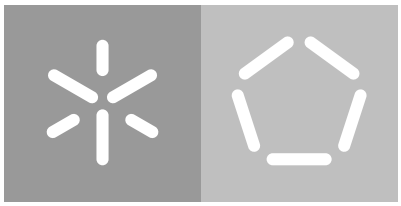


Universidade do Minho
Escola de Engenharia
Departamento de Informática

António Manuel de Melo Castanheira

Implementação de Sistema de Monitorização e Controlo de Interoperabilidade Clínica

April 2021



Universidade do Minho
Escola de Engenharia
Departamento de Informática

António Manuel de Melo Castanheira

Implementação de Sistema de Monitorização e Controlo de Interoperabilidade Clínica

Master dissertation
Master Degree in Informatics Engineering

Dissertation supervised by
Professor Doutor José Machado
Doutor Hugo Peixoto

April 2021

Despacho RT - 31 /2019 - Anexo 3

DIREITOS DE AUTOR E CONDIÇÕES DE UTILIZAÇÃO DO TRABALHO POR TERCEIROS

Este é um trabalho académico que pode ser utilizado por terceiros desde que respeitadas as regras e boas práticas internacionalmente aceites, no que concerne aos direitos de autor e direitos conexos.

Assim, o presente trabalho pode ser utilizado nos termos previstos na licença abaixo indicada.

Caso o utilizador necessite de permissão para poder fazer um uso do trabalho em condições não previstas no licenciamento indicado, deverá contactar o autor, através do RepositóriUM da Universidade do Minho.

Licença concedida aos utilizadores deste trabalho



Atribuição
CC BY

https://creativecommons.org/licenses/by/4.0/

AGRADECIMENTOS

Quero agradecer aos meus orientadores toda a atenção dispensada durante o processo de desenvolvimento desta dissertação, em especial ao Doutor Hugo Peixoto, pela disponibilidade e pela simplicidade com que lidou com todas as minhas questões.

Queria também agradecer aos meus pais por me apoiarem sempre durante o meu percurso e por me ter sustentado, pois só com os seus sacrifícios consegui sustentar-me e atingir os meus objetivos até hoje. Aos meus irmãos, por terem compreendido quando não pude estar tanto tempo com eles e por me apoiarem sempre. Por último, aos meus amigos por estarem sempre lá para mim e por me ajudarem sempre que lhes pedia opiniões sobre o que escrevi.

Despacho RT - 31 /2019 - Anexo 4

DECLARAÇÃO DE INTEGRIDADE

Declaro ter atuado com integridade na elaboração do presente trabalho académico e confirmo que não recorri à prática de plágio nem a qualquer forma de utilização indevida ou falsificação de informações ou resultados em nenhuma das etapas conducente à sua elaboração.

Mais declaro que conheço e que respeitei o Código de Conduta Ética da Universidade do Minho.

ABSTRACT

There has been a significant increase in the quantity of information that is stored digitally, as health infrastructures increasingly use digital systems to store information about the actors in their universe. It is therefore crucial that this information is governed by a set of rules in order to allow it to be understood without losing important data. With the increasing use of digital tools for storing and exchanging information, it is also important to monitor the information that is exchanged between systems to ensure the privacy and security of stakeholders, as well as ensuring the proper functioning of the system.

Therefore, the purpose of this work was to implement a tool to monitor messages exchanged on the *Agência de Interoperação, Difusão e Arquivo (AIDA)* platform, in order to actively contribute to solve the issues addressed in the *Regulamento Geral de Proteção de Dados (RGPD)*. This tool is based on technologies like Docker containers, the ElasticSearch database and Kibana, as a monitor interface.

Keywords: GDPR, ElasticSearch, Interoperability, Electronic Health Record

RESUMO

Atualmente tem-se verificado um aumento significativo da quantidade de informação armazenada digitalmente, devido a cada vez mais as infraestruturas de saúde recorrerem a sistemas digitais para armazenar a informação relativa aos intervenientes no seu universo. É, portanto, fulcral que essa informação seja regida por um conjunto de normas, de modo a permitir que seja compreendida sem se perderem dados importantes. Com o aumento do uso de ferramentas digitais para armazenamento e troca de informação também se torna importante a monitorização dos dados trocados entre os sistemas, de modo a garantir a privacidade e segurança dos intervenientes, bem como garantir o bom funcionamento do sistema.

Nesse sentido, o trabalho consistiu na implementação de uma ferramenta para a monitorização de mensagens trocadas na plataforma AIDA - Agência de Interoperação, Difusão e Arquivo - de maneira a contribuir de forma ativa para resolver as questões abordadas no RGPD. Esta ferramenta é baseada em tecnologias como *containers* Docker, a base de dados ElasticSearch e a interface de monitorização Kibana.

Keywords: RGPD, ElasticSearch, Interoperabilidade, Registo Médico Eletrónico

CONTEÚDO

1 INTRODUÇÃO 1
1.1 Contexto 1
1.2 Motivação 2
1.3 Contribuições e Objetivos 3
1.4 Estrutura do documento 4
2 ESTADO DA ARTE 5
2.1 Interoperabilidade na saúde 5
2.1.1 Plataforma AIDA 7
2.1.2 HL7 8
2.2 Sistemas de monitorização de Sistemas de Informação 9
2.2.1 Datadog 10
2.2.2 Loggly 13
2.2.3 Nagios 14
2.2.4 ElasticStack 16
2.2.5 Grafana 18
2.3 Sistemas de Monitorização na Saúde 19
2.3.1 HealthIRA 19
2.4 Regulamento Geral de Proteção de Dados 20
2.4.1 O Regulamento Geral de Proteção de Dados na saúde 21
2.4.2 GDPR-Compliant Personal Data Management: A Blockchain-Based Solution 22
3 METODOLOGIAS E TECNOLOGIAS 23
3.1 Metodologia de Investigação 23
3.2 Tecnologias de Desenvolvimento 25
3.2.1 MirthConnect 25
3.2.2 Tecnologias de virtualização 26
3.2.3 Bases de Dados Não Relacionais 28
3.2.4 Logstash 33
3.2.5 Ferramentas de Monitorização 34
3.3 Metodologia de Avaliação 38
3.3.1 Avaliação de Performance 38
3.3.2 Análise SWOT 39
4 PLATAFORMA DE MONITORIZAÇÃO 40

4.1 Identificação do Problema 40
4.2 Requisitos Funcionais 41
4.3 Requisitos Técnicos 41
4.4 Arquitetura da Solução 42
4.4.1 Ambiente de Desenvolvimento 43
4.4.2 Ambiente de Produção 48
5 RESULTADOS, DISCUSSÃO E AVALIAÇÃO 53
5.1 Avaliação de Performance 53
5.2 Análise SWOT 55
6 CONCLUSÃO E TRABALHO FUTURO 57

7 PUBLICAÇÃO 63
7.1 Overcoming Challenges in Healthcare Interoperability Regulatory Compliance 63

LISTA DE FIGURAS

Figura 1 Níveis de interoperabilidade 5
Figura 2 Esquema de troca de informação na plataforma AIDA 7
Figura 3 Estrutura de uma mensagem A01 8
Figura 4 Arquitetura da aplicação proposta por [13] para sistema de monitorização de informação clínica 9
Figura 5 Exemplo de um *dashboard* do Datadog, com métricas sobre a base de dados MySQL 10
Figura 6 Exemplo de um *dashboard* do Datadog sobre monitorização de processos 11
Figura 7 Exemplo de um *dashboard* do Datadog sobre *logs* 11
Figura 8 Exemplo de um *dashboard* do Datadog sobre *logs* 12
Figura 9 Exemplo de um *dashboard* do Loggly 13
Figura 10 Exemplo de um *dashboard* de visão centralizada do Nagios 14
Figura 11 *Dashboard* de alertas do Nagios 15
Figura 12 *Dashboard* de relatórios do Nagios 15
Figura 13 Fluxo de dados do ElasticStack 16
Figura 14 Arquitetura da ferramenta de monitorização da *cloud* OpenStack, como proposto em [20] 17
Figura 15 Arquitetura da ferramenta de monitorização, como em [21] 18
Figura 16 Arquitetura da HealthIRA, como em [22] 19
Figura 17 Arquitetura da plataforma de gestão de dados, como em [23] 22
Figura 18 Esquema da aplicação da metodologia *Design Science Research (DSR)*, como proposta em [24] 24
Figura 19 Arquitetura de um canal Mirth 25
Figura 20 Esquema de um sistema baseado em *containers* 27
Figura 21 Esquema de um sistema baseado em máquinas virtuais 28
Figura 22 Exemplo da estrutura uma base de dados orientada a documentos 29
Figura 23 Modo *single-server* 30
Figura 24 Modo distribuído 30
Figura 25 Arquitetura da base de dados Solr, como em [38] 32
Figura 26 Exemplo do funcionamento do Logstash 33
Figura 27 Página de pesquisa do Kibana 34
Figura 28 Página de visualizações do Kibana 35

