10. Ficha de Atendimento Domiciliar (exclusivo para equipes de AD)

Essas fichas deverão ser digitadas no sistema de software e-SUS AB com CDS. Como um sistema transitório, essa ferramenta não tem a pretensão de ser um sistema exaustivo em relação às necessidades de informação das equipes de AB. No entanto, o CDS organiza um conjunto essencial de informações que estruturam o cadastro da AB e os registros de atendimentos realizados pelas equipes.

1.1.1 Cadastro da Atenção Básica

O Cadastro Nacional de Saúde, implementado pelo Sistema Cartão, ou simplesmente **CadSUS**, como é mais popularmente conhecido, é um sistema de informação de base

nacional que permite a **identificação dos usuários** das ações e serviços de saúde através de um número, único para cada cidadão, válido em todo o território nacional, e instrumentalizado por meio do Cartão Nacional de Saúde (CNS). O número do CNS permite a vinculação do usuário às ações e serviços de saúde realizados no Sistema Único de Saúde (SUS), em especial, vinculados aos profissionais, equipes e estabelecimentos de saúde responsável pelo atendimento. Ou seja, o CNS é também o instrumento necessário para a organização da Rede de Atenção à Saúde e de gestão do SUS, por meio do acesso a uma base nacional de dados de saúde desse cidadão. A Portaria nº 940/GM/MS, de 28 de abril de 2011, regulamenta o Sistema Cartão Nacional de Saúde, podendo hoje ser acessado por qualquer sistema que atenda minimamente os requisitos de interoperabilidade. Acesse <http://datasus.saude.gov.br/interoperabilidade> para mais informações.



O **Cadastro da AB** é uma extensão do **CadSUS**, no que se refere aos dados que apoiam as equipes de AB no mapeamento das características de saúde, sociais e econômicas da população adscrita ao território sob sua responsabilidade. Esse cadastro está organizado em duas dimensões: domiciliar e individual.

1. O **cadastro domiciliar** identifica as características socio sanitárias dos domicílios no território das equipes de AB. Esse cadastro busca identificar, também, situações de populações que habitam em locais que não podem ser considerados domicílio - por exemplo, situação de rua (IBGE, 2009) - e que, no entanto, devem ser monitoradas pela equipe de saúde. No cadastro domiciliar, o vínculo do cidadão ao seu domicílio é feito por meio do CNS do **responsável familiar**. Esse cadastro pode ser vinculado a mais de um responsável familiar e, portanto, a mais de um **núcleo familiar**.
2. O **cadastro individual**, associado ao cadastro domiciliar, identifica e vincula o usuário ao território das equipes de AB. Este é composto por duas partes: informações de identificação/sociodemográficas; e condições de saúde autorreferidas pelo usuário.

1.1.2 Fichas de Atendimento

No sistema com coleta de dados simplificada, todas as fichas de atendimento possuem um cabeçalho identificando a unidade, a equipe e o profissional que realizou o atendimento. Além disso, é possível registrar se foi um atendimento compartilhado com outro profissional. Da mesma forma, em todas as fichas, é possível identificar os usuários que receberam o atendimento pelo número do CNS e, ainda, o local do atendimento.

A **ficha de atendimento individual** contém o resumo do atendimento de nível superior. É nessa ficha que os profissionais de nível superior devem informar o que ocorreu no atendimento, e algumas informações fundamentais devem ser inseridas, tais como: tipo de atendimento; problema/condição avaliada; e conduta do atendimento. Para complementar essa ficha, tem-se a **ficha de procedimentos**, na qual é possível registrar os procedimentos e pequenas cirurgias realizadas no atendimento. A ficha de procedimentos é utilizada também para registrar procedimentos específicos realizados por técnicos de nível médio da unidade

Excepcionalmente, para os profissionais da Saúde Bucal, pode ser usada a **ficha de atendimento odontológico**, na qual se registram, de forma similar: o tipo de atendimento; o tipo da consulta odontológica; registros de vigilância em saúde bucal; e a conduta do atendimento – por exemplo, se foi agendada nova consulta na AB ou se o cidadão foi encaminhado para outro serviço.

A **ficha de visita domiciliar** busca, por meio de sua estrutura, coletar as informações sobre a realização de visitas domiciliares do agente comunitário de saúde (ACS). Assim como as outras fichas, esta ficha passa a ter o registro individualizado. Destaca-se que essa ficha passa a ser exclusiva para as visitas realizadas pelos ACS. Por sua vez, para o registro de atendimento no domicílio realizado por profissional de nível superior, é possível utilizar a ficha de atendimento individual, e informar o local de atendimento como sendo “Domicílio”.

Uma novidade para o registro de ações da AB é a **ficha de atividade coletiva**, usada para registrar ações administrativas, reuniões de equipe, ações de saúde, atividades coletivas de promoção de saúde ou atendimento em grupo.

Com exceção da ficha de atividade coletiva, as outras fichas passam a registrar vários atendimentos na mesma ficha. Com os registros organizados na vertical (em colunas), as fichas proporcionam um registro bastante objetivo. Entretanto, a ficha não tem intenção de ser exaustiva nas opções de preenchimento rápido, que em geral são acompanhados pelos campos “Outros”, em que é possível, quando necessário, registrar outras ações por meio de codificações e classificações utilizadas pelos sistemas de informações vigentes.

A ficha de **Marcadores de Consumo alimentar** permite a identificação de marcadores positivos ou negativos da alimentação e, de maneira mais dinâmica, a composição de indicadores. Para auxiliar a sua utilização e a orientação sobre práticas alimentares saudáveis, recomenda-se o uso do documento “Orientações para Avaliação de Marcadores de Consumo Alimentar na Atenção Básica”, disponível no Portal do DAB por meio do link: http://dab.saude.gov.br/portaldab/biblioteca.php?conteudo=publicacoes/marcadores_consumo_alimentar_atencao_basica

1.1.3 Sistema de software e-SUS AB com CDS

De fácil instalação e multiplataforma, o Sistema com CDS estrutura a digitação do cadastro e das fichas de atendimento, podendo ser utilizado também em conjunto com o Sistema e-SUS AB com PEC por meio do módulo CDS.

O sistema e-SUS AB com CDS é exclusivo para digitação e não tem funções gerenciais. Esse sistema possui um banco de dados embarcado, portanto, não é necessário fazer uma instalação de um SGBD em separado. Por ter funcionalidades limitadas às necessidades dessa aplicação, não se recomenda o armazenamento de um volume grande de informações no Sistema com CDS.



Figura 1.1: Tela Inicial do Sistema e-SUS AB com CDS

1.2 Sistema com Prontuário Eletrônico do Cidadão

O sistema com Prontuário Eletrônico do Cidadão (PEC) foi formulado para atender às equipes de Atenção Básica lotadas em unidades básicas de saúde (UBS) parcialmente ou totalmente informatizadas. Um Sistema com Prontuário Eletrônico é um sistema que amplia a integração e a gestão do cuidado pelos profissionais, e é indispensável do ponto de vista de gerar valor de uso aos sistemas de informação em saúde.

No entanto, um prontuário eletrônico pode ser implementado de diversas formas, considerando diferentes processos de trabalho. Nesse momento, cabe apontar algumas diretrizes usadas pelo sistema visando garantir a atenção integral à saúde do paciente tal como preconiza a Política Nacional de Atenção Básica (PNAB).

1.2.2 Sistema de Software e-SUS AB com PEC

O Sistema com PEC é um sistema complexo, pois busca estruturar o registro do conjunto de informações que apoiam a organização e troca de informação entre os profissionais das equipes de AB. É importante destacar que este é um sistema com prontuário

eletrônico, ou seja, não se limita a ser apenas o registro no prontuário eletrônico e amplia o conjunto de ferramentas e funcionalidades para atender a todas as diretrizes de um sistema de informação para a AB.

