

Critical Software S.A.

Especificação da Arquitetura do Software

Sistema de Informação de Saúde
para Monitoria e Avaliação



©2014 Copyright Critical Software S.A. Todos Os Direitos Reservados.

REFERÊNCIA CONTRATO:
CSW-2012-PRL-02976

Código Projeto: SISMA
Ref. Doc.: CSWMZ-SISMA-2013-SAS-
Data: 2013-04-23
Páginas: 36
Estado: Aprovado
Acesso: Confidencial Projecto
Versão: 1.0



SALVAGUARDA - Documento sob contrato com Jembi .

O trabalho descrito no presente documento foi desenvolvido sob contrato com Jembi. A responsabilidade pelo seu conteúdo é do autor ou organização que o preparou.

Parceiros:



www.criticalsoftware.com

Aprovação				
Versão	Nome	Funções	Assinatura	Data
01	Paulo Grácio	Technical Manager		2013-06-12
01	Luis Gaspar	Gestor de Projecto		2013-06-12

Autores e Contribuintes			
Nome	Contacto	Descrição	Data
Délio Nhantumbo	danhantumbo@criticalsoftware.com	Autor	2013-04-24
Paulo Grácio	pgracio@criticalsoftware.com	Contribuinte	
Fábio Ramos	fd-ramos@criticalsoftware.com	Contribuinte	2013-05-14

Lista de Acesso	
Acesso Interno	
Dep. Qualidade, Equipa de Projeto	
Acesso Externo	
Equipa de projecto, Dr. Alessandro Campione, Mr. David Mendes, Mr Pascal Brandt, equipa MOASIS e Jembi.	
O conteúdo do presente documento é copyright da Critical Software S.A.; Este documento é distribuído sob condição de não ser copiado no seu todo ou em parte, ou reproduzido de qualquer outra forma (seja por meio fotográfico ou qualquer outro meio) e o seu conteúdo não deverá ser em caso algum divulgado a qualquer pessoa ou entidade não referida nas listas de acesso interna e externa sem prévio consentimento por escrito das entidades envolvidas.	

Histórico de Revisões				
Versão	Revisão	Data	Descrição	Autor
	0.01	2013-04-23	Criação do documento	Délio Nhantumbo
	0.02	2013-05-	Revisão do documento	Gulam Khan

Histórico de Revisões

		07		
	0.03	2013-05-07	Actualização do documento/ Revisão	Délio Nhantumbo
	0.04	2013-05-17	Revisão do documento	Fábio Ramos
	0.05	2013-05-17		Délio Nhantumbo
01		2013-06-12	Versão enviada ao cliente.	Luis Gaspar

Índice

1	Introdução	6
1.1	Objetivo	6
1.2	Âmbito	6
1.3	Audiência	6
1.4	Definições e acrónimos	6
1.5	Estrutura do documento	8
1.6	Documentos aplicáveis	8
1.7	Documentos de referência	8
2	Normas, Convenções, Procedimentos e Ferramentas	10
2.1	Standards de design	10
2.2	Ferramentas de software	10
2.3	Tecnologias	10
3	Arquitectura Conceptual	12
3.1	Camada de Apresentação	13
3.2	Camada de Negócio (Lógica)	13
3.2.1	Serviço JDBC.....	13
3.2.1.1	<i>BatchHandler</i>	14
3.2.1.2	<i>JdbcConfiguration</i>	14
3.2.1.3	<i>StatementBuilder</i>	14
3.2.1.4	<i>IdentifierExtractor</i>	14
3.2.1.5	<i>StatementHolder</i>	15
3.2.2	Importação-Exportação.....	15
3.2.2.1	Serviço de exportação	16
3.2.2.2	Serviço de importação.....	16
3.2.2.3	Gestão de serviços de importação exportação	16
3.2.3	Módulo de Relatórios.....	16
3.2.3.1	Tabela de relatórios	17
3.2.3.2	Gráfico	18
3.2.3.3	Relatórios de Completude	19
3.2.3.4	Documentos	21
3.2.3.5	Pivot table.....	21
3.2.3.6	O módulo externo.....	21
3.2.3.7	Módulo de Apoio ao Sistema.....	21
3.3	Camada de Persistência de Dados	21
3.4	Interoperabilidade	22
3.4.1	Mecanismo de integração	23
3.4.2	Requisitos de Integração	24
3.4.3	Modos de integração	24
3.4.3.1	Integração Manual (Import/Export).....	24
3.4.3.2	Integração Automática	25
3.4.4	Sistemas para Integração.....	26
3.4.5	Gestão de Unidades Sanitárias.....	28
3.5	MyDatamart.....	30
3.6	Autenticação e Autorização	31
3.7	Logs e Auditoria	32
4	Estratégias de instalação	35

4.1	Instalação Online	35
4.2	Instalação Offline	35
5	Desenho da base de dados	36
5.1	Núcleo do Sistema	36
6	Questões em aberto	38

Índice de Tabelas

Tabela 1: Definições	6
Tabela 2: Acrónimos	7
Tabela 3: Documentos aplicáveis	8
Tabela 4: Documentos de referência	9
Tabela 5: Sistemas para integração	27

Índice de Figuras

Figura 1: Arquitectura conceptual	12
Figura 2: Diagrama JDBC Batch Handler	14
Figura 3: Diagrama JDBC StatemantManager	15
Figura 4: Serviço de Importação e Exportação	16
Figura 5: Tabelas de Relatórios	17
Figura 6: Definição de Tipo e Dimensões do Gráfico	18
Figura 7: Gráfico de Barras	19
Figura 8: Registro de relatórios de completude	20
Figura 9: Documentos	21
Figura 10: Persistência de dados	22
Figura 11: Importação/Exportação Manual	25
Figura 12: Integração Automática	26
Figura 13: Integração de Sistemas	28
Figura 14: Operações <i>Facility Registry API</i>	29
Figura 15: Facilities	30
Figura 16-MyDatamart	31
Figura 17: Ecrã de Autenticação	32
Figura 18: Estrutura de valor de dados	36
Figura 19: Diagrama de núcleo do sistema	37

1 Introdução

1.1 OBJETIVO

Este documento apresenta a arquitectura funcional para o sistema SIS-MA - Sistema de Informação de Saúde para Monitoria e Avaliação.

O nível de detalhe que o documento apresenta tem como principal objectivo fornecer às pessoas que irão estar envolvidas na implementação do SIS-MA, uma base estável para o entendimento da arquitectura do sistema baseado em DHIS2.

1.2 ÂMBITO

Este documento apresenta a arquitectura do sistema SISMA de acordo com o estabelecido para o entregável *DL-05 Especificação de Arquitectura do SIS-MA e disponibilização do sistema*.

1.3 AUDIÊNCIA

A audiência deste documento é constituída pelas entidades Jembi, MOSAIS e MISAU responsáveis pela especificação e validação do sistema, bem como pela equipa de projecto que venha a ser designada para participar no projecto e pelos elementos da Critical Software que serão responsáveis pelos trabalhos a desenvolver.

1.4 DEFINIÇÕES E ACRÓNIMOS

A Tabela 1 apresenta a lista de definições usadas neste documento. As definições apresentadas em [AD-1] são também aplicáveis.

Nome	Descrição
Documento Aplicável	Um documento é considerado aplicável se complementar este documento. Todo o seu conteúdo é diretamente aplicável como se tivesse sido escrito como anexo deste documento.
Documento de Referência	Um documento é considerado de referência se neste documento existem referências ao seu conteúdo, mas este não é totalmente aplicável. Os documentos de referência são principalmente utilizados para fornecer leitura e recursos adicionais.

Tabela 1: Definições

A Tabela 2 apresenta a lista de acrónimos usados neste documento. Os acrónimos apresentados em [AD-1] são também aplicáveis.

Acrónimo	Descrição
AD	Applicable Document (documento aplicável)
BIRT	Business Intelligence and Reporting Tools

Acrônimo	Descrição
CRUD	Create, read, update and delete
CSV	Comma Separated Values
CSW	Critical Software, S.A.
DHIS	District Health Information System
DXF	DHIS eXchange Format
GUI	Graphical User Interface
HTML	HyperText Markup Language
IXF	Indicator eXchange Format
JDBC	Java DataBase Connectivity
JSON	JavaScript Object Notation
MVC	Model View Controller
PDF	Portable Document Format
PNG	Portable Network Graphics
RD	Reference Document (documento de referência)
SDMX	Statistical Data and Metadata eXchange
SGBD	Sistema de Gestão de Base de Dados
SISMA	Sistema de Informação de Saúde para Monitoria e Avaliação
TBC	To be confirmed (a confirmar)
TBD	To be defined (a definir)
URL	Uniform Resource Locator
XML	eXtensible Markup Language
XSLT	Extensible Stylesheet Language Transformations
IoC	Inversion of Control

Tabela 2: Acrónimos

1.5 ESTRUTURA DO DOCUMENTO

A secção 1 (**Error! Reference source not found.**) apresenta uma descrição geral do conteúdo deste documento.

A secção 2 (Normas, Convenções, Procedimentos e Ferramentas) apresenta o enquadramento de standards, convenções e procedimentos utilizados para atingir um alto nível de homogeneidade nos resultados de todos os projectos, promover a adesão a boas práticas, maximizar capacidades de reutilização, minimizar riscos, manter custos e assegurar alta qualidade dos produtos

A secção 3 (Arquitetura Conceptual) apresenta a arquitectura conceptual do SIS-MA tendo com base a arquitectura DHIS2, faz a descrição das 3 camadas (camada de apresentação, da lógica de negócio e de persistência de dados), descreve o processo de interoperabilidade e ainda descreve o *MyDatamart*.

A secção 4 (Estratégias de instalação) apresenta e descreve as estratégias de instalação possíveis para o SIS-MA, modo *online* e *offline*, de forma a responder às diferentes necessidades de Moçambique.

A secção 5 (Desenho da base de dados) apresenta a estrutura de dados da aplicação, faz referência a tabelas importantes de suporte às funcionalidades chave para os diferentes Programas de Saúde do MISAU.

A secção 6 (Questões em aberto) descreve os diversos assuntos pendentes.

1.6 DOCUMENTOS APLICÁVEIS

A Tabela 3 apresenta a lista de documentos aplicáveis a este documento.

Documento aplicável	Número do documento
[AD-1] Proposta Sistema de Informação de Saúde para Monitoria e Avaliação, 2012-07-30, Critical Software SA	CSW-2012-PRL-02976-01
[AD-2] Adenda à Proposta Sistema de Informação de Saúde para Monitoria e Avaliação, 2012-12-18, Critical Software SA	CSW-2012-PRL-02976-05

Tabela 3: Documentos aplicáveis

1.7 DOCUMENTOS DE REFERÊNCIA

A Tabela 4 apresenta a lista de documentos de referência a este documento.