Figura 29	Página dos *dashboards* do Kibana	35
Figura 30	Página dos *dashboards* do Grafana	36
Figura 31	Página das *queries* de uma visualização	37
Figura 32	Primeira proposta da arquitetura da solução	42
Figura 33	Arquitetura da solução	43
Figura 34	Arquitetura da primeira solução local	44
Figura 35	*Dashboard* da contagem e listagem de mensagens	45
Figura 36	*Dashboard* dos destinos das mensagens	45
Figura 37	*Dashboard* dos episódios	46
Figura 38	Arquitetura da segunda simulação local	47
Figura 39	Arquitetura da terceira simulação local	48
Figura 40	Arquitetura da primeira solução implementada no Centro Hospitalar do Tâmega e do Sousa	49
Figura 41	*Dashboard* da contagem e listagem de mensagens	50
Figura 42	*Dashboard* dos destinos das mensagens	50
Figura 43	*Dashboard* dos episódios	51
Figura 44	Arquitetura da segunda solução implementada no Centro Hospitalar do Tâmega e do Sousa	51
Figura 45	Arquitetura da solução final de monitorização	52
Figura 46	Comparação entre os parâmetros de avaliação propostos no capítulo 3 e os resultados obtidos	54
Figura 47	Análise SWOT	55

LISTA DE TABELAS

Tabela 1	ElasticSearch vs BD relacional	30
Tabela 2	Análise comparativa entre Grafana e Kibana	37

GLOSSÁRIO

A

AIDA Agência de Interoperação, Difusão e Arquivo.

D

DI Departamento de Informática.

DSR Design Science Research.

E

ELK ElasticSearch Logstash Kibana - ElasticStack.

EST European Summer Time.

F

FHIR Fast Health Interoperability Resource.

H

HL7 Health Level Seven.

I

IOT Internet of Things.

M

MEI Mestrado em Engenharia Informática.

N

NO SQL Not Only SQL.

P

P2P Peer-to-Peer.

PCE Processo Clínico Eletrónico.

R

RGPD Regulamento Geral de Proteção de Dados.

S

SAAS Software as a Service.

SWOT Strengths, Weaknesses, Opportunities & Threats.

U

UE União Europeia.

UM Universidade do Minho.

UTC Coordinated Universal Time.

1

INTRODUÇÃO

Esta dissertação descreve o desenvolvimento de uma ferramenta para monitorizar as operações de tratamento de dados realizadas no seio da plataforma AIDA, uma plataforma que assegura a interoperabilidade entre sistemas de informação e que opera em alguns hospitais portugueses, contribuindo para que a plataforma opere em conformidade com a legislação mais recente no âmbito da proteção de dados. O trabalho foi realizado no contexto do *Mestrado em Engenharia Informática (MEI)*, realizado no *Departamento de Informática (DI)* da *Universidade do Minho (UM)*.

O capítulo introdutório aborda temas importantes para a compreensão do presente documento:

- Contexto e motivações para o desenvolvimento do trabalho proposto;
- Questões que a ferramenta desenvolvida pretende resolver na plataforma;
- Objetivos para responder às questões propostas;
- Estrutura do documento

1.1 CONTEXTO

Os avanços tecnológicos da última década deram origem a uma mudança radical na administração de sistemas na área da saúde: A informação hospitalar que era, no passado, guardada em papel, passou a ser guardada em sistemas informáticos. Esta mudança deu origem ao conceito de *Processo Clínico Eletrónico (PCE)*, que pode ser definido como informação sobre a vida de um indivíduo armazenada digitalmente com o objetivo de apoiar a continuidade dos cuidados, educação e pesquisa e garantir a confidencialidade durante todo o processo [1].

Uma das plataformas que apareceram com esses avanços foi a AIDA, uma plataforma baseada em agentes inteligentes que assegura a interoperabilidade entre unidades de saúde [2], que opera atualmente em alguns hospitais portugueses (Centro Hospitalar do Porto, Centro Hospitalar do Alto Ave, Centro Hospitalar do Tâmega e Sousa e Unidade Local de Saúde do Alto Alentejo) [3].

Com o surgimento deste género de plataformas começaram a surgir questões éticas no âmbito da privacidade dos dados pessoais. Questões sobre o acesso, processamento, tratamento e armazenamento de dados pessoais têm ganho cada vez mais importância na sociedade, levando à criação de legislação sobre esses assuntos, o RGPD [4], em vigor a nível europeu. Tal legislação trouxe um novo paradigma na área da interoperabilidade na saúde, tendo agora estas plataformas, como a AIDA, a necessidade de criar ferramentas para armazenarem e monitorizarem as operações de tratamento de dados realizadas no seu seio.

Nesse sentido, o trabalho aqui apresentado consiste na implementação de uma ferramenta para a monitorização da informação trocada em unidades de saúde, de maneira a contribuir de forma ativa para resolver as questões abordadas no RGPD, mais concretamente no artigo 30º, que define que deve existir um registo de todas as atividades de tratamento de dados.

A ferramenta desenvolvida utiliza ElasticSearch [1], uma Base de Dados orientada a documentos otimizada para pesquisa de texto, Logstash [2] e Kibana [3], plugins de ElasticSearch para recolha e processamento de dados - Logstash - e para visualização dos dados armazenados - Kibana. É também utilizado Docker [4], uma ferramenta de virtualização que permite o encapsulamento de aplicações e ferramentas em *containers* e, durante o desenvolvimento da ferramenta, foi também utilizado MirthConnect [5], uma ferramenta de integração para processamento de mensagens de saúde.

1.2 MOTIVAÇÃO

A escolha deste tema de dissertação deveu-se a uma grande curiosidade pessoal pela realização de mais um projeto na área da monitorização aplicacional e também na área das aplicações informáticas na saúde, tendo já realizado projetos em ambas as áreas. Sendo as ferramentas propostas para este projeto tecnologias em crescimento em termos de utilização atualmente, tornou-se também relevante a ambientação a essas tecnologias.

Além da motivação relacionada com a área do projeto, acresce também a motivação de trabalhar uma vez mais com os orientadores desta dissertação, considerando que os projetos realizados em cadeiras lecionadas por ambos, na área da administração de bases de dados, da mineração de dados e da interoperabilidade sido satisfatórios e despertado interesse nas respetivas áreas.

Por último, constituiu também motivação o facto deste projeto ser realizado num grupo de investigação, sendo por essa razão desenvolvido num ambiente mais profissional.

1 https://www.elastic.co/products/elasticsearch
2 https://www.elastic.co/products/logstash
3 https://www.elastic.co/products/kibana
4 https://www.docker.com/
5 https://www.nextgen.com/products-and-services/integration-engine

1.3 CONTRIBUIÇÕES E OBJETIVOS

A ferramenta proposta pretende responder a algumas questões pertinentes que inevitavelmente foram colocadas aquando do aparecimento de legislação ao nível da proteção de dados. Tal legislação trouxe para as plataformas na área da saúde acrescidas preocupações com a forma como a sua interação com a informação clínica relativa aos pacientes que usufruem de cuidados de saúde é realizada. Tornou-se imperativo que estas possuam componentes para o registo e monitorização das operações de tratamento de dados pessoais, de modo a que possam atuar em conformidade com a mesma. Por essa razão, a criação de ferramentas como a exposta neste documento é, então, cada vez mais uma prioridade das empresas e grupos de investigação que gerem essas plataformas.