Na versão atual do sistema, como se pode ver na imagem de sua tela inicial, estão disponíveis os seguintes módulos:

- **Módulo Cidadão:** contempla o cadastro do cidadão e a integração com o CadSUS. Todo cidadão atendido no Sistema com PEC precisa de um cadastro, mesmo que este ainda não possua um CNS.
- **Módulo Agenda:** contempla as necessidades de configuração de agenda dos profissionais, marcação de consulta, controle de chegada e controle de faltosos da agenda, além das reservas de horários por demandas administrativas e de atividades programáticas da equipe / profissional.
- **Módulo Atendimentos:** contempla o controle da lista de atendimento (fluxo do cidadão na unidade), acolhimento à demanda espontânea (inserção na lista de atendimento e escuta inicial), prontuário eletrônico usando o modelo de Registro Clínico Orientado à Problemas - RCOP (Folha de Rosto, SOAP, Lista de Problemas, etc), entre outras funcionalidades.
- **Módulo Relatórios:** contempla a geração de relatórios de cadastro, situação de saúde, atendimentos do território, entre outros.
- **Módulo Administração:** contempla o cadastro geral e a configuração do sistema, cadastro do profissional (usuário do sistema), controle de unidades, controle de perfil de usuário, importação dos dados do CNES, exportação do BPA, entre outros.

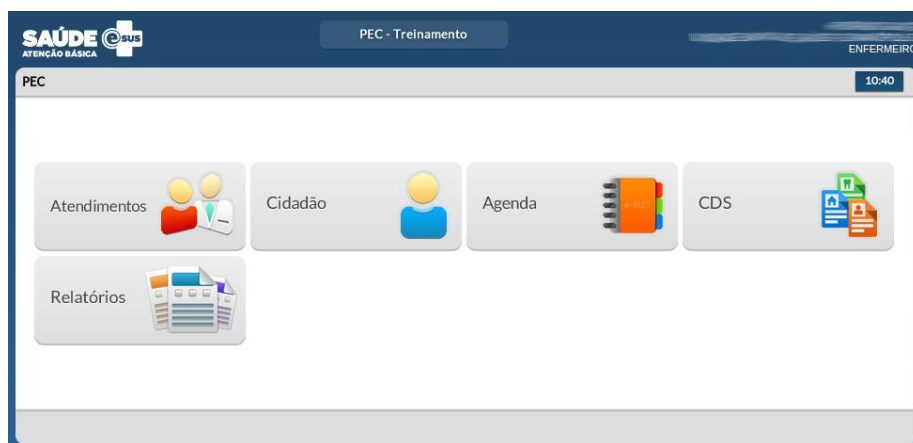


Figura 1.2: Tela Inicial do Sistema e-SUS AB com PEC

1.3 Sistema de Informação em Saúde para a Atenção Básica

A Portaria nº 1.412, de 10 de julho de 2013, instituiu o Sistema de Informação em Saúde para a Atenção Básica (**SISAB**) como o novo sistema de informação nacional da Atenção Básica, em substituição ao atual Sistema de Informação da Atenção Básica (**SIAB**).

Destaca-se, inicialmente, que tanto o Sistema com CDS quanto o Sistema com PEC fornecerão as mesmas informações ao SISAB. Para quem utiliza o Sistema e-SUS AB com PEC, o próprio sistema se encarregará de organizar as informações a serem enviadas ao SISAB. Para os municípios que utilizam outros sistemas com prontuário eletrônico, também será possível gerar as informações de acordo com o modelo de coleta simplificada (similar ao Sistema com CDS) e então enviar os dados.

Todos os dados do Sistema e-SUS AB são organizados no Módulo Transmissão de Dados para envio ao SISAB. No próximo capítulo, será explicado com mais detalhes como esse processamento ocorre. Para mais informações sobre Transmissão dos Dados para o SISAB, consulte o [Manual de Uso do Sistema com Prontuário Eletrônico do Cidadão \(PEC\)](#), especialmente na [seção 3.6](#).

O consolidado das informações enviadas ao sistema nacional podem ser consultadas por meio do Painel do SISAB, por meio do link: sisab.saude.gov.br, como podemos ver na Figura 1.3.

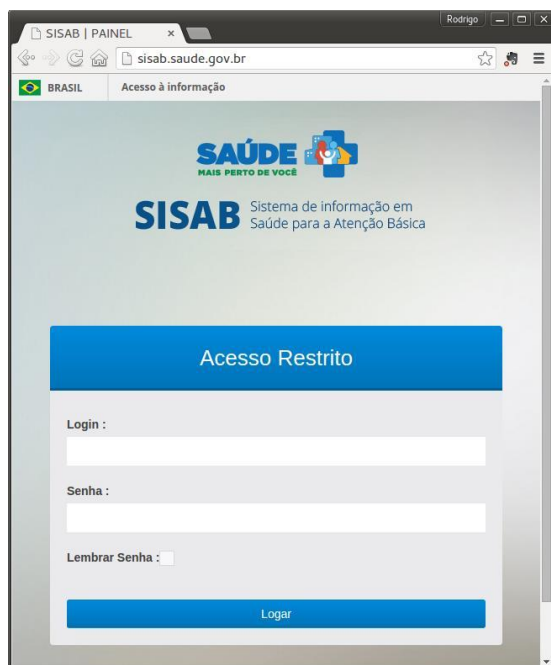


Figura 1.3: Tela Inicial do SISAB

Para entender melhor como funciona o fluxo de transmissão dos dados para o SISAB, na seção seguinte, amplia-se a discussão, contemplando as regras gerais de envio dos dados.

1.3.1 Fluxo de Transmissão de Dados para o SISAB

O fluxo de transmissão dos dados para o novo sistema de informação se dá de forma similar ao fluxo de envio dos dados do antigo SIAB pelo município. Este fluxo é definido em **portaria anual**, que determina o prazo máximo de envio das informações do sistema organizadas por competência mensal, porém em **fluxo contínuo**.



Figura 1.4: Esquema do fluxo de transmissão do SISAB

Conforme ilustrado na Figura 1.4, como regra geral, as competências fazem um recorte da produção de informação dentro de um mês de competência que corresponde exatamente aos dias do mês em questão. Ao encerrar a competência, o município terá do primeiro ao vigésimo dia do mês subsequente como prazo máximo para enviar os dados para a base nacional. Na Figura 1.5, podemos ver um fluxo mínimo de envio dos dados de um Sistema Próprio ao SISAB.

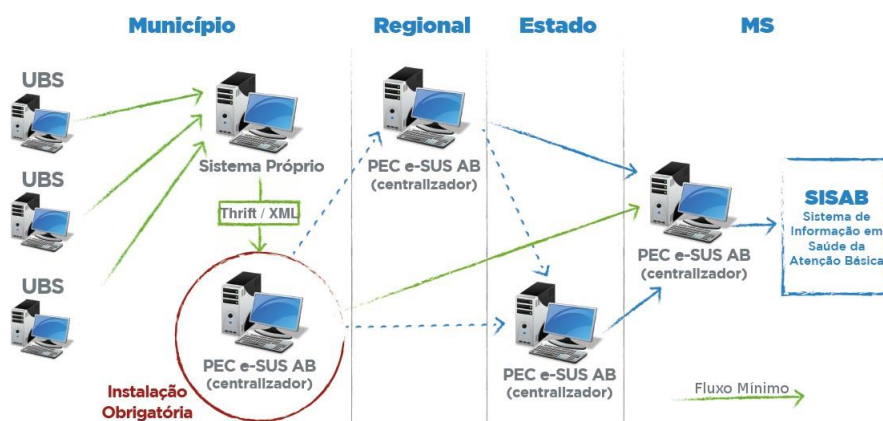


Figura 1.5 – Modelo de Integração com Sistema Próprio

Capítulo 3. Modelo de Integração

O Sistema e-SUS AB está usando o framework de comunicação da Apache Thrift para implementar os recursos de importação e exportação dos dados do sistema próprio para o Sistema e-SUS AB, pois este framework oferece suporte a múltiplas linguagens de programação, bem como também a formatos padronizados, o que aumenta a capacidade de outros sistemas de enviar informações via Sistema e-SUS AB ao SISAB.