Documento de referência	Número do documento
[RD-1] Ferramentas de Software	CSWMZ-SISMA-2013-MMO-00072
[RD-2] Especificação de Requisitos Funcionais	CSWMZ-SISMA-2013-SRS-00085

Documento de referência	Número do documento
[RD-3] DHIS2 User Manual	731, Version 2.11 2013-04-24 15:03:58

Tabela 4: Documentos de referência

2 Normas, Convenções, Procedimentos e Ferramentas

Para atingir um alto nível de homogeneidade nos resultados de todos os projectos, promover a adesão a boas práticas, maximizar capacidades de reutilização, minimizar riscos, manter custos e assegurar alta qualidade dos produtos, a Critical Software, S.A. reforça os *standards* do Processo de Desenvolvimento de Software.

As secções seguintes listam os *standards*, convenções, procedimentos e ferramentas que devem ser seguidas e utilizadas neste projecto.

2.1 STANDARDS DE DESIGN

Este projecto tendo a sua base no sistema DHIS2 deverá seguir as recomendações da comunidade de forma a manter a homogeneidade do produto. Deve ser evitado o distanciamento do produto base, para que no futuro o MISAU possa beneficiar da evolução do DHIS2.

Sendo uma arquitectura de 3 níveis, é importante assegurar que em cada nível não são realizadas operações fora de contexto, mantendo as responsabilidades:

- Base de Dados para persistência de informação;
- Serviços para lógica de negócio;
- Módulos WEB para interfaces, GUI.

Mantendo esta abordagem será assegurado que os princípios base da arquitectura são mantidos sem que tal comprometa a arquitectura da solução. O envolvimento da comunidade DHIS2 tem sido assegurado com reuniões técnicas nas quais são envolvidos Lars Helge Øverland - Coordenador Técnico de Desenvolvimento, Ola Hodne Titlestad - Gestor de Projecto e Zeferino Saugene - Estudante de Doutoramento.

2.2 FERRAMENTAS DE SOFTWARE

As ferramentas de projecto estão identificadas de forma detalhada no documento CSWMZ-SISMA-2013-MM0-00072 **Error! Reference source not found..**

2.3 TECNOLOGIAS

As seguintes tecnologias são usadas no desenvolvimento do DHIS2, tecnologias essas que vão fazer parte do sistema SISMA:

- Hibernate (www.hibernate.org)
- Spring (www.springframework.org)
- Struts (struts.apache.org)
- Velocity (www.velocity.apache.org)
- Commons (www.commons.apache.org)
- JasperReports (jasperforge.org/projects/jasperreports)

- JFreeChart (www.jfree.org/jfreechart/)
- JUnit (www.junit.org)

Como suporte ao desenvolvimento é utilizado o Maven para gestão do processo de compilação e o Bazaar para controlo de versões do código fonte.

- Maven (apache.maven.org)
- Bazaar (bazaar-vcs.org)

3 Arquitetura Conceptual

O SIS-MA, através da utilização de DHIS2, é um sistema de informação baseado em dados de domínio da saúde e que permite a captura, agregação, análise e produção de indicadores e relatórios.

O sistema é desenvolvido em *Java EE – Java Enterprise Edition*, que é uma tecnologia utilizada para desenvolvimento de aplicações de alta robustez, através de uma arquitetura de três camadas:

- A camada de apresentação é baseada na web.
- A camada de Negócio (Lógica).
- Camada de persistência de dados.

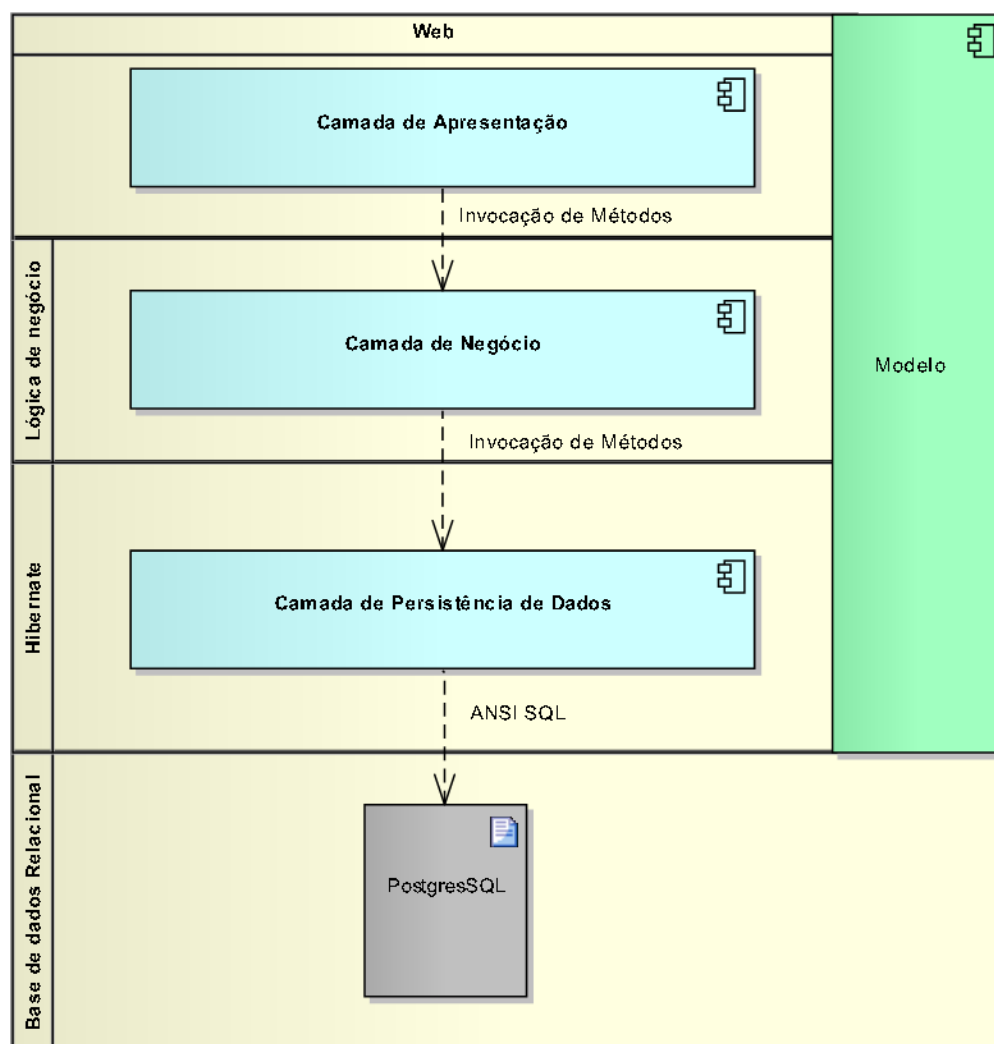


Figura 1: Arquitetura conceptual

3.1 CAMADA DE APRESENTAÇÃO

A camada de apresentação do SIS-MA é baseada na web. O sistema pode ser utilizado on-line bem como stand-alone. Esta camada é constituída por uma estrutura modular, onde se destacam dhis-web-report, dhis-web-maintenance, dhis-web-validationrule, entre outros, que são agregados no módulo web do portal (dhis-web-portal). Esta estrutura permite que cada módulo tenha seu próprio domínio e que os desenvolvimentos sejam feitos de forma contida. Os módulos web são baseados em Struts e seguem os padrões MVC. É utilizado o Maven, que é uma ferramenta de gestão e compilação que permite a gestão de dependências de packages, para *layout* de directórias o que implica que os ficheiros são separados e guardados em directórias específicas.

3.2 CAMADA DE NEGÓCIO (LÓGICA)

Esta camada é responsável pela definição de toda lógica de negócios. Todas as principais classes, como as responsáveis pela persistência, lógica de negócios e apresentação, são mapeadas como *Spring managed beans*. Um *bean* é um objecto que é instanciado, injectado e administrado por um *IoC container Spring*. As dependências entre os *beans* são injectados pelo *container IoC*, o que permite ter baixas ligações, reconfiguração e testes. Esta camada lida com troca de informações entre uma base de dados e a interface de utilizador.

Os serviços encontrados no módulo dhis-service-core, basicamente, fornecem métodos que delegam para um método correspondente na camada de persistência, ou contém lógica simples e auto-explicativa. Por causa da sua complexidade, alguns serviços como existente nos módulos dhis-service-datamart, dhis-service-import-export, dhis-service-jdbc, e dhis-service-reporting são detalhados nas sub-seções abaixo (4.2.1, 4.2.2 e seguintes).

3.2.1 SERVIÇO JDBC

O serviço JDBC contém um conjunto de componentes que lidam com conexões JDBC e instruções SQL.

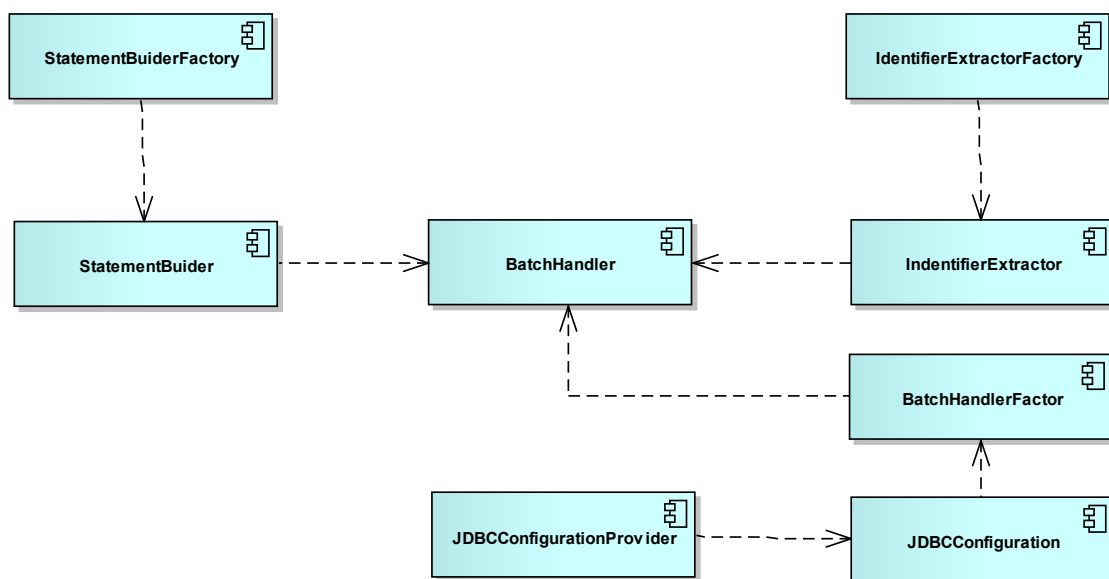


Figura 2: Diagrama JDBC Batch Handler

3.2.1.1 BatchHandler

A interface *BatchHandler* é responsável pelo fornecimento de métodos de inserção, actualização e verificação de existência de objectos. Este interface tem como objectivo proporcionar operações de alto desempenho e é bastante útil quando se trabalha com maior volume de dados e, ainda, tem a capacidade de gerir objectos de inserção de milhares de dados em cada *commit* na base de dados de forma transparente. Este interface, é obtido a partir do *BatchHandlerFactory*.