Estando este projeto inserido no grupo de investigação Algoritmi, da Universidade do Minho, responsável pelo funcionamento da plataforma AIDA, é também uma prioridade deste grupo a criação de mecanismos para monitorizar o funcionamento e a *performance* dos componentes da plataforma, beneficiando também da sua capacidade para armazenar métricas relacionadas com o seu funcionamento, para que, para além de atuar em conformidade com o RGPD, seja possível também minimizar e detetar em menos tempo possíveis erros e avarias.

Tendo o exposto anteriormente em conta, é importante responder a diversas questões ao longo desta dissertação. A criação de ferramentas para a eficaz monitorização da troca de informação entre os sistemas de informação da plataforma AIDA levou à necessidade de garantir que a monitorização dessa troca é eficaz e que a ferramenta permita que a plataforma atua em conformidade com o RGPD.

Outra questão pertinente prende-se com outro aspeto da plataforma AIDA: sendo uma prioridade do grupo de investigação a criação de mecanismos de monitorização das aplicações médicas constituintes do ambiente em que atua a plataforma, é necessário perceber de que forma é possível a integração de tais mecanismos sem descurar a principal funcionalidade da ferramenta - a monitorização da troca de informação.

A resolução dessas questões é, portanto essencial, tendo sido imperativa a criação de objetivos concretos para lhes responder de forma objetiva, divididos por 3 fases, por forma a serem mais facilmente cumpridos:

- Primeira fase - Realização de um estudo sobre as tecnologias a utilizar no desenvolvimento da ferramenta de monitorização, sobre a legislação de proteção de dados e sobre conceitos importantes para o projeto. Deste modo é possível perceber se é efetivamente possível, com a utilização da ferramenta de monitorização, garantir que a plataforma AIDA comece a atuar em conformidade com o RGPD.
- Segunda fase
  - Definição dos requisitos funcionais e técnicos que a solução deve cumprir.

  - Realização de testes às tecnologias e à arquitetura definida, através da construção de uma simulação local do ambiente em que a ferramenta viria a atuar, para depreender se é possível garantir a efetiva monitorização da troca de dados no seio da plataforma.

- Terceira fase - Implementação da solução em ambiente real e a adição dos mecanismos de monitorização aplicacional, verificando desta forma se é realmente possível integrá-los sem descurar a monitorização da troca de informação.

1.4 ESTRUTURA DO DOCUMENTO

O presente documento encontra-se dividido em 6 capítulos:

- Neste capítulo, capítulo 1, é feita uma introdução ao tema, abordando assuntos importantes para uma melhor compreensão do projeto, como o contexto e as motivações deste, as questões que a ferramenta pretende resolver e os objetivos para lhes responder;

- No segundo capítulo são expostos os conceitos relevantes para o desenvolvimento - Interoperabilidade na saúde, a plataforma AIDA, projetos na área e a legislação de proteção de dados;

- No terceiro capítulo são apresentadas as metodologias de investigação e avaliação aplicadas no desenvolvimento do projeto e as tecnologias utilizadas;

- No quarto capítulo é apresentada a solução desenvolvida e os diversos aspetos relevantes durante o seu desenvolvimento: Requisitos funcionais e técnicos, as arquiteturas utilizadas na simulação local e na implementação e as características e funcionalidades da ferramenta;

- No quinto capítulo são apresentados os resultados inerentes da implementação da ferramenta em ambiente real, feita a sua discussão e avaliação;

- No último capítulo é feita a conclusão da dissertação e a apresentação do trabalho futuro.

2

ESTADO DA ARTE

Este capítulo do presente documento descreve o estudo feito sobre um conjunto de assuntos relevantes para o desenvolvimento da ferramenta e as análises realizadas a aplicações e serviços que atuem na mesma área da solução proposta.

2.1 INTEROPERABILIDADE NA SAÚDE

Não existe uma definição concreta de Interoperabilidade. Segundo [5], este conceito consiste na capacidade de duas entidades, humanas ou máquinas, acederem e usarem dados de várias fontes e sistemas, confiável e rapidamente, de modo a funcionarem sem a ocorrência de erros. Outra definição de interoperabilidade, segundo [6], é a a capacidade de dois ou mais sistemas trocarem informação e de a utilizarem.

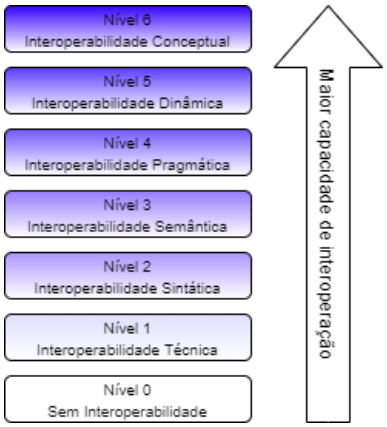


Figura 1: Níveis de interoperabilidade

Como exposto na Figura1, existem diversos níveis de interoperabilidade, diferenciando-se pela capacidade que as entidades têm de comunicar entre si [7]:

- Nível 0: Não existe comunicação entre os sistemas.
- Nível 1: Existe um protocolo de comunicação entre os sistemas.
- Nível 2: A informação trocada entre os sistemas possui formato e estrutura comuns.

- Nível 3: Para além da informação trocada entre os sistemas ter um formato comum, o significado desta também é interpretado da mesma forma.
- Nível 4: Todos os intervenientes conhecem o contexto em que a informação é trocada.
- Nível 5: Os sistemas têm a capacidade de tirar vantagens sobre mudanças de estado, para além de compreender os efeitos causados pela troca de informação.
- Nível 6: Os intervenientes estão em conformidade com as premissas e restrições de cada ambiente real.

Este conceito tem um papel cada vez mais importante quando se trata de desenvolver e manter *software*, especialmente na área da saúde, na partilha de dados entre múltiplas aplicações e serviços trazendo diversos benefícios quando aplicado. Atualmente os cuidados de saúde são prestados a um mesmo paciente por diversas unidades de saúde, tais como dentistas, laboratórios de análises, hospitais ou clínicas. Sendo assim, a informação clínica e pessoal sobre um paciente está replicada nas unidades de saúde que este frequenta. A comunicação entre os diversos sistemas das entidades é, então, essencial, para que haja acesso a essa informação pelas diversas entidades, garantindo melhores cuidados de saúde, pois, com acesso a mais informação os prestadores de saúde podem tomar decisões mais acertadas e cometer menos erros.

Os sistemas hospitalares funcionam, cada vez mais, com múltiplos componentes, muitas vezes construídos em *hardware* e sistemas operativos diferentes e utilizando diferentes linguagens, podendo estes fatores acarretar custos financeiros e burocráticos incomportáveis [8] e acabando por dificultar a comunicação entre os mesmos. A aplicação do conceito de interoperabilidade viria reduzir os custos que a utilização desses componentes sem comunicação eficaz entre si acarretava.

Com o aumento dos custos inerentes aos sistemas de saúde e das deficiências destes, é essencial que sejam desenvolvidos projetos que resolvam as diversas deficiências detetadas nas plataformas de saúde, tais como problemas ao nível da *standardização* de formatos e arquiteturas utilizadas nestas ou a utilização de ontologias ou terminologias clínicas incompatíveis entre elas [9].

De modo a facilitar a comunicação entre os componentes sem perda de informação foram criados standards, tais como o *Health Level Seven (HL7)* [10], que especifica a sintaxe da informação trocada pelos componentes, definindo também a forma como a informação deve ser trocada.

2.1.1 *Plataforma AIDA*

A AIDA é uma plataforma baseada em agentes inteligentes que assegura a interoperabilidade entre unidades de saúde, utilizando a norma HL7 v2, e que opera atualmente no Centro Hospitalar do Porto, no Centro Hospitalar do Alto Ave, no Centro Hospitalar do Tâmega e Sousa e na Unidade Local de Saúde do Alto Alentejo, unidades de saúde públicas portuguesas [3]. É um módulo transversal a cada uma destas unidades de saúde que permite diminuir a integração ponto a ponto entre os diversos componentes das mesmas e estabelecer normas de comunicação mais fiáveis e seguras que facilitem a integração ou remoção de componentes no sistema sem alteração da estrutura global. Esta plataforma estabelece canais de comunicação com os diversos componentes existentes nas unidades de saúde, envia, recebe, gere e armazena a informação e responde a pedidos com os recursos necessários de modo a poder responder no tempo correto.