A partir da versão 2.0 do Sistema e-SUS AB, pode-se optar por desenvolver a exportação de cadastros e atendimentos à população usando API Thrift, disponível em várias linguagens, ou ainda por meio de arquivos padronizados no formato XML.

De forma geral, como podemos ver na Figura 3.1, o município que optar por utilizar um sistema próprio, após fazer cadastro no [Sistema de Controle de Uso do Sistema e-SUS AB](#), poderá exportar os dados de seu sistema local para o SISAB por meio dos seguintes passos:

1. Escolher a tecnologia mais adequado à plataforma tecnológica do sistema próprio em uso no município;
 - a. Apache Thrift, via APIs, ver seção [3.1](#) e [3.2](#);
 - b. XML, usando as estruturas das APIs porem gerando manualmente os arquivos, ver seção [3.3](#);

2. Exportar os arquivos de dados usando a tecnologia escolhida;
3. Armazenar os arquivos em local adequado e seguro;
4. Importar os arquivos no Sistema e-SUS AB, por meio do Módulo de Transmissão dos Dados -> Controle de Recebimento de Fichas;
5. Processar os dados importados;
6. Verificar inconsistência por meio do Relatório de Inconsistências, no módulo de transmissão; e por último;
7. Transmitir os dados ao SISAB, via Centralizador Nacional do e-SUS AB, por meio do Módulo de Transmissão dos Dados -> Controle de Envio de Fichas;

A seguir, este capítulo traz mais explicações sobre o framework Apache Thrift e alguns conceitos que devem ser compreendidos para se fazer um processo mais rápido de exportação dos dados a partir dos sistemas próprios. Na sequência, na seção [3.2](#), serão abordados os tipos de APIs disponíveis para exportação de dados

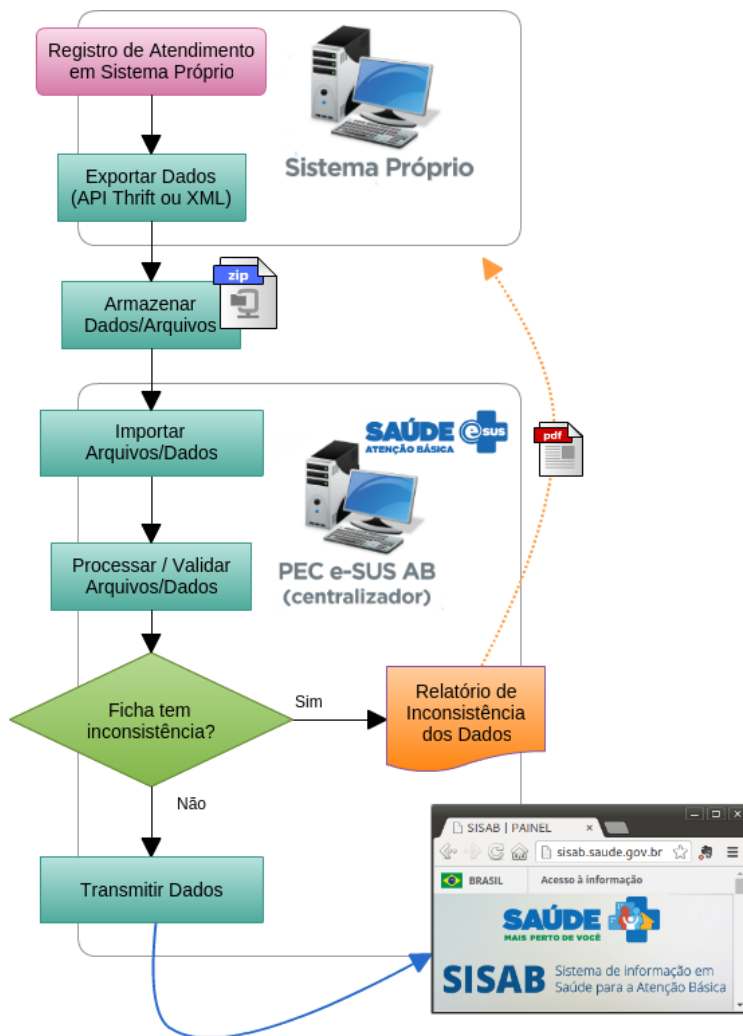


Figura 3.1: Resumo Esquemático de Exportação de Dados ao SISAB

3.1 Apache Thrift

O Apache Thrift é um *framework* RPC (Remote Procedure Call), ou seja, é um conjunto de bibliotecas de sistema que auxiliam desenvolvedores a implementar chamadas de procedimentos remotos, porém oferecendo uma estrutura para utilização de múltiplas linguagens de programação entre clientes e servidores.

O termo em inglês *thrift* significa “brechó”. Remetendo, assim, à noção de troca de objetos, o Apache Thrift oferece um ambiente de troca de serviços entre aplicações por meio

de uma estrutura para desenvolvimento de serviços escaláveis entre linguagens. Dessa forma, oferece não apenas suporte à geração de código para várias linguagens mas também uma pilha de software, simplificando o desenvolvimento de serviços relacionados à rede de computadores.

Antes de tornar-se um projeto de software livre do Apache, Thrift era uma biblioteca usada internamente no Facebook criada pelos seus desenvolvedores justamente pela necessidade de oferecer serviços em diferentes linguagens. Atualmente o Apache Thrift é usado por grandes desenvolvedores de software como Twitter e LinkedIn.

Abaixo alguns dos benefícios de se usar o Apache Thrift:

- Serialização entre linguagens com menor sobrecarga do que as alternativas como SOAP, devido à utilização do formato binário;
- A biblioteca leve, enxuta e limpa, sem estrutura para código. Não há arquivos de configuração XML;
- O formato de nível de aplicação e de serialização são claramente separados, podendo ser modificados de forma independente;
- Os estilos de serialização predefinidos incluem: binário, HTTP-friendly e binário compacto;
- Sem dependências de construção ou de software não-padrão. Não mistura licenças de softwares incompatíveis;
- Oferece suporte para várias linguagens de programação (em níveis variados), incluindo Java™, PHP, C/C++, Delphi, Python, Ruby, Smalltalk, Haskell, Erlang, Node.js, Go, D, entre outras;
- Compatível com plataformas de computação distribuída como Apache Hadoop e Apache Cassandra.