3.2.1.2 JdbcConfiguration

Este interface contém informações sobre a configuração actual do *SGBD JDBC*, mais especificamente as *driver class*, ligação *URL*, nome utilizador e *password*. O objecto desta interface é obtido a partir do *JdbcConfigurationProvider* componente, que actualmente usa o provedor de configuração *Hibernate* interna para transmitir a informação.

3.2.1.3 StatementBuilder

Este interface é responsável por disponibilizar métodos que contém de instruções *SQL* e, é obtido a partir do *StatementBuilderFactory*, que tem capacidades para determinar o tempo de execução actual do *SGBD* e proporcionar uma implementação apropriada.

3.2.1.4 IdentifierExtractor

Este interface fornece métodos para recuperar os últimos identificadores gerados a partir da *SGBD*. O mesmo é obtido a partir do *IdentifierExtractorFactory*, que é capaz de determinar o tempo de execução do *SGBD* e proporcionar uma implementação apropriada.

3.2.1.5 StatementHolder

Esta interface faz a manutenção e fornecimento de ligações *JDBC* e *statements*. Pode ser obtido a partir do componente *StatementManager* que pode ser inicializado usando o *initialise()* e método *destroy()* para fechar ligação. Quando inicializado, abre uma ligação a base de dados e mantém a variável numa *thread*, *ThreadLocal*, o que significa que para todos os pedidos efectuados para a interface, ele retornará a mesma instância o que de certo modo melhora o desempenho da própria base de dados ou *statements* podem ser reutilizadas para múltiplas operações. O *StatementManager* é normalmente utilizado na camada de persistência de dados para as classes que trabalham directamente com *JDBC*, como o *DataMartStore*.

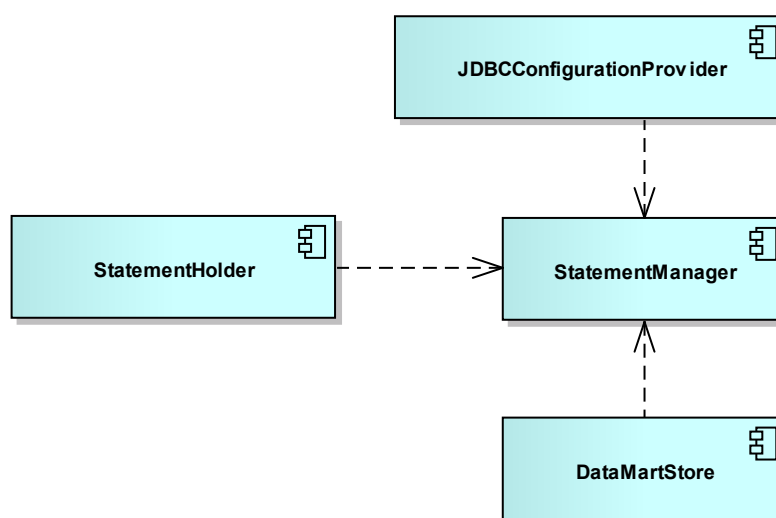


Figura 3: Diagrama JDBC StatementManager

3.2.2 IMPORTAÇÃO-EXPORTAÇÃO

O módulo de importação e exportação contém classes responsáveis por produzir e consumir ficheiros com formato específico. O processo de exportação, exporta os dados da base de dados para um determinado repositório e segundo os formatos abaixo indicados, estes dados, podem ser posteriormente importados em uma máquina que funciona em modo *offline*. O Processo de importação tem três variantes que são:

- Importação – importa os dados directamente para a base de dados
- Visualização – importa os dados para um local temporário, e permitir que o utilizador faça a filtragem e eventualmente importar dados.
- Análise – Revela anomalias nos dados de importação.

Os formatos suportados são os seguintes: DXF, IXF, DHIS 1.4 XML format, DHIS 1.4 Datafile format, CSV e PDF.

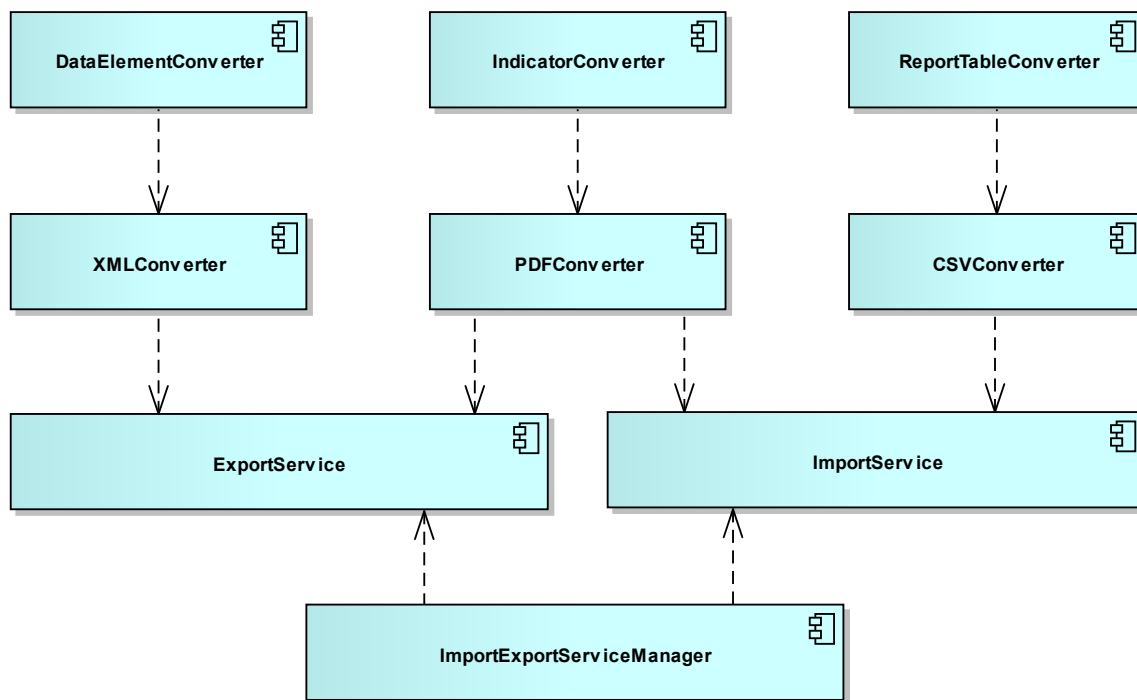


Figura 4: Serviço de Importação e Exportação

3.2.2.1 Serviço de exportação

A interface *ExportService* é responsável por instanciar conversores apropriados e invocar métodos de exportação. Para evitar que pedidos longos fiquem propensos ao limite de tempo, *timeout-errors*, na camada de apresentação, o trabalho real de exportação é efectuada numa *thread* separada. O serviço de exportação regista os seus conversores na classe *ExportThread* usando o método *registerXmlConverter* (*XmlConverter*) e inicia a execução.

3.2.2.2 Serviço de importação

A interface *ImportService* é responsável por instanciar conversores adequados e invocar métodos de importação. O processo de importação usa de forma consistente a interface *BatchHandler*.

3.2.2.3 Gestão de serviços de importação exportação

A interface *ImportExportServiceManager* fornece métodos para recuperar todos os *ImportServices* e *ExportServices*, bem como a recuperação destes com base num formato chave. Isto torna mais simples para recuperar o serviço correcto usando classes porque o nome do formato é utilizado como parâmetro para obter uma instância do serviço correspondente.

3.2.3 MÓDULO DE RELATÓRIOS

Este módulo contém componentes relacionados com a produção de relatórios descritos abaixo:

3.2.3.1 Tabela de relatórios

O objecto *ReportTable* representa uma tabela da base de dados de referências cruzadas. A tabela pode ser referenciada em qualquer quantidade das três dimensões, que são:

- Dimensão descritiva – pode conter elementos de dados, indicadores, ou conjunto de dados de completude.
- Dimensão período.
- Dimensão unidade organizacional.

O objectivo é ser capaz de personalizar tabelas para depois usar ferramentas de relatórios de terceiros como o *JasperReport* ou directamente nos formatos de saída como PDF ou HTML dentro do sistema.

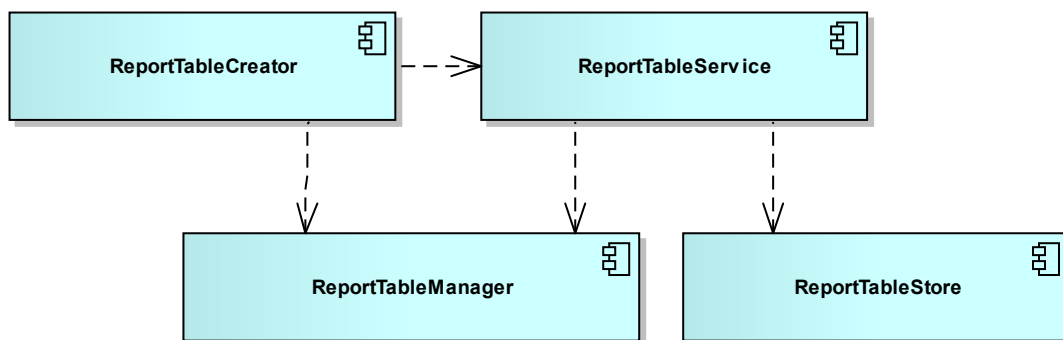


Figura 5: Tabelas de Relatórios

3.2.3.1.1 ReportTableStore

É responsável por persistir objectos *ReportTable* e, possuem, uma implementação em *Hibernate*.

3.2.3.1.2 ReportTableService

É responsável por executar toda lógica de negócios para relatórios tabelas, como a geração de períodos relativos, bem como delegar operações de CRUD ao *ReportTableStore*.