Foi desenhada para dar auxílio a aplicações médicas, bem como para controlar o fluxo de dados trocados numa rede de sistemas inteligentes de processamento de informação, com um nível ajustável de autonomia [2], possuindo diversas interfaces baseadas em webservices que permitem executar pesquisas e fazer a gestão do *data warehouse*, bem como visualizar a informação de forma facilitada. O seu objetivo principal é permitir a difusão e integração da informação gerada em contexto hospitalar [11].



Figura 2: Esquema de troca de informação na plataforma AIDA

Como está ilustrado na Figura2, a plataforma AIDA opera como um servidor de comunicação entre as diversas aplicações médicas das unidades de saúde em que esta está instalada, recolhendo a informação gerada nos sistemas de informação e armazenando-a na base de dados.

2.1.2 *HL7*

O HL7 é um conjunto de normas e padrões para a troca, partilha, integração e recuperação de informação eletrónica de saúde [10]. A aplicação deste conjunto de normas em sistemas de prestação de serviços médicos resolve um dos problemas principais existentes nos sistemas de saúde - a utilização de formatos diferentes para a informação clínica - pois esta passaria a ser estruturada da mesma forma nos sistemas em que este conjunto de normas estaria aplicado.

Um dos padrões mais utilizados é o HL7 v2, que define a estrutura da informação clínica trocada entre sistemas. Foi desenhado para ser utilizado em sistemas de cuidados de saúde, bem como em ambientes distribuídos, nos quais os dados estão espalhados por departamentos [12], como é o caso da plataforma AIDA.

Este padrão possui mensagens-tipo para as mais diversas operações realizadas em ambiente hospitalar, tais como entrada ou saídas de pacientes ou mensagens relativas a visitas a pacientes. Essas mensagens são constituídas por vários segmentos.

O exemplo seguinte (Figura 3) ilustra uma mensagem do tipo A01, que é enviada quando um paciente é admitido no hospital. Contém a informação principal do paciente e é comunicada a todos os componentes do sistema hospitalar, de modo a evitar a existência de mensagens repetidas.

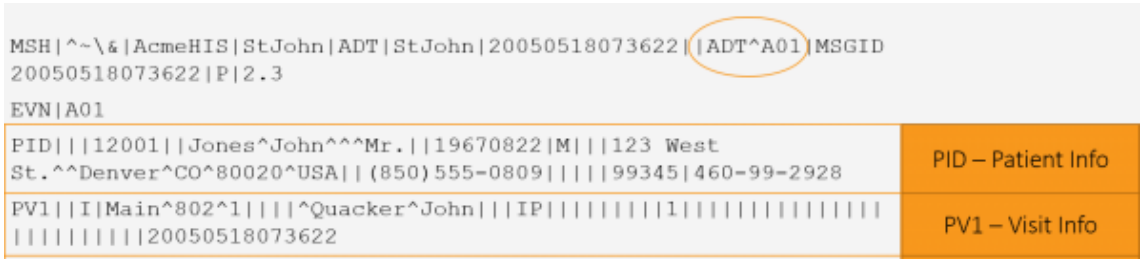


Figura 3: Estrutura de uma mensagem A01

O primeiro segmento da mensagem contém informação sobre as instalações que enviaram e receberam a mensagem, a data da mensagem, o tipo da mensagem (no caso, ADTÂ01), o ID desta e a versão do HL7.

O segundo segmento contém a informação relativa ao paciente que é admitido: o nome, a data de nascimento, o sexo, a morada completa e o telefone.

O terceiro e último segmento contém a informação relativa ao local onde o paciente vai ser colocado: O estado do registo (se é uma emergência, se está admitido, a sair ou em pré admissão), a cama que lhe foi atribuída, o nome do médico, o serviço hospitalar e a data em que foi admitido.

Em [13] é exposta a criação de uma aplicação pessoal de gestão de informação clínica que permita a interoperabilidade com outras aplicações do género, bem como com sistemas hospitalares. Esta aplicação recorre ao conjunto de standards HL7 *Fast Health Interoperability Resource (FHIR)* para definir a estrutura dos dados trocados com os outros sistemas, pois este conjunto de standards permite a gestão de atributos como os sinais vitais, medicação ou alergias, atributos importantes quando se tratam de aplicações como esta.

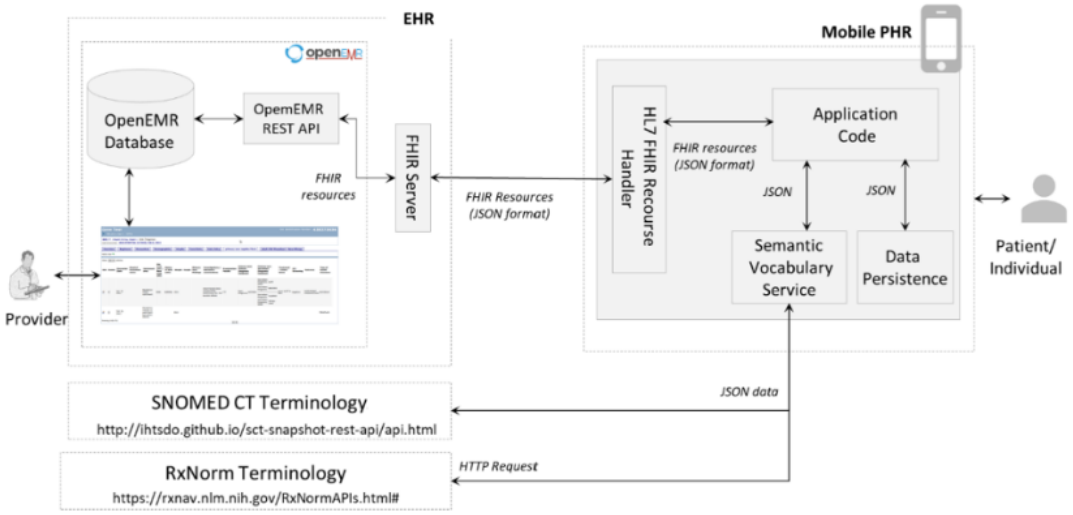


Figura 4: Arquitetura da aplicação proposta por [13] para sistema de monitorização de informação clínica

A informação é trocada entre os 2 componentes utilizando o formato JSON. A arquitetura da aplicação, como em [13], está ilustrada na Figura4. Esta aplicação está dividida em 2 componentes: Um sistema que guarda registos médicos, que assenta num software de gestão e arquivo de documentos médicos, o OpenEMR [1] e numa biblioteca Java, HAPI FHIR [2] que define classes para os tipos de recursos e atributos do conjunto de standards HL7 e disponibiliza um servidor e um cliente para aplicar essas normas, e a aplicação móvel.

2.2 Sistemas de monitorização de sistemas de informação

Nos dias de hoje, o mundo depende cada vez mais de serviços eletrónicos para o seu funcionamento. Serviços das mais variadas áreas necessitam de estar operacionais a todo o instante, com o mínimo possível de falhas. Para esse efeito, necessitam de ferramentas que monitorizem o seu funcionamento, para detetar e solucionar mais rapidamente possíveis problemas, de forma a minimizar os custos relacionados com o tempo em que estiverem fora de ação. É nesse sentido que aparecem ferramentas como as apresentadas nesta secção.

1 https://www.open-emr.org/
2 http://hapifhir.io/

2.2.1 *Datadog*

O Datadog [3] é um serviço de monitorização que trabalha como *Software as a Service (SaaS)* - É disponibilizado na cloud e os utilizadores têm acesso utilizando a internet [14]. Possui mais de 400 integrações com diversos sistemas, aplicações ou serviços, tais como a Amazon, ElasticSearch, GitHub ou Google, o que lhe dá uma grande versatilidade no que toca às soluções que permite monitorizar. Utiliza ElasticSearch para o armazenamento das métricas e eventos a monitorizar, o que lhe permite obter uma boa *performance* na monitorização. Está disponível apenas com planos pagos.

Esta ferramenta tem diversas funcionalidades: permite a criação de *dashboards* para monitorizar as diversas plataformas com as quais possui integrações, permite a emissão de alertas para email ou através de SMS quando uma determinada métrica ultrapassa limites definidos ou quando acontece um determinado evento indesejado e possui uma API REST para aceder à plataforma, possibilitando a criação de novas integrações.