3.1.1 Arquitetura do Thrift

Ao se avaliar os desafios da interação entre linguagens em um ambiente de rede, alguns componentes-chave foram destacados no Thrift:

- **Tipos:** Um sistema de tipo comum devem existir em toda linguagens de programação, sem exigir que o desenvolvedor do aplicativo use tipos de dados Thrift personalizados ou tenha que escrever seu próprio código de serialização.
- **Transporte:** Cada linguagem tem de ter uma interface comum para o transporte bidirecional de dados. Os detalhes de como um determinado transporte é implementado não deve importar para o desenvolvedor do serviço. O mesmo código do aplicativo deve rodar sobre um fluxo sockets TCP, na memória ou em arquivos no disco.
- **Protocolo:** Tipos de dados devem ter alguma maneira de utilizar a camada de transporte para codificar e decodificar a si mesmos. Mais uma vez, o desenvolvedor do aplicativo não precisa se preocupar com esta camada. Se o serviço usa um XML ou protocolo binário é irrelevante para o código da aplicação. Tudo o que importa é que os dados possam ser lidos e escritos de forma consistente e determinística;

- **Versionamento:** Para serviços robustos, os tipos de dados envolvidos devem fornecer um mecanismo para controle de versão. Especificamente, deve ser possível adicionar ou remover campos em um objeto ou alterar a lista de argumentos de uma função sem qualquer interrupção do serviço (ou pior, falhas de segmentação desagradáveis);
- **Processadores:** Por fim, gerar um código capaz de processar fluxos de dados para realizar chamadas de procedimento remoto em diferentes linguagens.

Estas características deram as diretrizes para a definição de uma boa arquitetura, como será descrito a seguir.

O *Thrift* implementa uma pilha (*stack*) de software que simplifica o desenvolvimento de aplicativos de comunicação em várias linguagens. Conforme podemos ver na Figura 3.2, na parte inferior da pilha está a interface física de entrada e saída dos dados, podendo ser um fluxo de rede ou ainda um arquivo do sistema. Essa interface influencia os níveis mais altos da pilha.

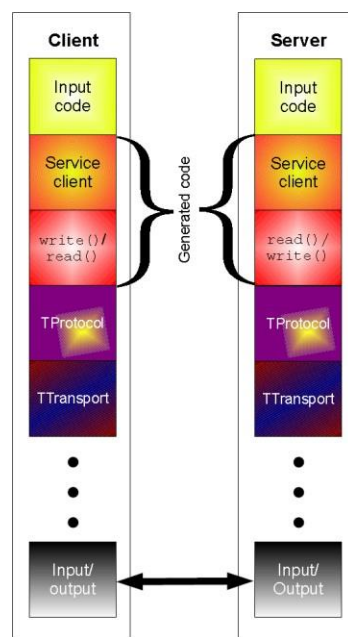


Figura 3.2 – Arquitetura cliente/servidor Apache Thrift API (Fonte: Wikipedia)

As camadas de transporte e protocolo que definem como os dados são movidos e quais formatos eles assumem, usando um processador, que contém os fluxos de entrada e saída. *TBinaryProtocol* define um protocolo binário eficiente para comunicação que pode ser usado sobre transporte *TFileTransport* que gera um arquivo para ser enviado ou salvo no disco, por exemplo.

Considerando outros tipos de implementação, no nível superior estão tipos de servidor que podem ser implementados com o auxílio de *Thrift*. Essa configuração pode incluir um servidor de encadeamento único para depuração (TSimpleServer), um servidor HTTP que pode fornecer URLs semelhantes ao REST (Representational State Transfer) usando THttpServer, ente outros. Portanto, o *Thrift* é escrito não apenas para simplificar a comunicação entre várias abordagens (protocolos e transportes), mas também para simplificar o desenvolvimento de servidor usando vários estilos de servidor.

O *Thrift*, como visto, ainda dá suporte a um sistema de tipos que permite comunicação entre linguagens. Esse sistema oferece suporte para tipos como *byte*, *short*, *int double*, *string* e tipos mais avançados, como contêineres (listas, mapas) e estruturas. Esse sistema de tipos genérico é a base comum (serialização e desserialização) para comunicação de dados.

3.2 APIs Thrift do Sistema e-SUS AB

Como estamos tratando especificamente da integração de Sistemas Próprios com o Sistema e-SUS AB, muita da complexidade da implementação já foi absorvida pelo Sistema e-SUS AB, que implementa toda a parte de servidor definindo o formato dos dados e disponibilizando as APIs geradas pelo *Thrift*, em diferentes linguagens, para que sejam usadas pelos desenvolvedores que pretendem exportar dados para o Sistema e-SUS AB.

Foram gerados dois pacotes de APIs para integração com o Sistema e-SUS AB:

- **API Thrift Cidadão** tem o objetivo de garantir uma forma de importar cadastros já consolidados no município ou estado, de modo que seja possível reutilizar esses cadastros no Sistema com PEC, no módulo cidadão.
- **API Thrift RAS** tem o objetivo de oferecer uma forma de importar os Registros de Atendimento Simplificado (RAS), como vimos na seção 3.1, semelhante ao formato usado pelo Sistema com CDS. Este modelo simplificado permite que os municípios que tem sistemas próprios e estruturados nas Unidades Básicas de Saúde possam exportar os dados para que estes sejam enviados ao SISAB via Centralizador Municipal do Sistema e-SUS AB.

A partir das APIs é possível gerar os arquivos exportados dos Sistemas Próprios para serem importados no Sistema e-SUS AB. A estrutura de cada API será detalhada mais a frente. Na próxima seção veremos, de forma genérica, como é gerado o arquivo para ser importado.

3.2.1 Gerando um Arquivo para o Sistema e-SUS AB

Como vimos na Figura 3.1, a implementação do *Thrift* ocorre em camadas para facilitar o desenvolvimento considerando diferentes problemas da integração. A parte do Servidor é implementada pelo Sistema e-SUS AB, portanto é ao desenvolvedor apenas gerar os dados a partir de sua aplicação na condição de **cliente** da aplicação.

O Sistema e-SUS AB, nesse momento, oferece duas formas de exportação de dados via arquivo, sendo: XML e thrift. Assim, nesta seção, serão apresentados os passos para gerar esse arquivo, abstraindo-se as especificidades de alguma linguagem em particular.

Quadro 1 – Passo a passo para geração do arquivo para o Sistema e-SUS AB - Thrift

Passo 1	Configurar a aplicação para usar Thrift
<p>Para que seja possível usar as APIs geradas pelo Apache Thrift, é necessário instalar e configurar seu ambiente de desenvolvimento para usar as bibliotecas do Apache Thrift.</p> <p>As linguagens que já estão com APIs disponíveis para o Sistema e-SUS AB são:</p> <ul style="list-style-type: none">• Java (requer Ant e Java 1.7);• Delphi (requer Delphi 2010 ou superior);• C# (requer Mono >= 1.2.6 or .NET framework >= 3.5);• PHP (requer PHP 5.0); e• Ruby (Ruby 1.8, bundler gem). <p>Mais informações sobre como usar as <i>libs</i> do <i>Thrift</i> estão disponíveis em: http://thrift.apache.org/lib/</p> <p>Caso haja necessidade de alguma outra API suportada pelo <i>Thrift</i> para o Sistema e-SUS AB, deve-se contatar a equipe de desenvolvimento, por meio do endereço: nti.dab@saude.gov.br</p>	
Passo 2	Baixar a API do Sistema e-SUS AB a ser utilizada
<p>O <i>download</i> da API gerada pelo Thrift para o Sistema e-SUS AB pode ser feito a partir do <i>site</i> do e-SUS AB, disponível em: http://dab.saude.gov.br/esus</p> <p>Na área de download, selecionar a opção “Integração Sistemas Próprios” e, na sequência, clicar em “Thrift Cidadão” ou “Thrift RAS”, conforme for a necessidade.</p>	
Passo 3	Copiar o código da API para a aplicação
<p>Após baixar o arquivo, descompactá-lo e copiar os arquivos correspondentes à linguagem usada por sua aplicação para dentro de seu código base.</p>	
Passo 4	Incluir a API e as bibliotecas na aplicação
<p>Incluir a API do Sistema e-SUS AB correspondente à linguagem utilizada e às bibliotecas do <i>Thrift</i> no código base da aplicação para que seja possível chamar as funções dentro desta.</p> <p>Para exemplos sobre como fazer esse procedimento, ver os exemplos para cada linguagem em: http://thrift.apache.org/tutorial/</p>	
Passo 5	Configurar a camada de transporte e protocolo

Após preparar a aplicação para usar o *Thrift*, basta definir como e em que formato os dados serão enviados, ou seja, deve-se definir as camadas de transporte e protocolo, já implementadas pelo *Thrift*.