3.2.3.1.3 ReportTableManager

O *ReportTableManager* é responsável pela criação e remoção de tabelas do relatório, bem como a recuperação de dados.

3.2.3.1.4 ReportTableCreator

O *ReportTableCreator* é o componente chave, e é responsável por:

- Exportação de dados relevantes para o datamart usando o *DataMartExportService* e ou *DataSetCompletenessService*. Os dados são posteriormente recuperados e usados para preencher a tabela do relatório.

- Criar tabelas de relatório através do *ReportTableManager*.
- Incluir possíveis valores de regressão.
- Preencher a tabela de relatório usando o *BatchHandler*.
- Remover a tabela de relatório usando o *ReportTableManager*.

3.2.3.2 Gráfico

O objecto *chart* (*gráfico*) representa as preferências de gráficos. Os gráficos são baseados em períodos ou unidade organizacional. Um Gráfico tem três dimensões, nomeadamente, *valor*, *categoria*, e *filtro*. A dimensão valor pode conter qualquer número de indicadores.

Num gráfico baseado no período, a dimensão categoria possui qualquer número de períodos enquanto a dimensão filtro contém uma única unidade organizacional.

Num gráfico baseado em unidade organizacional, a dimensão categoria contém qualquer número de unidades organizacionais enquanto a dimensão filtro contém um único período. Estão disponíveis 2 tipos de gráficos nomeadamente o gráfico de barras e o de linha.



Figura 6: Definição de Tipo e Dimensões do Gráfico

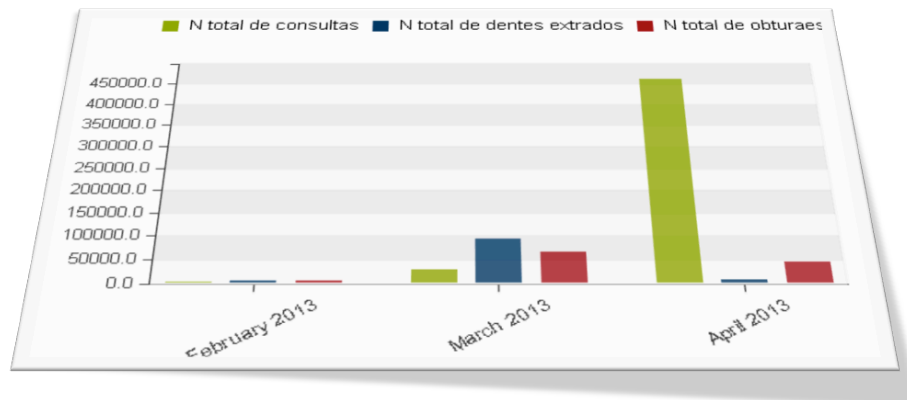


Figura 7: Gráfico de Barras

3.2.3.3 Relatórios de Completude

O objectivo principal da funcionalidade de completude do “data set” é guardar o número de “data sets” que tenham sido completados. Permite ao utilizador marcar o actual “data set” como completo. Esta funcionalidade disponibiliza também a percentagem da completude com base de relatórios de “data sets” da unidade organizacional completos comparando com o número total do relatório de “data set” da unidade organizacional. Disponibiliza também o número de relatórios de “data set” completado dentro do prazo.

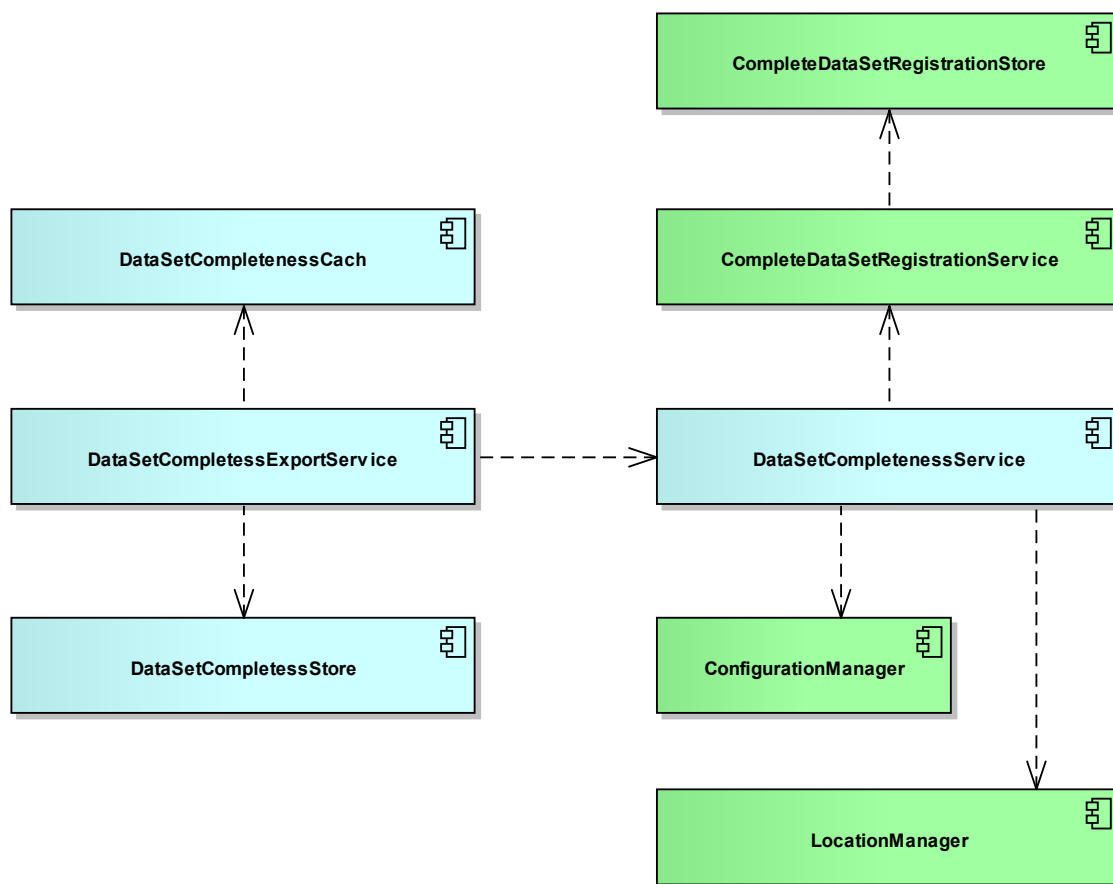


Figura 8: Registro de relatórios de completude

3.2.3.3.1 Registro de Completude de Dados

O objecto “*CompleteDataSetRegistration*” representa um “data set” marcado como completo por um utilizador. Esta propriedade guarda o “data set”, período, unidade organizacional e data. O “*CompleteDataSetRegistrationStore*” é responsável pela persistência de objectos “*CompleteDataSetRegistration*”. Disponibiliza métodos que retornam colecções de objectos requisitados por diferentes variantes de “data sets”, períodos e unidades organizacionais como parâmetros de entrada. O serviço “*CompleteDataSetRegistrationService*” é um método responsável por invocar a camada “store”.

3.2.3.3.2 Serviço de exportação de Completude de Dados

O *DataSetCompletenessExportService* é responsável por escrever *DataSetCompletenessResults* numa tabela de base de dados chamada *aggregateddatasetcompleteness*. Esta funcionalidade é considerada como fazendo parte de dados marcados. Estes dados podem ser utilizados tanto dentro do DHIS2 bem como por aplicações de relatórios de terceiros como o MS Excel.

3.2.3.4 Documentos

O objecto de documento representa um documento que é enviado para o sistema ou um *URL*. O *DocumentStore* é responsável por persistir objectos de documentos, enquanto o *DocumentService* é responsável pela lógica de negócios.

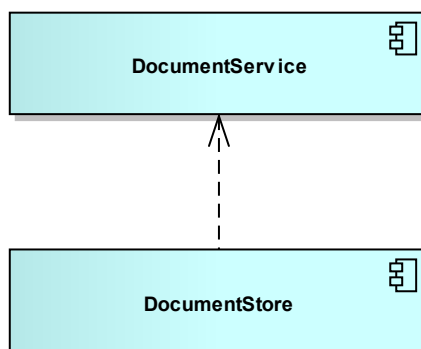


Figura 9: Documentos

3.2.3.5 Pivot table

Uma *Pivot table* representa uma tabela dinâmica que pode armazenar qualquer número de indicadores, períodos, unidades organizacionais e agregação correspondente a valores dos indicadores.

3.2.3.6 O módulo externo

O componente *LocationManager* é responsável pela comunicação entre DHIS2 e o *file system* do sistema operativo. Ele contém métodos que fornecem acesso de leitura de ficheiros através do método *InputStream* e, acesso de escrita de ficheiros no sistema operativo através do método *OutputStream*.

3.2.3.7 Módulo de Apoio ao Sistema

O projecto de apoio sistema (*system support project*) contém classes de suporte que são gerais e podem ser reutilizados ao longo de todo sistema.

3.3 CAMADA DE PERSISTÊNCIA DE DADOS

A camada de persistência de dados tem como base o *hibernate* que é capaz de correr em qualquer grande SGBD. O *hibernate* abstrai o SGBD subjacente, neste caso o PostgreSQL, e permite uma definição clara e fácil de propriedades de conexão à base de dados num ficheiro de propriedades denominado *hibernate.properties*. O sistema baseia-se na integração de *Spring-Hibernate*. Par

Objectos mais importantes têm suas correspondentes na *store* de implementação do *Hibernate*. Esta localização fornece métodos para as operações CRUD e de consultas para os diferentes tipos de objectos.