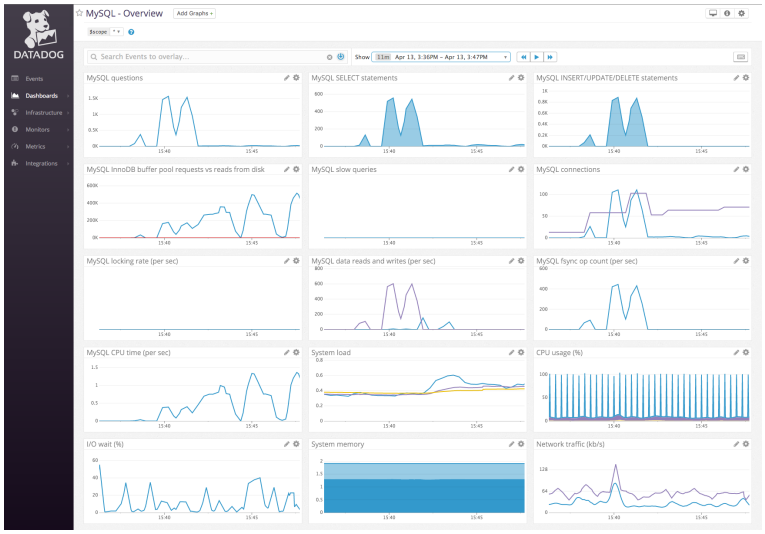


Figura 5: Exemplo de um *dashboard* do Datadog, com métricas sobre a base de dados MySQL

Na Figura5 está ilustrado um *dashboard* da integração do Datadog com a base de dados MySQL, que permite a visualização de diversas métricas relativas ao funcionamento da base de dados, como a *performance* das *queries* realizadas ou o número de conexões ativas, sendo possível a criação de novas métricas, através de *queries* SQL.

3 https://www.datadoghq.com/

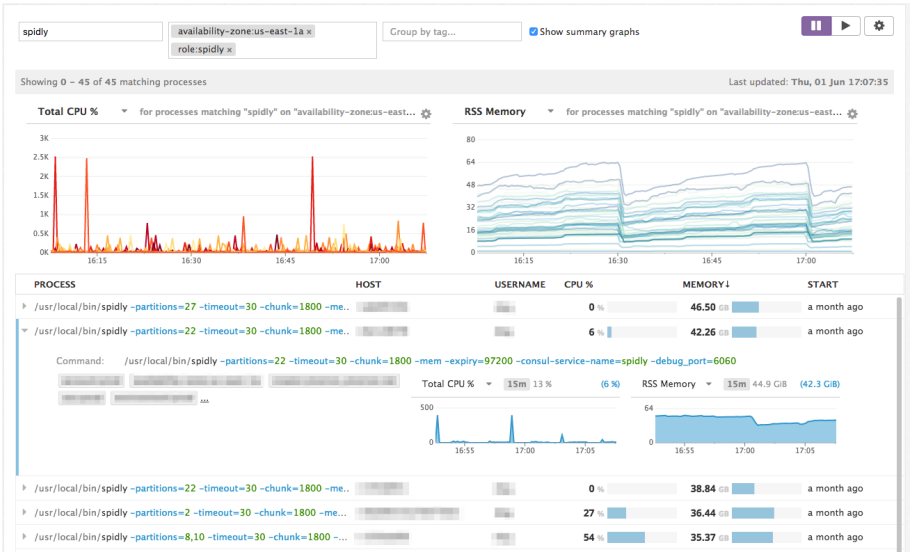


Figura 6: Exemplo de um *dashboard* do Datadog sobre monitorização de processos

Na Figura6 está ilustrado um *dashboard* relativo aos processos a correr na infraestrutura monitorizada. É possível monitorizar métricas como a quantidade de CPU e memória que cada processo gasta, bem como observar a listagem dos processos.

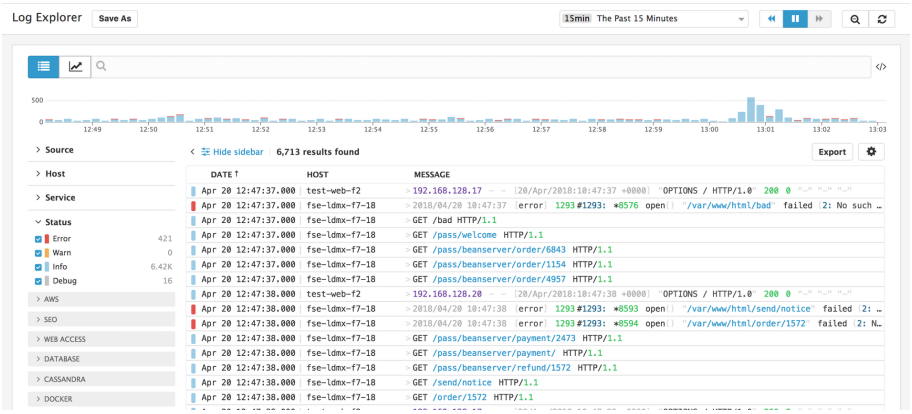


Figura 7: Exemplo de um *dashboard* do Datadog sobre *logs*

No *dashboard* da Figura7 é possível pesquisar e lançar alertas relacionados com os *logs* das tecnologias monitorizadas. O Datadog permite a recolha de *logs* de diversas fontes:

- Máquinas da infraestrutura;
- Aplicações - Java, Python, Go, Ruby, .Net, PHP, NodeJS, Javascript e Android;
- Ambiente Docker;
- Servidores na cloud - AWS, Azure, Google Cloud e Heroku;

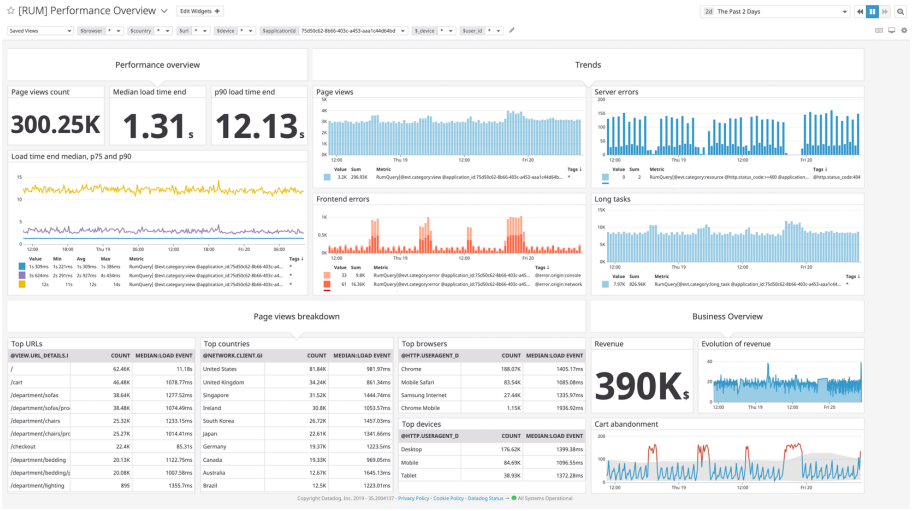


Figura 8: Exemplo de um *dashboard* do Datadog sobre *logs*

O *dashboard* da Figura8 permite a visualização de métricas relacionadas com a utilização dos websites monitorizados. É possível monitorizar a *performance* das páginas ou do código do *frontend*, a quantidade de pedidos realizados por página ou os erros relacionados com o servidor ou com as páginas dos websites.

Bazaarvoice

O Bazaarvoice [4] é um serviço que oferece uma solução que permite a marcas e retalhistas recolher e mostrar vários tipos de conteúdo gerado por utilizadores, como avaliações de produtos, perguntas e respostas e conteúdo de redes sociais nos seus *websites*. A equipa de desenvolvimento cresceu desde 30 pessoas para mais de 200 em apenas alguns anos. A manutenção desde rápido crescimento obrigou este serviço a encontrar maneiras de acelerar a inovação e a escalada sustentável da sua equipa de desenvolvimento [15]. Este serviço utiliza inúmeras tecnologias e, por essa razão, é importante utilizar uma ferramenta de monitorização que pudesse recolher métricas de todas as tecnologias utilizadas, bem como seja capaz de adicionar facilmente novas tecnologias. Para esse efeito foi escolhido o Datadog, pois possui inúmeras integrações com tecnologias, permitindo que as equipas de desenvolvimento criassem *dashboards* para cada um dos seus sistemas de maneira quase automática [15].