O Sistema e-SUS AB utiliza:

- **TTransport: StreamTransport;**
 - Java: TIOStreamTransport,
 - Delphi: TStreamTransportImpl,
 - C#: TStreamTransport,
 - PHP5: TMemoryBuffer,
 - Ruby: IOStreamTransport.
- **TProtocol: BinaryProtocol;**
 - Java: TBinaryProtocol,
 - Delphi: TBinaryProtocolImpl,
 - C#: TBinaryProtocol,
 - PHP5: TBinaryProtocol,
 - Ruby: BinaryProtocol.

Para mais detalhes, ver os exemplos de funções de serialização disponíveis no manual:

- Exemplo na linguagem Java: **ThriftSerializer.java**
- Exemplo na linguagem Delphi: **ThriftSerializer.pas**

Passo 6 **Preencher os dados**

Nesse momento, é possível criar as rotinas que irão preencher os dados conforme definido pelas APIs. Como já mencionado, as APIs geradas pelo *Thrift* definem as estruturas dos dados do e-SUS AB (DadoTransporteThrift ou CidadaoTransportThrift).

Nos capítulos a seguir, serão detalhadas as estruturas e a melhor forma de utilizar essas APIs para importar os dados ao Sistema e-SUS AB.

Passo 7 **Empacotar os dados**

O empacotamento dos dados a serem enviados ao Sistema e-SUS AB é feito por meio de um conjunto de arquivos compactados por uma biblioteca ZIP. Cada arquivo a ser compactado corresponde a uma “Ficha” serializada dos dados.

Dentro dos pacotes ZIP, cada arquivo (*entries*) deve ser nomeado com a extensão do dado empacotado, a saber:

- Fichas CDS/RAS (DadoTransporteThrift): extensão “.esus”; e
- Cadastro de Cidadão (CidadoTransportThrift): extensão “.cidadao”.

Para mais detalhes ver o exemplo disponível no manual:

- Exemplo na linguagem Java: [ZipWriterExemplo.java](#)
- Exemplo na linguagem Delphi: [ZipWriterExemplo.pas](#)

A serialização dos dados é feita pela própria estrutura do *Thrift*, conforme definido pelo TProtocol e encapsulado pelo TTransport, de acordo com o formato que foi configurado anteriormente.

Fonte: DAB/SAS/MS.

3.3 Suporte XML do Sistema e-SUS AB

Como alternativa ao Thrift para exportação dos dados de sistemas próprios para o Sistema e-SUS AB, a partir da versão 2.0, o Sistema e-SUS AB traz suporte à importação no formato **XML**, exceto para a API do Cidadão.

Para tal, são disponibilizados os XSD (XML Schema Definition) para validação dos XML exportados pelo sistema próprio. Os **XSD** correspondem ao modelo **RAS**, e têm o objetivo de oferecer uma forma de importar os Registros de Atendimento Simplificado (RAS), como visto na seção 3.1.

Quadro 2 – Passo a passo para geração do arquivo para o Sistema e-SUS AB - XML

Passo 1	Preparar a aplicação para usar XML
Para que seja possível usar XML em sua aplicação para exportar dados para o e-SUS AB não é necessário ter nenhuma estrutura auxiliar, considerando que os dados são gerados em texto puro, usando o formato XML W3C. Para mais informações sobre XML siga o link: https://www.w3.org/standards/xml/core	
Passo 2	Baixar os XSDs do Sistema e-SUS AB a ser utilizado
O <i>download</i> dos XSDs para validar o XML de exportação para o e-SUS AB estão disponível em: https://github.com/esusab/integracao/tree/master/XSD	
Passo 3	Preencher os dados no XML
Nesse momento, é possível criar as rotinas que irão preencher os dados a partir das definições dos XSDs.	
Passo 4	Empacotar os dados (Arquivo ZIP)
O empacotamento dos dados a serem enviados ao Sistema e-SUS AB é feito por meio de um conjunto de arquivos compactados por uma biblioteca ZIP. Cada arquivo a ser compactado corresponde a uma "Ficha" dos dados a serem enviados. Dentro dos pacotes ZIP, cada arquivo (<i>entries</i>) deve ser nomeado com a extensão do dado empacotado, a saber: ".esus.xml"	

Capítulo 4. API Cidadão

Esta API permite integrar o Sistema com PEC a outro sistema de informação migrando uma base de cadastro de cidadãos, minimizando o esforço de recadastramento e digitação. O cidadão é identificado pelo CNS ou CPF, sendo assim, os dados já existentes serão alterados com a nova importação quando for localizado o mesmo cidadão. Esta importação pode ser realizada em qualquer momento.

Para entender melhor a aplicação e forma de uso da API, a seguir será detalhada a estrutura da API, bem como será apresentado um exemplo de implementação usando a linguagem Java.

4.1 Estrutura da API Cidadão

A API Cidadão implementa a estrutura dos dados que devem ser salvos em arquivo, como vimos na Seção 3.2.1. Esta estrutura recebeu o nome de `CidadaoTransportThrift` e pode ser visualizada na Figura 4.1, com destaque aos dados obrigatórios. Dois campos são marcados como condicionais, pois o preenchimento deles depende de outro valor em bloco, a saber: `CNS` (`naoPossuiCNS=true`) e `nomeMae` (`desconheceNomeMae=false`).

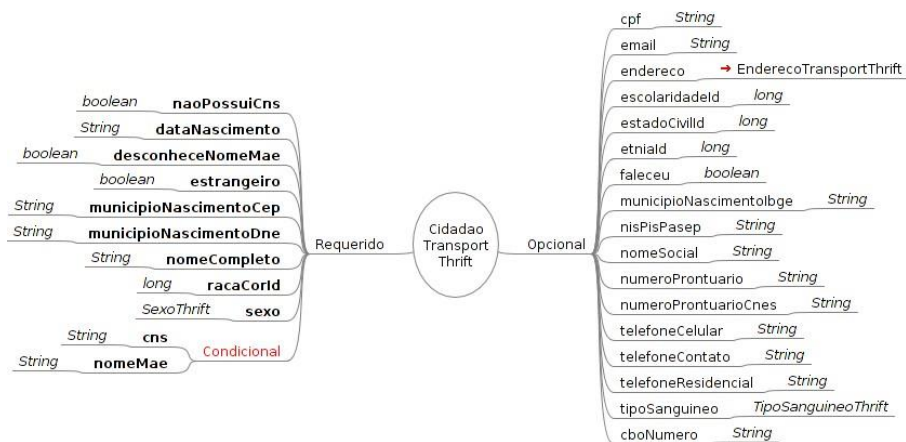


Figura 4.1 Estrutura dos dados da API `CidadaoTransportThrift`

A API ainda usa algumas estruturas auxiliares como: `SexoThrift`, `TipoSanguineoThrift` e `EnderecoTransportThrift` (ver Figura 4.2). Além dessas estruturas auxiliares ainda é necessário fazer referência a um conjunto de informações provenientes de Tabelas de Referência, a saber: `Escolaridade`, `Raça/Cor`, `Etnia`, `Localidade` (para `Município de Nascimento` e `Município de Residência`), `CBO` e `Estado Civil`.