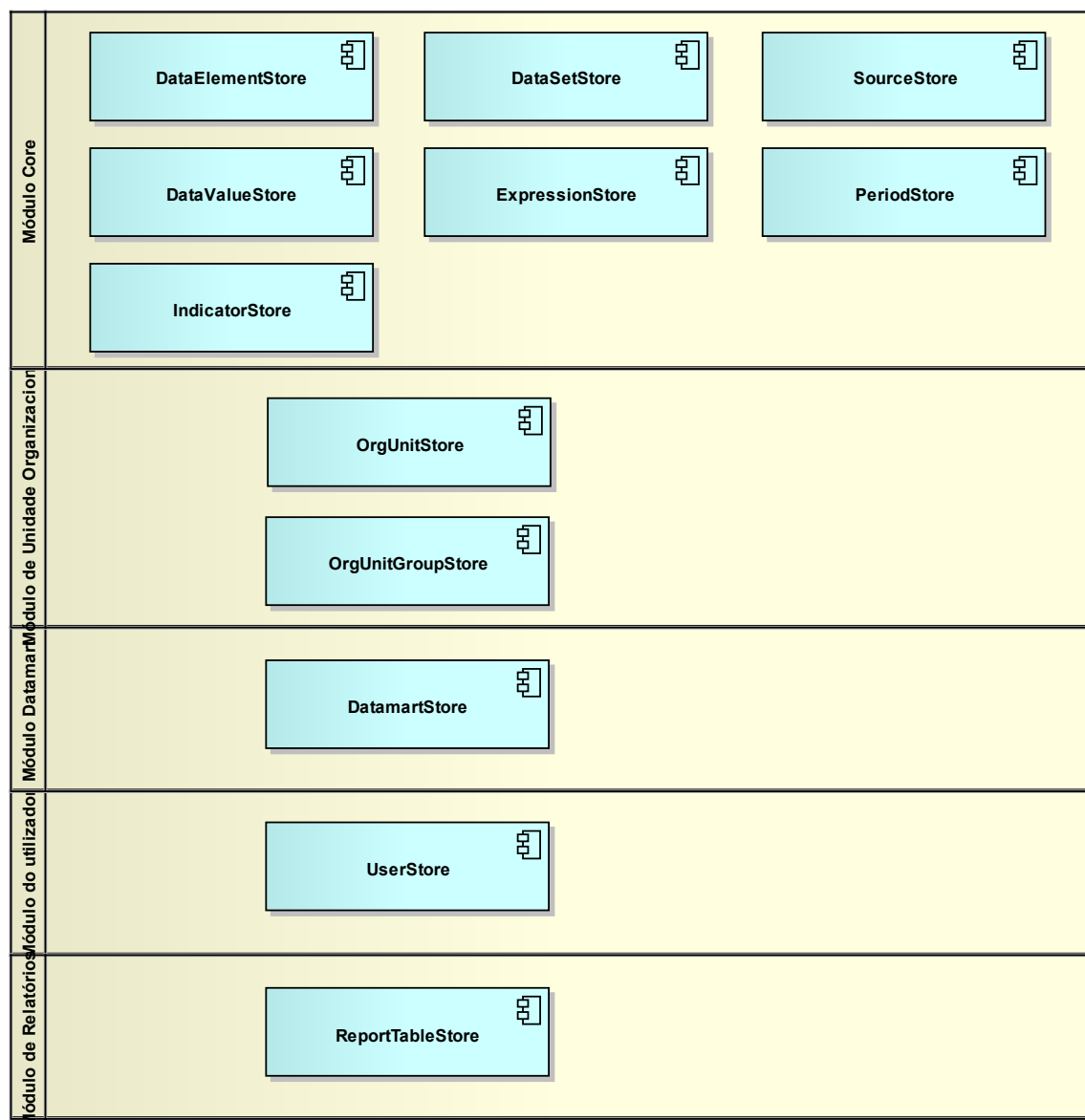


Figura 10: Persistência de dados

3.4 INTEROPERABILIDADE

A interoperabilidade visa a troca de informação entre sistemas tendo em vista a melhoria da qualidade de dados, melhoria na análise de dados através da integração e adoção de *standards* de agregação de dados.

3.4.1 MECANISMO DE INTEGRAÇÃO

O DHIS2 foi construído reutilizando um componente para melhor gerir diversos cenários de integração. Esta funcionalidade é fornecida pelo *Apache Camel*¹ que fornece uma implementação de *Enterprise Integration Patterns - EIP*. Esta *framework* complementa a API Web, fornecendo os meios para consumir serviços de outros sistemas realizar operações de troca de informação entre sistemas. Porque os requisitos de integração são sempre dependentes de contextos específicos, o mecanismo de integração foi desenhado para ser configurado na própria aplicação.



Este mecanismo de Integração é recomendado face à solução Mith Connect. Sendo a principal vantagem do Mirth Connect o suporte de mensagens HL7 versão 2.x e 3.0, X12, EDI, que não faz parte dos requisitos para a plataforma, a utilização do Motor de Integração DHIS2 será a mais adequada pois é uma solução já acoplada no sistema.

O mecanismo de integração assenta na definição de rotas. A rota define o fluxo de uma mensagem específica de externo no DHIS2. As rotas podem consistir em *timers*, pedidos de clientes através do protocolo *HTTP*, transformações, processamento de mensagens, etc. As rotas podem ser definidos em XML ou em Java. Rotas XML têm a vantagem de serem facilmente configurados e carregadas em um a instância DHIS2 em execução. Cenário de integração mais complexa pode requer a utilização da linguagem de programação Java. As rotas escritas em Java precisam de ser disponibilizados no *Classpath* da aplicação *web* durante a inicialização da aplicação.

Abaixo um exemplo usado internamente pelo módulo de integração do DHIS2 para efectuar a transformação de um *input* SDMX para o formato DXF2. O código está simplificado para uma melhor leitura do mesmo.

```
package org.hisp.dhis.integration.routes;
// imports not shown ....
public class SDMXDataIn extends RouteBuilder {
    @Override
    public void configure() throws Exception {
        DescriptionDefinition desc = new DescriptionDefinition();
        desc.setText("Sdmx data input");
        RouteDefinition sdmxDataIn = from("direct:sdmxDataIn").
            convertBodyTo(java.lang.String.class, "UTF-8").
            to("xslt:transform/cross2dxf2.xsl").
            convertBodyTo(java.io.InputStream.class).
            inOut("dhis2:data?orgUnitIdScheme=CODE&dataElementIdScheme=CODE&importStrategy=NEW_AND_UPDATES");
        sdmxDataIn.setDescription(desc);
        sdmxDataIn.setId("internal-sdmx-input");
    }
}
```

Algumas considerações sobre o código apresentado:

¹ <http://www.eaipatterns.com/>

- A integração DHIS2 está configurada para pesquisar e carregar de forma automática as rotas Java que se encontram na package `org.hisp.dhis.integration.routes`. Assim, qualquer `RouteBuilder` class como a apresentada anteriormente existente no Classpath será automaticamente carregada.
- Este exemplo apresenta um caso em que é utilizada uma transformação XSLT. Através do url `"xslt:transform/cross2dx2.xsl"` é feita uma pesquisa do ficheiro no classpath. É possível usar também referências absolutas para os ficheiros se assim se pretender.
- O endpoint `dhis2:data` é um componente específico que pode ser usado para importar dados usando o formato DXF2;

3.4.2 REQUISITOS DE INTEGRAÇÃO

Durante a fase 1, fase de estudo, foram definidos os seguintes requisitos que devem ser cumpridos pelos potenciais sistemas a estabelecerem troca de informação com o SISMA:

- Os sistemas devem ser reconhecidos pelo Ministério Saúde como oficiais;
- Os sistemas a serem integrados devem suportar *standards* integração do SIS-MA, SDMX-HD de acordo com:
 - *eGovernment Interoperability Framework for Mozambique*²
 - SESP³
- Para cada sistema a integrar deverá estar identificado, de forma detalhada, qual a informação a armazenar, Definição de formulários no SIS-MA.

3.4.3 MODOS DE INTEGRAÇÃO

A integração com outros sistemas deve ser efectuada através dos modos seguintes:

- Manual – Import/Export
- Automático – comunicação máquina/máquina através da API Web

3.4.3.1 Integração Manual (Import/Export)

Os dados devem ser importados/exportados de e para o servidor a nível Nacional, por um utilizador com competências técnicas para a execução da actividade, por meio de um dispositivo de armazenamento

² UTICT – Unidade Técnica de Implementação da Política de Informação, Versão Outubro de 2010.

³ Para a comunicação de dados clínicos agregados, recomenda-se o uso de Statistical Data and Metadata Exchange (SDMX) para a área de Saúde (SDMX-HD)

de dados (CD/DVD, USB) e, os mesmos, devem ser importados/exportados de e para o sistema externo através dos mesmos mecanismos de armazenamento por um utilizador com competências técnicas para a execução da actividade.

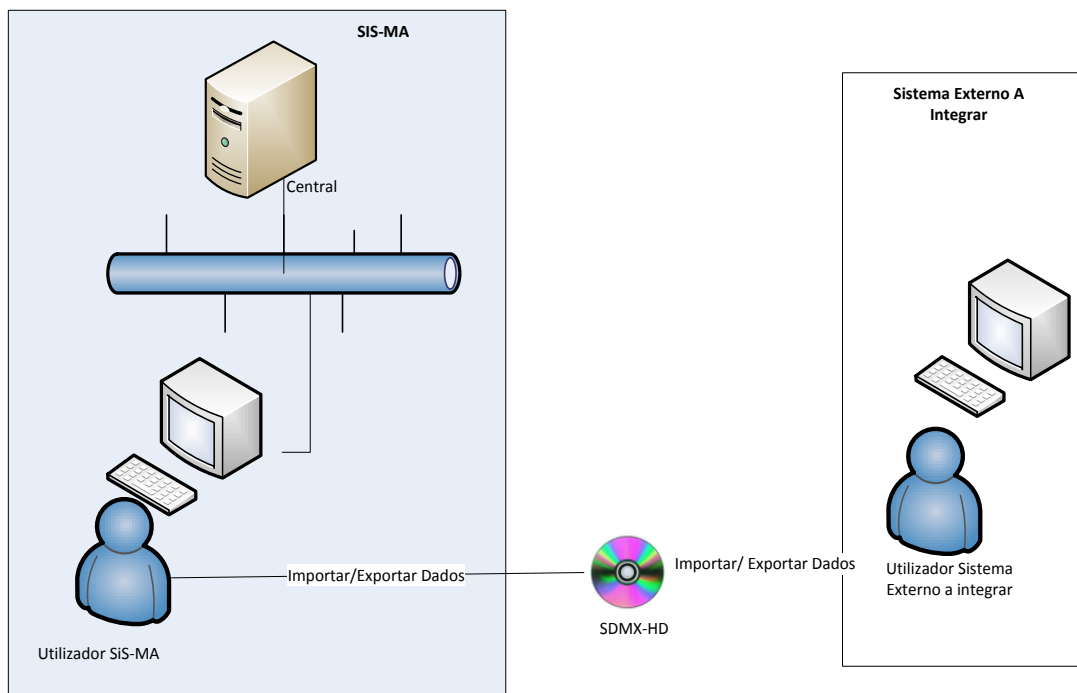


Figura 11: Importação/Exportação Manual

3.4.3.2 Integração Automática

Na integração automática, os dados serão transferidos entre os diferentes sistemas a integrar no SIS-MA via API Web através de protocolos de comunicação existentes como o caso do HTTP, HTTPS segundo um determinado formato de dados.