4 https://www.bazaarvoice.com

2.2.2 *Loggly*

O Loggly [5] é um serviço de monitorização na *cloud* que, tal como o Datadog, trabalha como SaaS. Com esta ferramenta é possível armazenar e monitorizar *logs* de toda a infraestrutura.

Permite a integração com ferramentas de DevOps, tais como o GitHub ou o Slack, disponibilizando alertas para estas ferramentas e a partilha e edição colaborativa de dashboards. Possui integrações com diversas fontes de *logs*, tais como serviços da Amazon ou aplicações como o Docker ou o Django. Possui um plano grátis e diversos planos pagos.

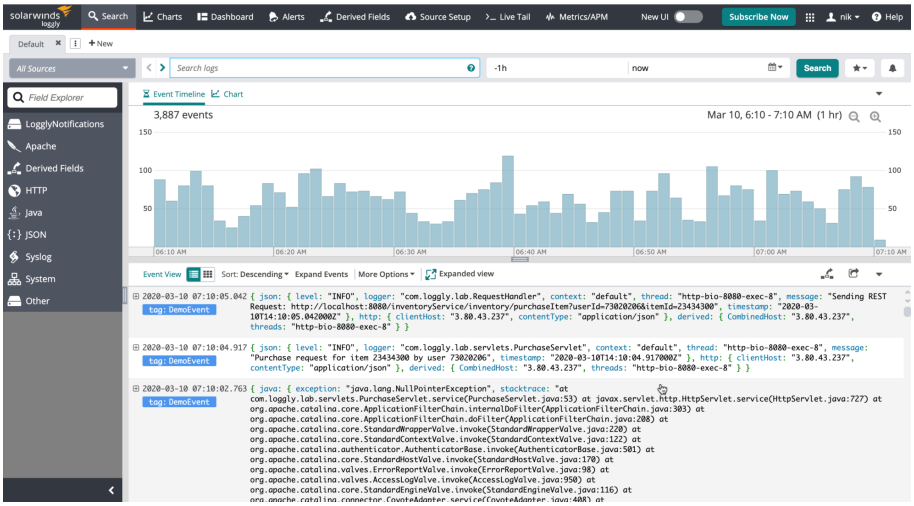


Figura 9: Exemplo de um *dashboard* do Loggly

Na Figura9 está ilustrado um *dashboard* do Loggly sobre dados de demonstração. Neste dashboard é possível pesquisar por *logs* específicos e ordená-los de várias formas.

Stanley Black&Decker

A Stanley Black&Decker é uma empresa que providencia ferramentas, armazéns, segurança eletrónica e comercial e soluções personalizadas para diversas áreas, desde montagem automóvel, construção industrial, aviação ou tecnologia [16]. A empresa utiliza o Loggly para monitorizar os *logs* das suas soluções, dando às suas equipas ferramentas para solucionar problemas nas aplicações móveis e de *Internet of Things (IoT)*, suportando monitorização de *performance*, segurança e de cumprimento com o regulamento de segurança de dados de cartões de crédito e oferecendo também grande escalabilidade à medida que novas soluções são lançadas [17].

5 https://www.loggly.com/

2.2.3 *Nagios*

O Nagios [6] é um sistema de monitorização *open-source* cujo propósito é a monitorização de aplicações , sistemas e serviços [18]. Possui diversas soluções comerciais, como o Nagios XI, o Nagios Log Server e o Nagios Fusion. Todas estas soluções derivam da solução *open-source*, o Nagios Core. Esta solução permite a monitorização de serviços de rede (HTTP, SMTP, etc.), recursos de sistema (utilização do disco, carga do processador, etc.), aplicações (através de *plugins*) ou o lançamento de alertas, por email.

Esta tecnologia utiliza agentes para recolher métricas de diversas fontes e pode correr testes funcionais para detetar problemas. Quando deteta alguma anomalia tenta correr o teste ou recolher a métrica um número específico de vezes (personalizável) e, se a anomalia continua, lança um alerta.

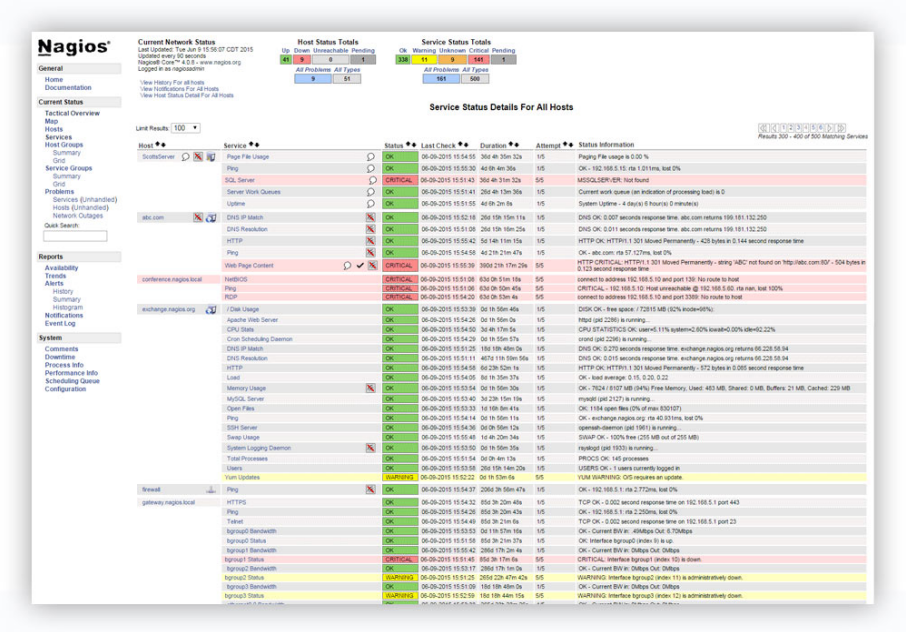


Figura 10: Exemplo de um *dashboard* de visão centralizada do Nagios

No *dashboard* da Figura10 é possível ter uma visão centralizada de toda a infraestrutura monitorizada, revendo o estado e a informação relativa a todas as operações.

6 https://www.nagios.com/

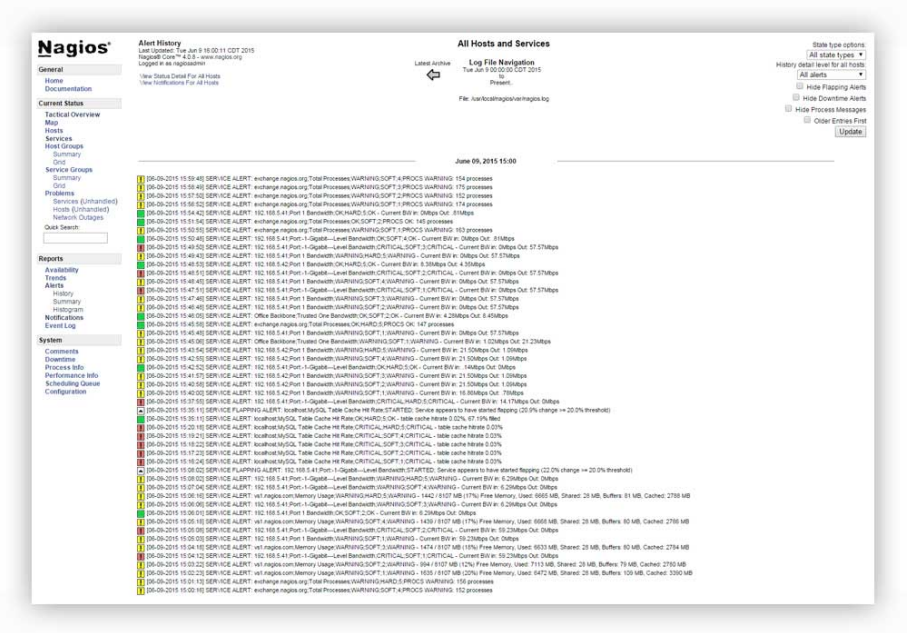


Figura 11: *Dashboard* de alertas do Nagios

No *dashboard* da Figura11 é possível criar, editar, eliminar e observar todos os alertas enviados, sendo estes divididos por gravidade: verde - gravidade baixa; amarelo - gravidade média; vermelho - gravidade alta.