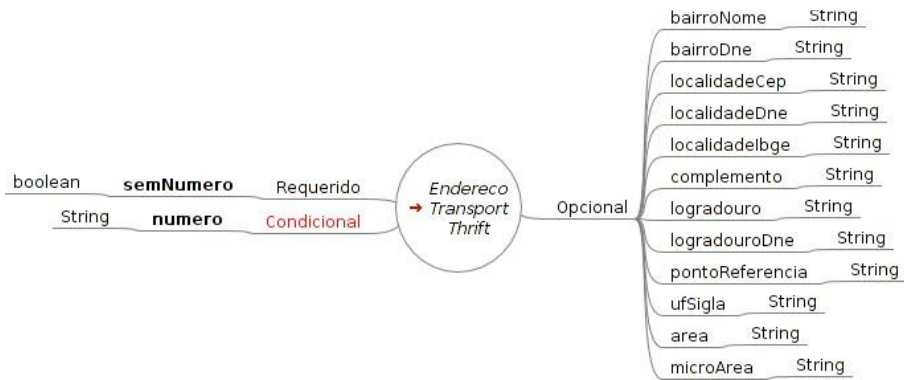


Figura 4.2 Estrutura dos dados da API EnderecoTransportThrift

4.2 Exemplo de uso da API Thrift Cidadão

Nesta seção iremos apresentar um exemplo de uso da API Thrift Cidadão com comentários para tornar mais fácil a compreensão dos conceitos usados e das variáveis do próprio Sistema e-SUS AB em relação aos dados do cidadão.

O exemplo a seguir não pretende ser completo, portanto alguns detalhes julgados irrelevantes foram suprimidos, portanto este código deve ser adaptado para ser executado.

Exemplo 4.1 – Implementação na Linguagem Java

[ExemploThriftCidadaoJava.java](#)

Capítulo 5. API RAS

Esta API é recomendada aos municípios que optarem pela utilização de outros sistemas de informação, aqui chamados de **Sistemas Próprios**, e que desejam transmitir seus dados produzidos na Atenção Básica para o Ministério da Saúde. A estrutura RAS (Registro de Atendimento Simplificado) é baseada em um conjunto mínimo de informações em saúde coletadas pelos Sistemas PEC e CDS, conforme descrito no Capítulo 2, e deverá ser adotada como processo padrão para transmissão dos dados ao Sistema de Informação em Saúde para a Atenção Básica (SISAB), como vimos anteriormente (ver Seção 1.3.1).

Para entender melhor a aplicação e a forma de uso da API, a seguir será detalhada a estrutura da API, bem como será apresentado um exemplo de implementação usando a linguagem Java, ou ainda utilizando texto puro no formato XML.

5.1 Estrutura da API RAS

A API RAS implementa a estrutura dos dados que devem ser salvos em arquivo, como vimos na Seção 3.2.1. Esta estrutura recebeu o nome de Camada de Transporte de Dados ([ver Dicionário de Dados](#)) e pode ser visualizada na Figura 5.3, com destaque aos dados requeridos (obrigatórios). Esta API tem um comportamento diferente da API Cidadão, pois ela tem antes dos dados de atendimento uma camada para identificar o envelope dos dados que estão sendo transmitidos.

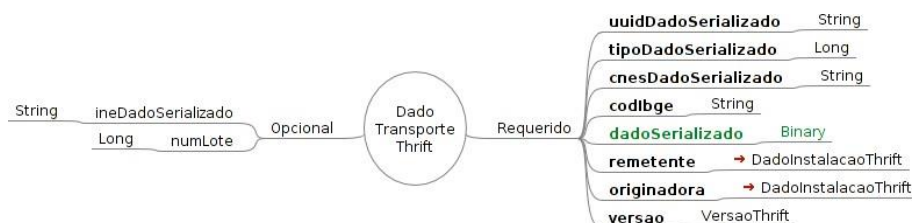


Figura 5.3 Estrutura dos dados da API DadoTransporteThrift

A seguir mais detalhes sobre a Camada de Transporte de Dados :

- **uuidDadoSerializado**: UUID do dado (identificador "universal" gerado na criação do item vide https://en.wikipedia.org/wiki/Universally_unique_identifier);
- **tipoDadoSerializado**: Tipo/classe do objeto serializado (<http://esusab.github.io/integracao/docs/dicionario/dicionario.html#tipodadoserializado>);
- **cnesDadoSerializado**: CNES da unidade de saúde ;
- **codIbge**: Código IBGE do dado serializado
- **ineDadoSerializado**: código do Identificador Nacional da Equipe (INE) da equipe que efetivamente gerou a informação/registro de atendimento;

- **numLote**: Número do Lote, definido pelo sistema próprio;
- **dadoSerializado**: Dado estruturado a partir de uma ficha . É neste campo que será vinculada o registro de atendimento/cadastro que está sendo enviada, usando as estruturas de fichas descritas no [dicionário de dados](#).
- **remetente**: Identifica a instalação que **enviou** o dado, por meio da estrutura [DadoInstalacao](#) , conforme descrito a seguir;
- **originadora**: Identifica a instalação que cadastrou/**digitou**, por meio da estrutura [DadoInstalacao](#). Para envio de dados por Sistemas Próprios, em geral, originadora e remetente são a mesma instalação;
- **versao**: Identifica a **versão do E-SUS AB** usando Versao

Considerando o fluxo de informações de um Sistema Próprio para o Sistema e-SUS AB, alguns campos devem ser definidos para que seja possível de identificá-los no Módulo de Controle de Recebimento dos dados, e eventualmente no Relatório de Inconsistências dos arquivos. Para isso é utilizada um estrutura auxiliar, [DadoInstalacao](#) (ver Figura 5.3), para descrever os dados da instalação.



Figura 5.4 Estrutura dos dados da API DadoInstalacao

A seguir mais detalhes sobre a **DadoInstalacaoThrift**:

- **contraChave**: Identifica o software que está gerando a informação. Não demanda nenhuma descrição mandatória, mas sugere-se usar o NomeDoSoftware_Versao;
- **uuidInstalacao**: Identificador UUID da instalação do sistema, associado à contraChave auxilia no controle de fluxo da aplicação que gerou a informação. Cada instalação deve controlar seu próprio identificador, gerado apenas no primeiro envio.
- **cpfOuCnpj**: Cpf do responsável ou CNPJ da empresa responsável, em geral usa-se o CPF do profissional cadastrado no Centralizador Municipal que receberá os dados.
- **nomeOuRazaoSocial**: Nome do responsável ou Razão Social da empresa responsável
- **fone**: Telefone da pessoa ou empresa responsável
- **email**: Email da pessoa ou empresa responsável

5.3 Entidades do Dicionário de Dados

A seguir, são apresentadas as entidades, detalhadas no [Dicionário de dados das APIs](#), usadas na API RAS. Para mais informações baixe o dicionário de dados disponível no site do e-SUS AB, na seção de [Integração](#).