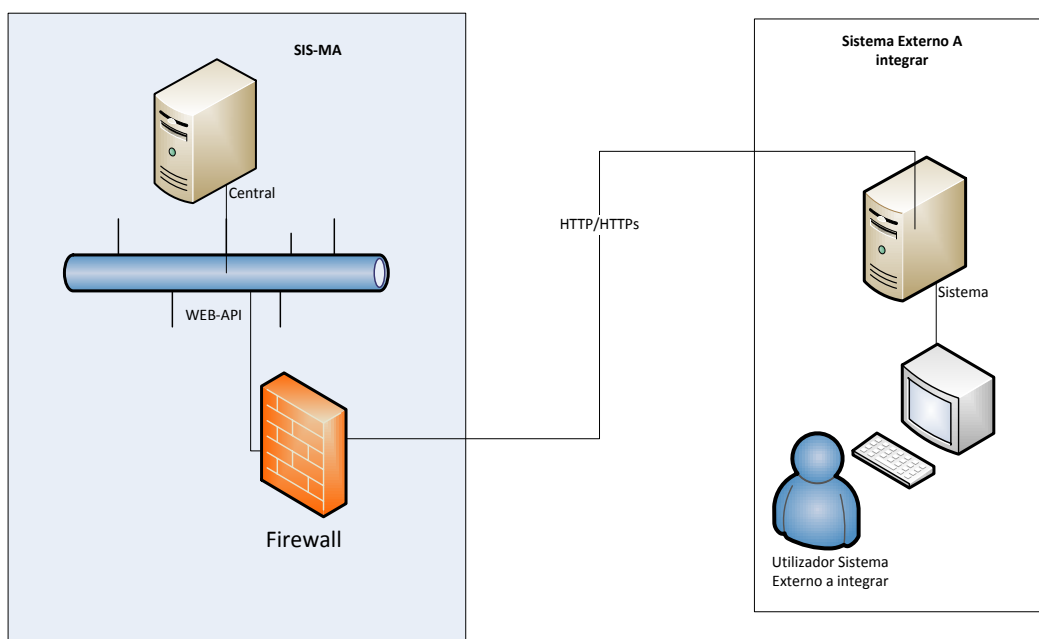


Figura 12: Integração Automática

Esta API Web está em conformidade com as regras de estilo de toda arquitectura. Isto implica que:

- A API Web fornece uma interface, de fácil navegação e legível por máquina, para o modelo de dados completo do DHIS, podendo desta forma ser facilmente efectuada a troca de informação entre sistemas.
- Os dados são acedidos através de uma interface uniforme (URL's) usando um protocolo bem conhecido (HTTP). Não são envolvidos formatos de transporte, apenas o protocolo HTTP bem testado, bem entendido, que actualmente é o protocolo principal de web. Isto significa que pode ser desenvolvido um sistema usando apenas o modelo de dados do SISMA sem conhecer a tecnologia específica do DHIS2 e suas restrições.
- Todos os dados, incluindo meta-dados, relatórios, mapas e gráficos, conhecidos como recursos, podem ser utilizados pela maioria dos formatos de representação popular da web tais como HTML, XML, JSON, PDF, PNG. Estes formatos são suportados em dispositivos e linguagens de programação oferecendo assim aos outros programadores uma grande variedade de opções de execução.

3.4.4 SISTEMAS PARA INTEGRAÇÃO

Durante a fase 1, fase de estudo, foram identificados os seguintes sistemas para considerar para integração:

Sistema	Descrição	Observação
SIS-H	Sistema de Informação para os Hospitais (Internamento)	Implementado em tecnologia Java/Oracle. Dados Agregados.
SISROH	Sistema de Informação de Registo de Óbitos Hospitalares	Implementado em Visual Basic/MySQL Dados individuais, no entanto produz relatórios com os dados agregados.
SIMAM	Sistema de Informação de Medicamentos e Artigos Médicos	É uma base de dados que guarda informação de stock. Usado para calcular o stock necessário. Usa base de dados MS-Acess, e produz relatório de dados agregados.
misau.gov.mz	Portal do Ministério da Saúde	
SISTAFE	Sistema de Administração Financeira do Estado.	Dados individuais.
SIFO	Sistema de Informação de Formação	Implementado em tecnologia VB.Net e usa base de dados SQL Server. Dados Agregados.
	Sistema de Gestão de Actividades de ONG's	Implementado em tecnologia Java Oracle Apex. Usa base de dados Oracle.
	Portal do Governo	
ETR.Net	PNCL/DNSP/Programa Nacional de Combate a Lepra	Usa base de dados MS-Acess. Recolha de dados individuais
E-SIP-SAUDE	Sistema Informação de Gestão dos Recursos Humanos	Implementado em tecnologia Visual Basic. Usa a base de dados MS-Acess. Dados Individuais.
SIFin	Sistema de Informação da Formação inicial	
SESP		

Tabela 5: Sistemas para integração

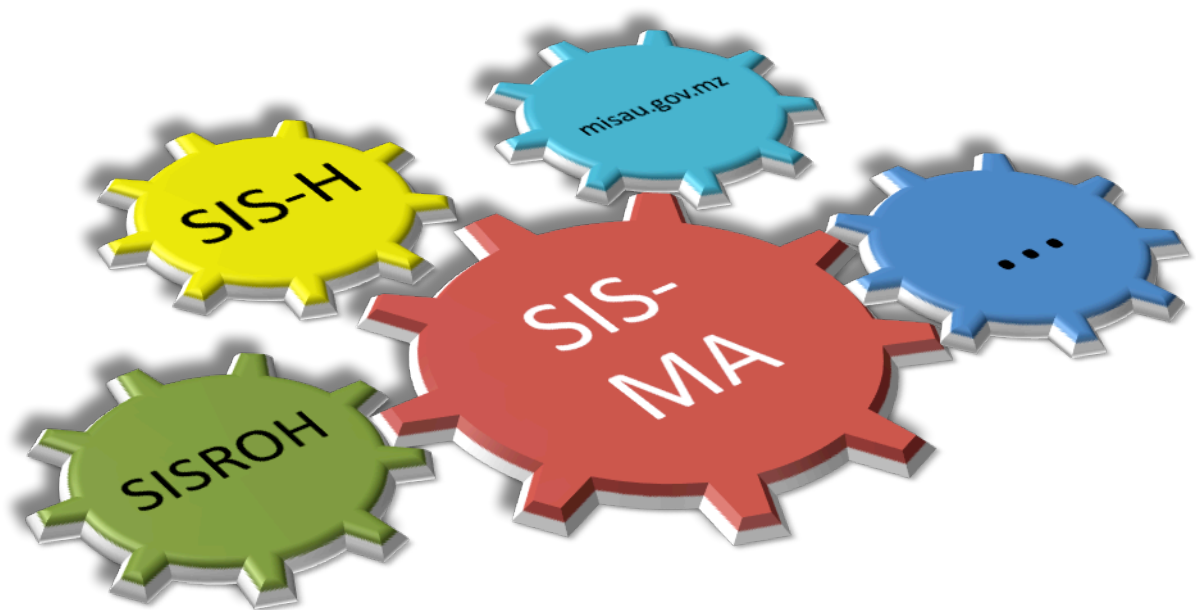


Figura 13: Integração de Sistemas

3.4.5 GESTÃO DE UNIDADES SANITÁRIAS

Para a configuração e consulta centralizada de Unidades Sanitárias e das diversas Unidades Organizacionais, o DHIS2 disponibiliza uma implementação FRED API (*The Facility Registry API*)⁴.

Esta implementação permite:

- Listar as Unidades Sanitárias/Unidades Organizacionais;
- Obter dados de uma Unidade Sanitária/Unidade Organizacional em específico;
- Criar uma Unidade Sanitária/Unidade Organizacional;
- Actualizar dados de uma Unidade Sanitária/Unidade Organizacional.

O acesso às operações é disponibilizado via WEB, através de interações HTTP, sendo necessário uma autenticação para a devida execução destas operações.

⁴https://facility-registry-api.readthedocs.org/en/latest/api_specifications.html

GET FACILITIES	GET /FACILITIES.JSON
GET A FACILITY	GET /FACILITIES/{ID}.JSON
CREATE A FACILITY	POST /FACILITIES.JSON
UPDATE A FACILITY	PUT /FACILITIES/{ID}.JSON
DELETE A FACILITY	DELETE /FACILITIES/{ID}.JSON

Figura 14: Operações *Facility Registry API*

Esta implementação permite uma gestão eficaz e única de Unidades Sanitárias, recorrendo a uma interacção simples e dinâmica. Permite assim um fácil acesso a entidades/sistemas externos a dados da Estrutura Organizacional aos diferentes níveis, incluindo detalhes de cada Unidade Organizacional. Possui também a possibilidade de definição das coordenadas das US e respectiva visualização no MAP.

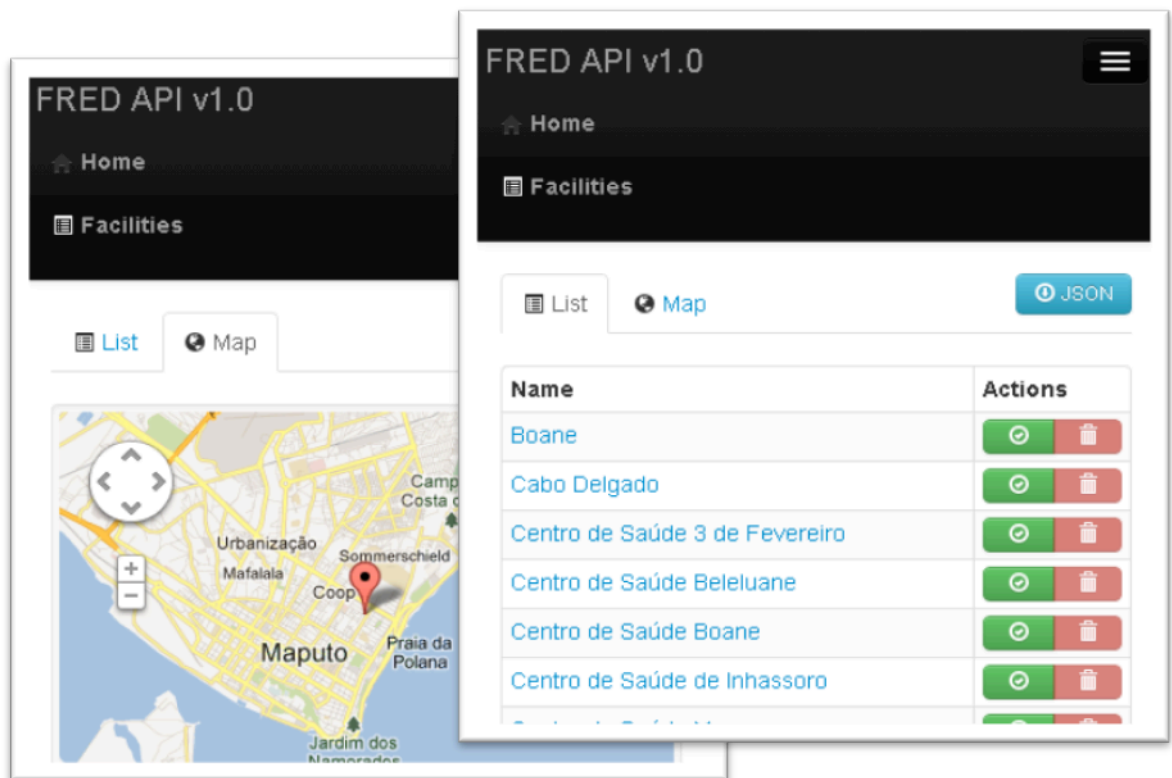


Figura 15: Facilities

3.5 MYDATAMART

MyDatamart é uma aplicação que se destina a utilizadores que pretendam realizar análise de dados de forma *offline*. A principal função do *MyDatamart* é de permitir conectar a uma instância *DHIS2 Online*, efectuar o *download* de meta-dados e dados e armazenar dados numa base de dados local.