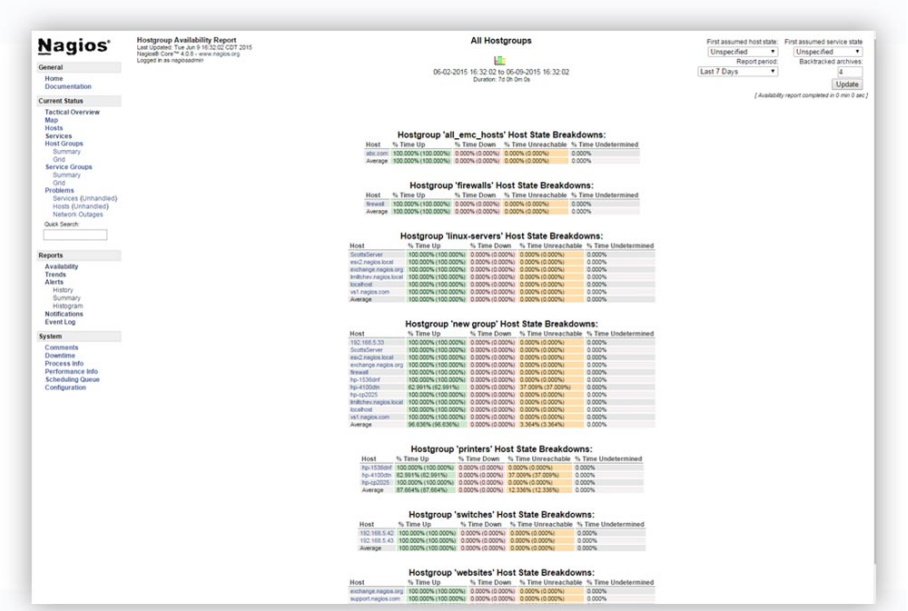


Figura 12: *Dashboard* de relatórios do Nagios

No *dashboard* da Figura12 é possível observar relatórios criados pelo Nagios sobre a percentagem de tempo em atividade de toda a infraestrutura monitorizada.

Network's Server Monitoring and Analysis Using Nagios

Em [19] é apresentada uma ferramenta de monitorização de rede. O Nagios é utilizado para monitorizar o servidor e as aplicações que correm na sua rede, através de verificações do estado do sistema, notificando a equipa quando encontra alguma anomalia. A utilização de *plugins* permite ajudar a notificar o gestor da rede quando ocorrem problemas nesta.

2.2.4 *ElasticStack*

O *ElasticSearch Logstash Kibana - ElasticStack (ELK)* é um conjunto de diversas tecnologias *open-source* que quando utilizadas em conjunto constituem uma ferramenta de monitorização. O Logstash [7] é uma ferramenta de processamento de dados que recolhe dados de múltiplas fontes, realiza operações de filtragem e processamento e encaminha-os para múltiplos destinos, possuindo integrações com inúmeras fontes, podendo recolher métricas ou texto. O ElasticSearch [8] é uma base de dados orientada a documentos, RESTful, em que cada documento armazenado tem um índice atribuído, para uma mais rápida pesquisa sobre os mesmos e um *timestamp* associado. Foi construída para lidar com grandes volumes de dados, tendo sempre grande disponibilidade, a capacidade de ser distribuída por múltiplas máquinas, tolerante a falhas e altamente escalável. O Kibana [9] é a camada de visualização desta ferramenta, permitindo a visualização, em *dashboards*, dos dados armazenados no ElasticSearch e possuindo uma página para pesquisas nesses dados.

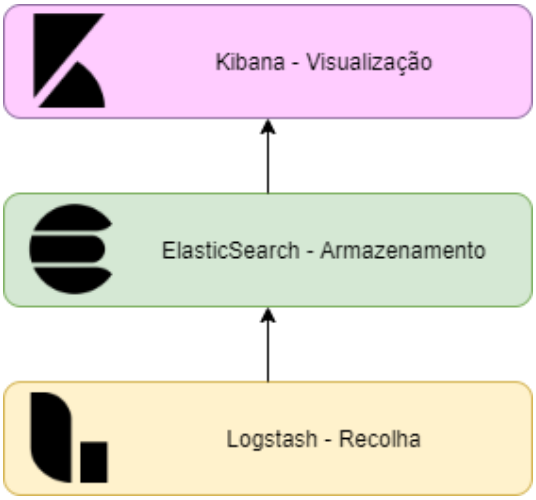


Figura 13: Fluxo de dados do ElasticStack

7 https://www.elastic.co/logstash
8 https://www.elastic.co/products/elasticsearch
9 https://www.elastic.co/products/kibana

Como é possível observar na Figura13, os dados, quando estas 3 tecnologias são utilizadas em conjunto, são recolhidos pelo Logstash e enviados, por uma API, para o ElasticSearch, para serem armazenados. Posteriormente, estes são recolhidos pelo Kibana, para serem apresentados em *dashboards* ou então serem efetuadas pesquisas sobre eles.

Log-Based Cloud Monitoring System for OpenStack

Em [20] é apresentada uma ferramenta de monitorização dos *logs* dos componentes do OpenStack [10], uma ferramenta *open-source* de gestão de serviços na *cloud*, em tempo real, gerando alertas quando é encontrado um erro nos *logs* recolhidos. Desta forma, a ferramenta facilita a leitura dos registos por parte dos administradores.

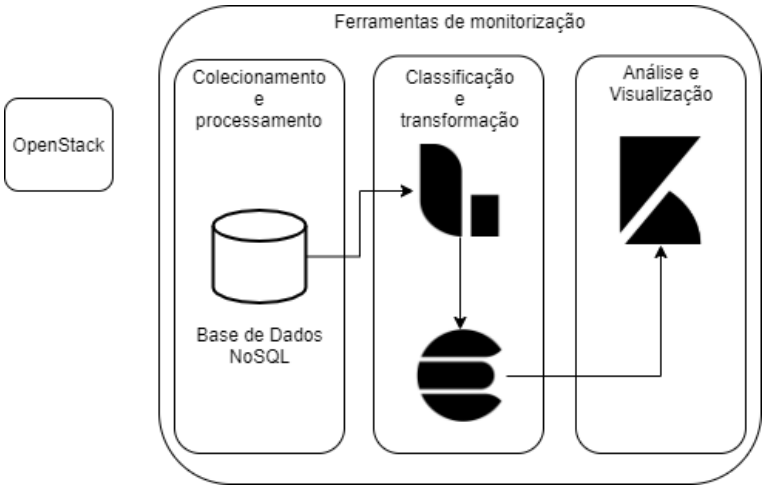


Figura 14: Arquitetura da ferramenta de monitorização da *cloud* OpenStack, como proposto em [20]

Na Figura14 encontra-se exposta a arquitetura da ferramenta de monitorização, que consiste em 3 módulos: Recolha e Processamento, Classificação e Transformação e Análise e Visualização. No primeiro módulo é utilizada uma base de dados NoSQL para armazenar os dados, que são recolhidos pelo Logstash através das APIs do OpenStack e enviados para a base de dados. No segundo módulo os dados são classificados consoante a sua origem e transformados para um formato único pelo Logstash, que os envia para o Kibana para visualização e para o ElasticSearch, para serem armazenados para futuras pesquisas textuais. Por último, os dados são apresentados no Kibana, em *dashboards*, podendo também serem realizadas *queries* ao ElasticSearch.

10 https://www.openstack.org/

2.2.5 *Grafana*

O Grafana [11] é um *software* de visualização e criação de *dashboards open-source*, que possui integrações com inúmeras fontes. Possui também diversos tipos de visualizações, tais como campos de textos, gráficos de barras ou de linhas, tabelas, entre outros, permitindo desta forma uma grande personalização da monitorização. Esta ferramenta tem várias funcionalidades, como criação, edição, visualização e partilha de *dashboards*, havendo a possibilidade de serem provenientes de múltiplas fontes, um sistema de alertas, enviados por email, para o Slack ou para outros serviços e a realização de *queries* sobre as métricas armazenadas nas fontes de dados.