Quadro 2 – Entidades usadas na API RAS

Profissional/cabeçalho
<ul style="list-style-type: none"> • UnicaLotacaoHeaderThrift: essa entidade é utilizada para representar o profissional responsável pelas fichas. • VariasLotacoesHeaderThrift: essa entidade é utilizada para representar o profissional responsável pela ficha, bem como os outros que o auxiliaram na atividade. • ProfissionalCboRowItemThrift: entidade usada em listas de profissionais. • HeaderCdsCadastroThrift: entidade utilizada para representar o profissional que realizou uma ação (cadastro individual, domiciliar, ou visita domiciliar), e a respectiva data.
<p>XSD para suporte ao XML:</p> <ul style="list-style-type: none"> v2000unicalotacaoheader.xsd v2000variaslotacoesheader.xsd v2000profissionalcborowitem.xsd v2000headercdscadastro.xsd
Atendimento
Atendimento Individual
<ul style="list-style-type: none"> • AtendimentoIndividualMasterThrift: entidade que organiza os dados de Atendimento Individual (cabeçalho e atendimentos). <ul style="list-style-type: none"> ○ AtendimentoIndividualChildThrift: entidade que organiza os dados de Atendimento Individual, individualmente. ○ ProblemaCondicaoAvaliacaoAIThrift: entidade que organiza os Problemas e Condições avaliados no atendimento individual. ○ OutrosSiaThrift: entidade que organiza os Exames Solicitados e/ou Avaliados referenciados pelo código do SIGTAP.
<p>XSD para suporte ao XML:</p> <ul style="list-style-type: none"> v2000fichaatendimentoindividualmaster.xsd v2000fichaatendimentoindividualchild.xsd v2000problemacondicaoavaliacaoai.xsd v2000outrossia.xsd

Procedimentos
<ul style="list-style-type: none"> ● ProcedimentoMasterThrift: entidade que organiza a realização de procedimentos realizados por um profissional, tanto os individualizados quanto os consolidados. <ul style="list-style-type: none"> ○ FichaProcedimentoChildThrift: entidade que organiza a realização de procedimentos realizados a um cidadão, individualmente.
<p>XSD para suporte ao XML:</p> <p>v2000fichaprocedimentomaster.xsd v2000fichavisitadomiciliarchild.xsd</p>
Atendimento Odontológico Individual
<ul style="list-style-type: none"> ● AtendimentoOdontologicoMasterThrift: entidade que organiza os dados de Atendimento Odontológico Individual (cabeçalho e atendimentos). <ul style="list-style-type: none"> ○ FichaAtendimentoOdontologicoChildThrift: entidade que organiza os dados de Atendimento Odontológico Individual, individualmente. ○ ProcedimentoQuantidadeThrift: entidade que organiza os itens da lista Outros Procedimentos do atendimento odontológico.
<p>XSD para suporte ao XML:</p> <p>v2000fichaatendimentoodontologicomaster.xsd v2000fichaatendimentoodontologicochild.xsd v2000procedimentoquantidade.xsd</p>
Atividade Coletiva
<ul style="list-style-type: none"> ● FichaAtividadeColetivaMasterThrift: entidade que organiza os dados de uma Atividade Coletiva. <ul style="list-style-type: none"> ○ ParticipanteRowItemThrift: entidade que organiza os itens da lista de participantes de uma Atividade Coletiva.
<p>XSD para suporte ao XML:</p> <p>v2000fichaatividadecoletiva.xsd v2000participanterowitem.xsd</p>
Visita Domiciliar
<ul style="list-style-type: none"> ● VisitaDomiciliarMasterThrift: entidade que organiza as Visitas Domiciliares realizadas por um profissional. <ul style="list-style-type: none"> ○ FichaVisitaDomiciliarChildThrift: entidade que organiza as Visitas Domiciliares realizadas a um cidadão, individualmente.

XSD para suporte ao XML:

v2000fichavisitadomiciliarmaster.xsd
v2000fichavisitadomiciliarchild.xsd

Marcadores de Consumo Alimentar

- **FichaConsumoAlimentar:** entidade que organiza as informações do questionário de Marcadores de Consumo Alimentar
 - **PerguntaQuestionarioCriançasMenoresSeisMeses:** bloco de respostas aplicado exclusivamente para crianças menores de seis meses
 - **PerguntaQuestionarioCriançasDeSeisVinteTresMeses:** bloco de respostas aplicado exclusivamente para crianças maiores de seis meses e até 24 meses não completos
 - **PerguntaQuestionarioCriançasComMaisDoisAnos:** bloco de respostas aplicado exclusivamente para crianças com 2 (dois) anos ou mais
 - **PerguntaCriançasMenoresSeisMesesEnum:** enumerador das questões para crianças menores de seis meses
 - **PerguntaCriançasDeSeisVinteTresMesesEnum:** para crianças maiores de seis meses e até 24 meses não completos
 - **PerguntaCriançasComMaisDoisAnosEnum:** para crianças com 2 (dois) anos ou mais
 - **RespostaMultiplaEscolhaEnum:** controlador de respostas múltiplas
 - **RespostaUnicaEscolhaEnum:** controlador de resposta única

XSD para suporte ao XML:

v2000fichaconsumoalimentar.xsd
v2000perguntaquestionariocriançascommaisdoisanos.xsd
v2000perguntaquestionariocriançasdeseisvintetresmeses.xsd
v2000perguntaquestionariocriançasmenoresseismeses.xsd
v2000respostamultiplaescolhaenum.xsd
v2000respostaunicaescolhaenum.xsd
v2000perguntacriançascommaisdoisanosenum.xsd
v2000perguntacriançasdeseisvintetresmesesenum.xsd
v2000perguntacriançasmenoresseismesesenum.xsd

Atendimento Domiciliar**Avaliação de Elegibilidade**

- **FichaAvaliacaoElegibilidadeThrift:** entidade que organiza as informações da Ficha de Avaliação de Elegibilidade, , exclusivo para Serviço de Atenção Domiciliar

<p>XSD para suporte ao XML: v2000fichaavaliacaoelegibilidade.xsd</p>
<p>Atendimento Domiciliar</p>
<ul style="list-style-type: none"> • FichaAtendimentoDomiciliarMasterThrift: entidade que organiza as informações da Ficha de Atendimento Domiciliar com as informações do profissional, exclusivo para Serviço de Atenção Domiciliar <ul style="list-style-type: none"> ○ FichaAtendimentoDomiciliarChildThrift: entidade que organiza o atendimento individualizado em cada ficha de atendimento
<p>XSD para suporte ao XML: v2000fichaatendimentodomiciliarmaster.xsd v2000fichaatendimentodomiciliarchild.xsd</p>
<p>Cadastro</p>
<p>Cadastro Domiciliar</p>
<ul style="list-style-type: none"> • CadastroDomiciliarMasterThrift: entidade que organiza os dados de um Cadastro Domiciliar. <ul style="list-style-type: none"> ○ EnderecoLocalPermanenciaThrift: entidade que organiza os dados de Endereço (Local de Permanência, para Situação de Rua) de um Cadastro Domiciliar. ○ CondicaoMoradiaThrift: entidade que organiza os dados de condição de moradia do Cadastro Domiciliar. ○ FamiliaRowThrift: entidade que organiza os itens da lista de famílias de um domicílio.
<p>XSD para suporte ao XML: v2000cadastrodomiciliar.xsd v2000enderecolocalpermanencia.xsd v2000condicaomoradia.xsd v2000familiarow.xsd</p>
<p>Cadastro Individual</p>
<ul style="list-style-type: none"> • CadastroIndividualMasterThrift: entidade que organiza os dados de Cadastro Individual do cidadão adscrito. <ul style="list-style-type: none"> ○ IdentificacaoUsuarioCidadaoThrift: entidade que organiza os dados de identificação do cidadão no Cadastro Individual. ○ InformacoesSocioDemograficasThrift: entidade que organiza as informações sociodemográficas do cidadão no Cadastro Individual.

- **CondicoesDeSaudeThrift:** entidade que organiza os dados de Cadastro Individual do cidadão adscrito.
- **EmSituacaoDeRuaThrift:** entidade que organiza os dados específicos sobre cidadão em situação de rua.

XSD para suporte ao XML:

v2000cadastroindividual.xsd
v2000identificacaousuariocidadao.xsd
v2000informacoessociodemograficas.xsd
v2000condicoesdesaude.xsd
v2000emsituacaoderua.xsd

Fonte: DAB/SAS/MS.