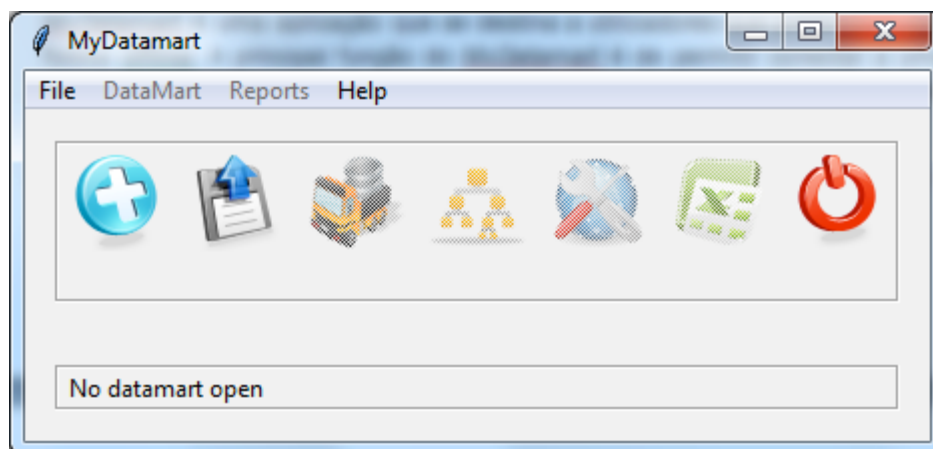


Figura 16-MyDatamart

Foi criado para operar em conjunto com Microsoft Excel, que pode ser usado para criar tabelas dinâmicas, ligando a uma base de dados local que pode ser usada como fonte de dados. Depois do *download* inicial de dados é possível serem efectuados downloads incrementais, por exemplo, baixar os dados mais recentes de cada mês. O *MyDatamart* também ajuda na produção de vistas SQL convenientes para uma análise mais poderosa e com o processo de conexão de tabelas dinâmicas para a base de dados local. Após a instalação dos drivers de base de dados necessários e do ficheiro executável do *MyDatamart* o workflow pode ser abaixo descrito:

1. Criar um novo ficheiro datamart.
2. Efectuar login no sistema central SISMA.
3. Fazer *download* dos meta-dados (elementos de dados, as Unidades de Saúde, etc)
4. Fazer *download* de dados (Registros).
5. Criar uma nova *Pivot Table* (tabela dinâmica) do *MyDatamart* ou link para um já existentes e actualizar a fonte de dados a partir do Microsoft Excel.

Esta aplicação é bastante útil em situações onde existem necessidades de produzir relatórios de diversos tipos sobre um determinado indicador numa determinada província ou distrito que não possua conectividade ao sistema central.

3.6 AUTENTICAÇÃO E AUTORIZAÇÃO

A autenticação e autorização serão feitas de forma integrada, utilizando o paradigma de utilizador *password*.

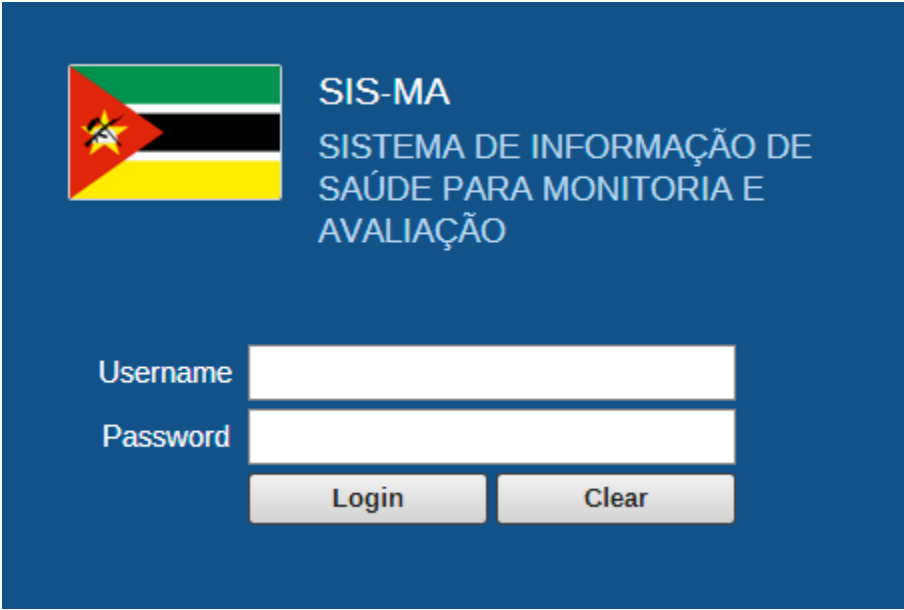


Figura 17: Ecrã de Autenticação

A utilização de um mecanismo de utilizador/*password* vai permitir controlar os acessos à aplicação e restringir o conteúdo/funcionalidades disponíveis na aplicação.

Os pedidos de novas contas e gestão das contas existentes será feito através da aplicação SIS-MA, sendo a gestão feita de forma centralizada.

3.7 LOGS E AUDITORIA

Os registos de *logs* de auditoria fornecem informações úteis sobre as medidas tomadas pelos utilizadores que se ligam e efectuam operações no SIS-MA. Por exemplo, é possível rastrear as alterações efectuadas por um utilizador a nível dos dados, o acesso a aplicação SIS-MA, criação e preenchimento de fichas/formulários e geração de relatórios. O registo de *logs* de auditoria pode dar mais detalhes de informação as investigações, ou acções tomadas pelos utilizadores sobre os problemas de qualidade de dados gerados.

O registo de *logs* de auditoria detalhada deve fornecer informação sobre todas as alterações efectuadas no SIS-MA sobre seguintes pontos:

- Acesso ao SIS-MA – Utilizador que acedeu a aplicação e quando.
- Alteração de dados na Base de Dados – Que alterações efectuou na base de dados.
- Geração de relatórios – Que relatórios produziu.
- Submissão de fichas – Que fichas/formulários criou/submeteu e quando.
- Importação e exportação de dados – Quem realizaram a operação de importação/exportação de dados.

O histórico de auditoria deve incluir a seguinte informação:

- Acção – Que acções o utilizador efectuou no SIS-MA
- Conteúdo do processo – Que conteúdo acedeu e de que forma.
- O utilizador – Quem efectuou uma determinada acção.
- A data e hora – A data e hora da realização da acção.

Deve produzir também informações sobre a ocorrência de problemas na aplicação tais como:

- Que tipo de erro e qual sua origem.
- Qual a causa do erro.

Os logs serão implementados recorrendo à tecnologia Commons Logging⁴ com Log4J⁵.

```
# Configuration file for log4j

# Log to file setup

log4j.appender.file = org.apache.log4j.RollingFileAppender
log4j.appender.file.File = dhis.log
log4j.appender.file.MaxFileSize = 100KB
log4j.appender.file.MaxBackupIndex = 3
log4j.appender.file.layout = org.apache.log4j.PatternLayout
log4j.appender.file.layout.ConversionPattern = * %-5p %d{ISO8601} %m (%F [%t])%n

# Log to console setup

log4j.appender.console = org.apache.log4j.ConsoleAppender
log4j.appender.console.layout = org.apache.log4j.PatternLayout
log4j.appender.console.layout.ConversionPattern = * %-5p %d{ISO8601} %m (%F [%t])%n

# Categories (order: DEBUG, INFO, WARN, ERROR, FATAL)

# Default logging level and appender

log4j.rootCategory = WARN, console
```

⁴ <http://commons.apache.org/proper/commons-logging/>

⁵ <http://logging.apache.org/log4j/1.2/>

```
# DHIS 2 logging level  
  
log4j.logger.org.hisp.dhis = INFO  
  
# Ehcache logging level  
  
log4j.logger.org.hibernate.cache=ERROR  
  
# Spring framework logging level  
  
log4j.logger.org.springframework = WARN  
  
log4j.logger.org.springframework.web = WARN
```

O exemplo anterior apresenta um ficheiro de configuração, com algumas das capacidades que a API de log nos oferece.

4 Estratégias de instalação

O SIS-MA é um sistema que pode ser acedido através da Internet/*Web* ou como sistema instalado localmente. As alternativas de instalação são definidas como:

- *Online*
- *Offline*

4.1 INSTALAÇÃO ONLINE

Uma instalação *Online* implica uma única instância do sistema configurada num servidor conectado à internet. Os utilizadores estabelecem ligação ao servidor central através da Internet usando um *browser*, à semelhança do que acontece com sistemas como *Gmail* ou *Facebook*.

4.2 INSTALAÇÃO OFFLINE

Uma instalação *Offline* implica que várias instâncias autónomas *Offline* sejam instaladas para os utilizadores finais, geralmente a nível distrital. Este tipo de instalação segue o modelo de operação em vigor no Módulo Básico.

O sistema é mantido pelos utilizadores finais/agentes de saúde do distrito, que insere os dados e gera relatórios. O sistema também tipicamente é mantido a nível nacional por uma equipa de *super-utilizadores* que fazem visitas regulares aos Distritos. Os dados são movidos para níveis hierárquicos superiores pelos utilizadores finais que produz ficheiros de transmissão de dados para serem enviados electronicamente por e-mail ou fisicamente pelo correio ou viagens de pessoal.

5 Desenho da base de dados

O modelo de dados é flexível em todas as dimensões de modo a permitir a captura de qualquer variável. Esta metodologia responde à necessidade de evolução do sistema, com a integração de diferentes Programas de Saúde em diferentes fases.

O Modelo baseia-se no paradigma de *DataValue*, Valor, que pode ser capturado para um *DataElement*, Variável, que representa o dado capturado num determinado período para uma determinada Unidade Organizacional.

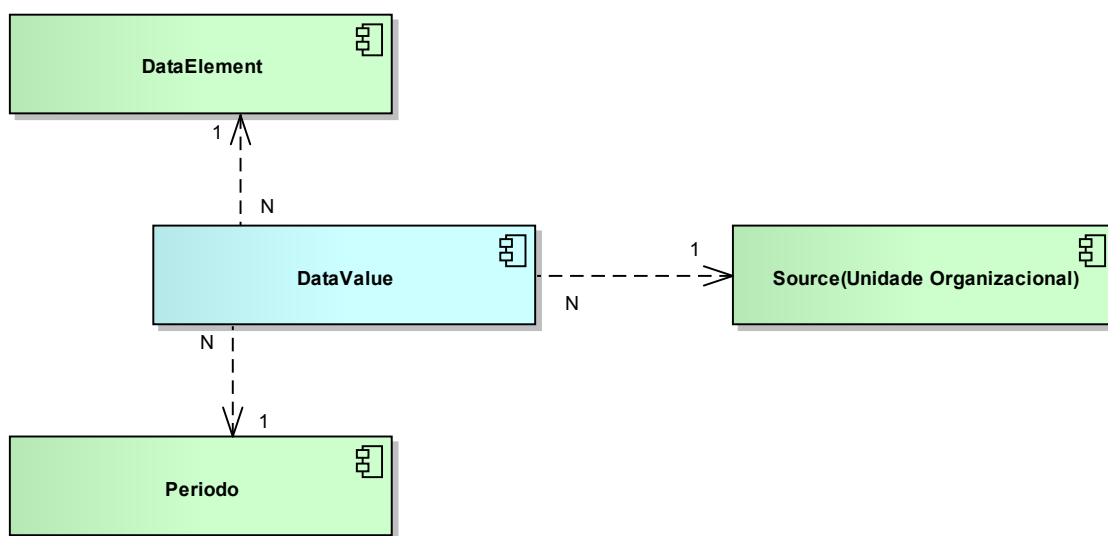


Figura 18: Estrutura de valor de dados

5.1 NÚCLEO DO SISTEMA

O conceito central para a captura de dados é o *DataSet*. O *DataSet* é uma colecção de *DataElements* onde os dados são representados em forma de lista, *grid* ou formulário personalizado. O *DataSet* é associado a um período, *PeriodType*, que representa a frequência de captura de dados.

O conceito chave e central para a análise de dados e elaboração de relatórios é o indicador. Um indicador é uma fórmula matemática composto por *dataElements* e números e, está associado a um tipo de indicador *indicatorType* que é representado em forma de percentagem.

A maioria dos objectos tem correspondência em grupos (*indicatorGroup*, *DataElementGroup*) de objectos de modo a melhorar e aperfeiçoar a análise de dados.

Os dados do Source code, podem ser encontrados no módulo API do projecto, e o detalhe do código fonte pode ser aí encontrado. O diagrama abaixo representa a colecção de alguns objectos de maior relevo.

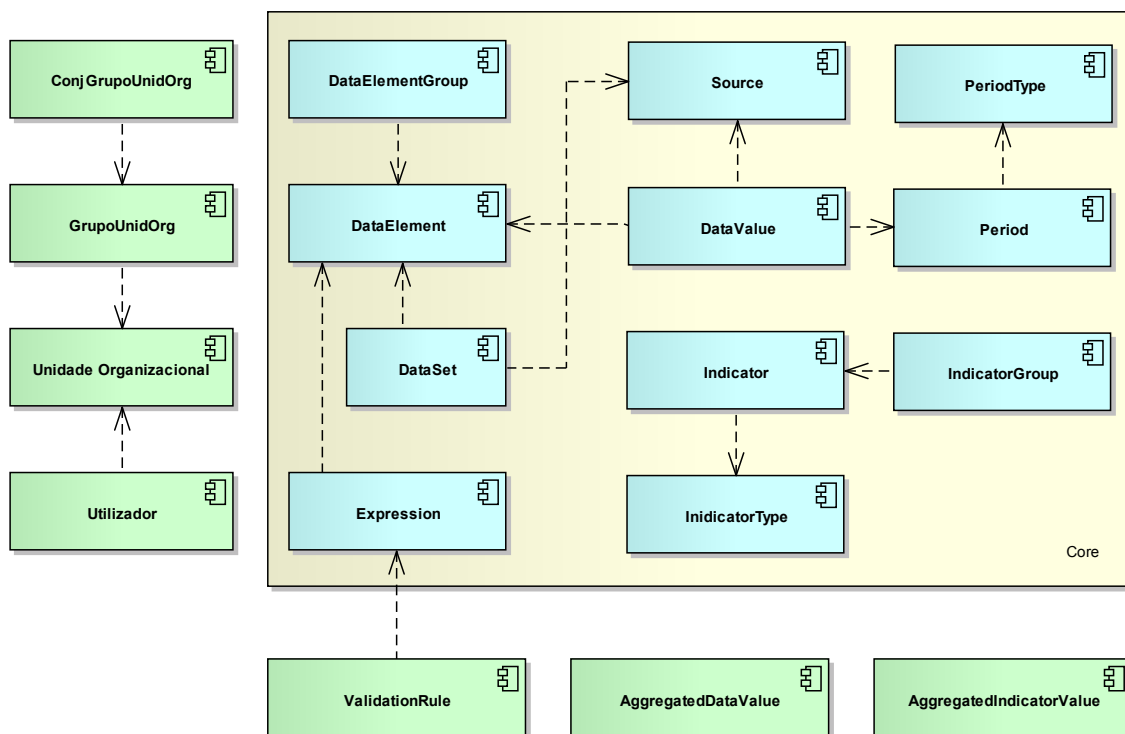
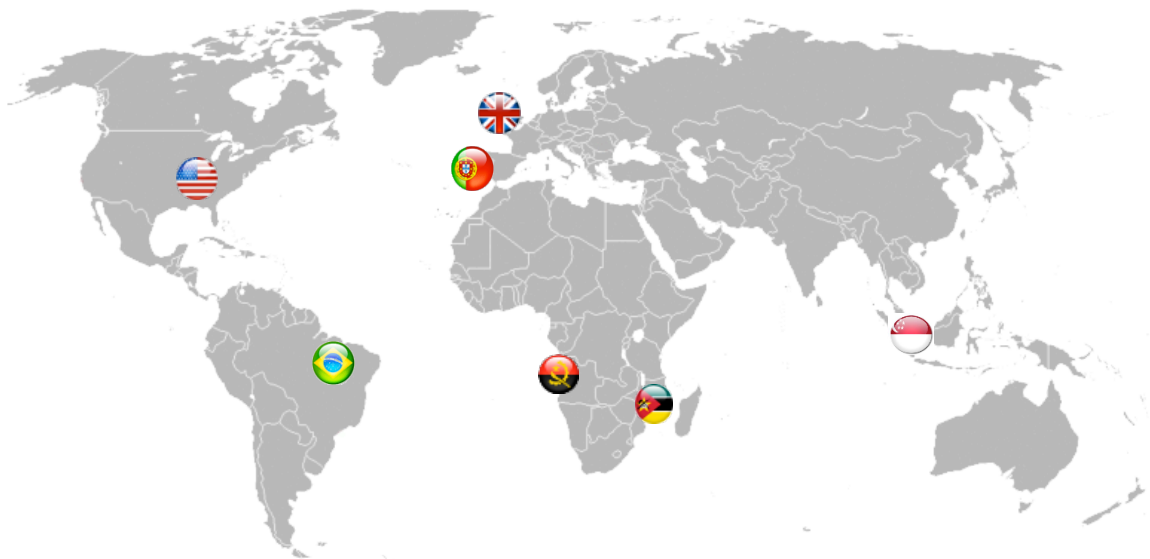


Figura 19: Diagrama de núcleo do sistema

6 Questões em aberto

As questões pendentes estão a ser geridas usando relatórios com pontos de situação das acções em aberto. Para cada uma das acções é identificado o responsável, e a data limite esperada de fecho da acção.



COIMBRA HEADQUARTERS, PORTUGAL

Parque Industrial de Taveiro, Lote 49
3045-504 Coimbra, Portugal
Tel.: +351 239 989 100
Fax: +351 239 989 119

LISBON OFFICE, PORTUGAL

Campus do Lumiar, Edifício M8
Estrada do Paço do Lumiar
1649-038 Lisboa, Portugal
Tel.: +351 217 145 430
Fax: +351 217 145 432

OPORTO OFFICE, PORTUGAL

Rua Eng.º Frederico Ulrich, nº 2650
4470-605 Moreira da Maia, Portugal
Tel.: +351 229 446 927/8
Fax: +351 229 446 929

SÃO PAULO, BRAZIL

Parque Tecnológico UNIVAP
Av. Shishima Hifumi 2911 - Urbanova
S. José dos Campos, S. Paulo
SP - CEP: 12244-000, Brazil
Tel: +55 12 3949 2512

SINGAPORE OFFICE, SINGAPORE

3 Temasek Avenue, #31-02 Centennial Tower,
Singapore, 039190
Tel: +65 68 3677 98
Fax: +65 68 3677 96

CHICAGO OFFICE, USA

70 West Madison St, Ste 5750
Chicago
Illinois, USA, 60602
Tel.: +1 (312) 772-6232

SOUTHAMPTON OFFICE, UK

2 Venture Road
Southampton Science Park
Southampton SO16 7NP, UK
Tel.: +44 (0) 23 8011 1339
Fax: +44 (0) 870 762 2487

YEOVIL OFFICE, UK

Yeovil Innovation Centre
Barracks Close, Copse Road
Yeovil, Somerset BA22 8RN, UK
Tel.: +44 (0)1935 385911
Fax: +44 (0)870 762 2487

MAPUTO OFFICE, MOZAMBIQUE

Rua Pereira Marinho, 179
Bairro da Sommerchild
Maputo, Mozambique
Tel.: +258 826 038 600

LUANDA OFFICE, ANGOLA

Rua Pedro Félix Machado, nº16, Mutamba
Luanda, Angola



CMMI is registered in the U.S. Patent and
Trademark Office by Carnegie Mellon University