5.4 Esquema Geral de um Arquivo RAS

Como visto anteriormente, na seção [3.2.1](#), a estrutura do arquivo a ser enviado pode tomar várias formas, no entanto este deve compor essencialmente um arquivo zip com vários arquivos referenciando um registro/cadastro, podendo este registro pai (Master) estar organizando uma lista de registro do tipo filho (Child).

- um arquivo zip pode conter vários arquivos .esus ou .esus.xml.

-um arquivo .esus ou .esus.xml deve conter apenas uma camada de transporte.

-uma camada de transporte deve conter apenas uma ficha (que estará armazenada no campo dadoSerializado).

-Uma ficha pode ter zero, um ou vários campos child, conforme a descrição de cada ficha no dicionário de dados.

Na Figura 5.1 e 5.2, podemos ver estruturas esquemáticas dos arquivos gerados pela API RAS Thrift e em formato XML RAS, respectivamente. Veja que os arquivos ZIP gerados são um conjunto de “fichas” de atendimento ou de cadastros realizados pelo sistema próprio. Enquanto os arquivos ZIP agrupam fichas, as fichas podem agrupar atendimentos realizados por um mesmo profissional em um mesmo contexto (child). Para entender melhor o conceito de ficha consulte o [Manual de Preenchimento do Sistema com CDS](#).

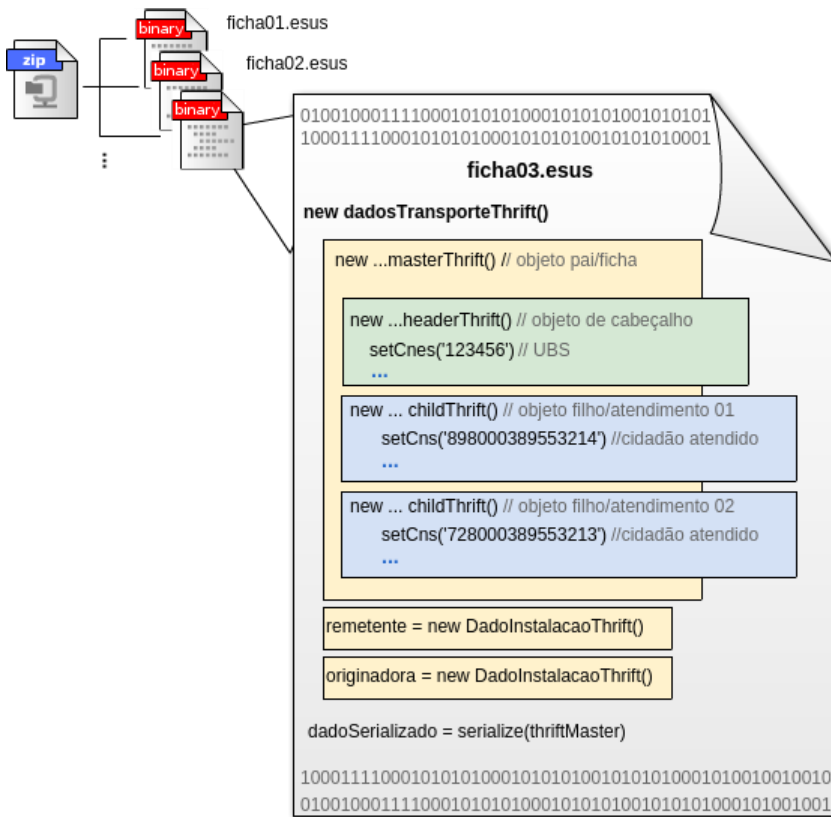


Figura 5.1 Esquema Didática de um Arquivo gerado pela API RAS Thrift

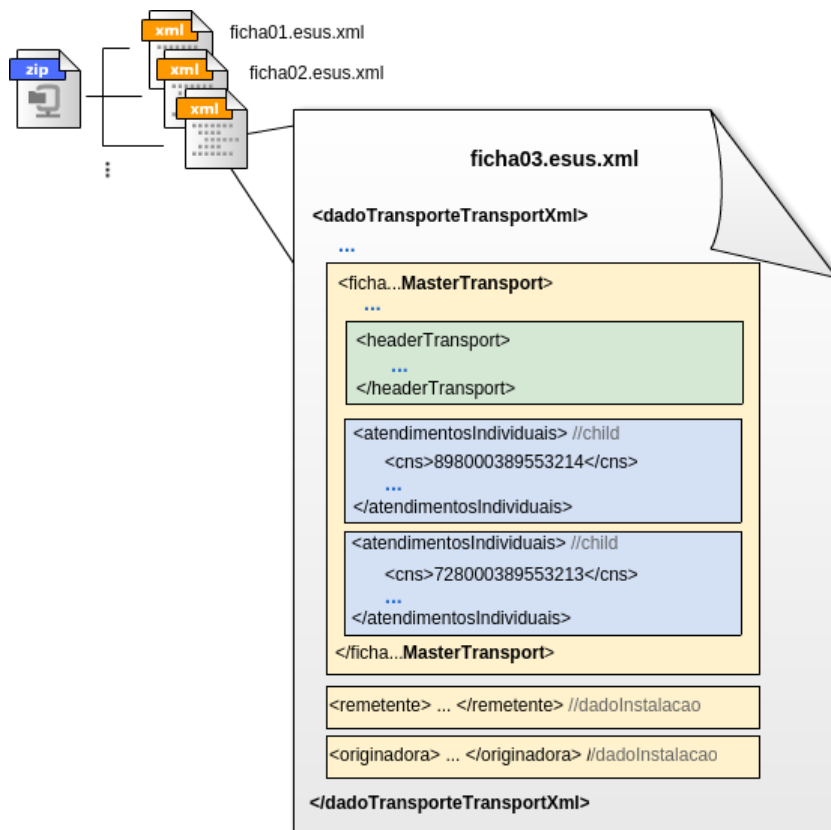


Figura 5.2 Esquema Didática de um Arquivo gerado em Formato XML RAS

5.5 Exemplos de uso da API Thrift RAS

Para auxiliar no desenvolvimento de soluções de integração a partir do uso das API e-SUS via Apache Thrift, foram implementados alguns exemplos comentados, disponíveis no endereço: <https://github.com/esusab/integracao/tree/master/thrift-exemplo>

5.6 Exemplo de uso do RAS exportado via XML

Para auxiliar no desenvolvimento de soluções de integração a partir do uso de XML no formato do e-SUS, foram implementados alguns exemplos comentados, disponíveis no endereço: <https://github.com/esusab/integracao/tree/master/xml-exemplo>

Conclusão

Este manual buscou trazer vários conceitos para que se torne mais fácil a implementação da exportação de dados para o Sistema e-SUS AB por meio das API Thrift geradas para integração com Sistema Próprios.

Para maiores detalhes consultar os manuais citados nas referência bibliográficas e em todo decorrer do texto.

Referências Bibliográficas

Apache Thrift Install - <http://thrift.apache.org/docs/install/>

Andrew Prunicki, Apache Thrift, 2009 Object Computing, Inc. EUA:
<http://jnb.ociweb.com/jnb/jnbJun2009.html>

Ministério da Saúde, Estratégia de e-Saúde para o Brasil, 2014 (em publicação)

Stratos Dimopoulos. Thrift Tutorial, 2013: <http://thrift-tutorial.readthedocs.org/>

Site do Apache Thrift - <http://thrift.apache.org>

Site do Sistema e-SUS Atenção Básica: <http://dab.saude.gov.br/esus>

Anexo I

Modelo de Troca de Informação do e-SUS AB