Real-Time I/O-Monitoring of HPC Applications with SIOX, Elasticsearch, Grafana and FUSE

Em [21] é apresentada uma ferramenta de monitorização que utiliza Grafana para visualização da informação. Esta ferramenta vem resolver um problema relacionado com a *performance* dessas aplicações, devido aos padrões de leitura e escrita utilizados por estas, que muitas vezes custava bastantes recursos. A ferramenta foca-se em detetar em que aplicações existem os problemas mencionados anteriormente, utilizando uma abordagem diferente: Enquanto outras ferramentas com estes objetivos recolhem informação através de variáveis de sistemas, esta ferramenta recolhe-as diretamente das aplicações. As métricas analisadas referem-se aos tempos de leitura e escrita, número de bytes realizados nas operações e número de operações realizadas.

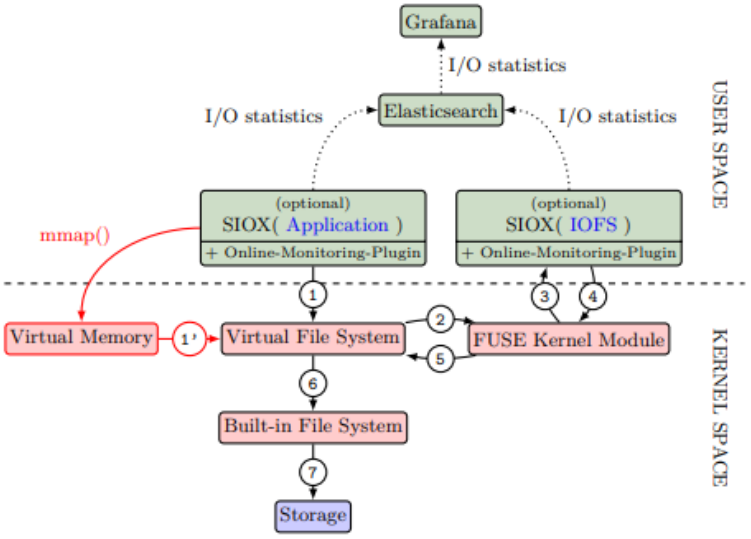


Figura 15: Arquitetura da ferramenta de monitorização, como em [21]

11 https://grafana.com/

Na Figura15 está ilustrada a arquitetura da ferramenta de monitorização. As métricas são enviadas pelas aplicações para o ElasticSearch, que as armazena. Posteriormente são recolhidas pelo Grafana, para serem visualizadas em *dashboards*.

2.3 SISTEMAS DE MONITORIZAÇÃO NA SAÚDE

Tal como em outras áreas, a área da saúde depende cada vez mais de sistemas informáticos para realizar as suas atividades. São utilizados hoje em dia sistemas informáticos para a gestão de pacientes ou para armazenamento de informação clínica. Esses sistemas precisam de ser monitorizados, para que ocorram o mínimo possível de falhas na sua atividade, minimizando os custos com manutenção e a quantidade de informação perdida e melhorando a qualidade do serviço prestado.

2.3.1 *HealthIRA*

A HealthIRA é uma solução de integração para sistemas hospitalares. Esta solução surgiu para dotar os administradores de sistemas hospitalares de uma vista sobre os níveis de produção da sua instituição, bem como para permitir às equipas de informática um maior controlo, perceção e monitorização da informação trocada. Foca-se nas aplicações que estão em produção, nas quais é difícil fazer alterações, seja porque não existe acesso ao seu código-fonte ou porque tais mudanças iriam levar muito tempo a serem implementadas. Utiliza um *sniffer* escrito na linguagem Go, para recolha de mensagens, MirthConnect para a interpretação das mesmas, um balanceador de carga (HAProxy [12]), Logstash para recolha de *logs*, ElasticSearch para armazenamento de dados e Kibana para visualização de informação [22].

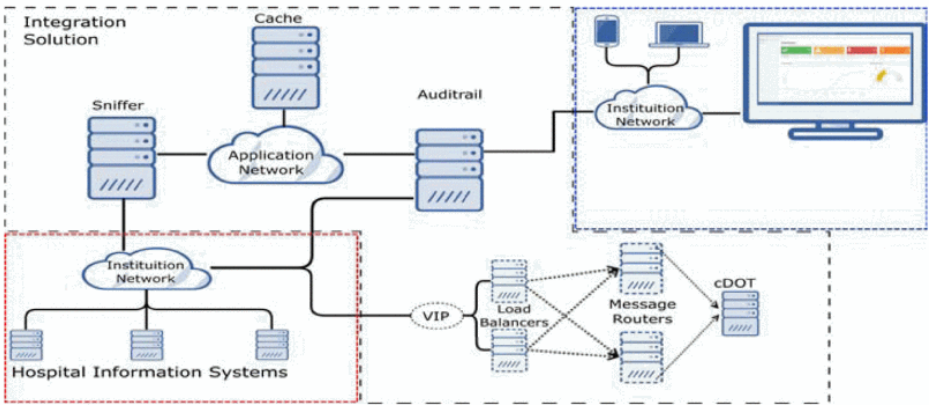


Figura 16: Arquitetura da HealthIRA, como em [22]

12 http://www.haproxy.org/

A Figura16 apresenta a arquitetura desta solução. O *sniffer* processa e extrai o conteúdo das mensagens HL7 recolhidas e posteriormente transforma-as para o formato JSON, sendo por último armazenadas localmente ou enviadas para o módulo de auditoria, para mais processamento.

Como forma de promover a interoperabilidade entre sistemas, a solução tem um módulo para ser utilizado como um servidor central de comunicação, para troca de mensagensem vários formatos, oferecendo alta disponibilidade e balanceamento de carga. Esse módulo consiste em duas instâncias de serviços HAProxy [13], com um IP virtual, assegurando que se se um servidor falhar, o outro estará disponível para manter o serviço.

O serviço de auditoria é baseado nas tecnologias do ELK. No servidor ElasticSearch (audiotrail) são armazenados as mensagens HL7 enviadas pelo *sniffer* e os logs recolhidos pelo Logstash, sendo os dados visualizados no Kibana. Esta informação é utilizada para monitorização do sistema hospitalar e da troca de mensagens efetuada no mesmo.

Outro módulo da solução é um servidor de cache, tornando mais rápido o acesso à informação. Este módulo consiste numa base de dados que implementa o conjunto de standards HL7 FHIR. Todas as mensagens recebidas pelo módulo de roteamento de mensagens são duplicadas e enviadas para a base de dados de cache.

Esta solução inclui também um módulo para transformação de documentos *online*, que permite a transformação de mensagens HL7 do formato HL7 v2 para o formato openEHR, com o objetivo de as mensagens de ambos os formatos serem armazenadas num único repositório de informação clínica, utilizando o *engine* MirthConnect para realizar a transformação.

2.4 Regulamento Geral de Proteção de Dados

O RGPD da *União Europeia (UE)*, em vigor desde 25 de Maio de 2018, é um marco na evolução da legislação europeia de proteção de dados [4]. Este regulamento cobre o processamento de dados pessoais de todos os residentes nos estados-membros da UE, independentemente do local em que o processamento ocorre.

13 http://www.haproxy.org/

Neste regulamento é designada dados pessoais a informação que, direta ou indiretamente, possa identificar um indivíduo. Essa informação inclui dados como o endereço IP, *cookies* e dados de localização. Segundo o RGPD, o tratamento de dados deve rege-se por 6 princípios [4]:

- Licitude, lealdade e transparência;
- Limitação de funcionalidades;
- Minimização dos dados;
- Exatidão;
- Limitação da conservação;
- Integridade e confidencialidade;

A mudança principal com a entrada em vigor desta legislação foi a necessidade de consentimento por parte do titular da informação. Outra mudança importante é o facto de ser obrigatório dar informação ao titular da informação relativamente ao local onde os dados são armazenados e sobre o periodo em que os dados são retidos, utilizando linguagem fácil de compreender.

2.4.1 *O Regulamento Geral de Proteção de Dados na saúde*

Na área da saúde, este regulamento trouxe benefícios e desafios. Por um lado, este permite a exceção por parte das aplicações e serviços da área de algumas das obrigações, incluindo a obrigatoriedade de requerer consentimento por parte dos titulares da informação para a realização de operações de tratamento de dados, pois essas operações são realizadas para fins de pesquisa científica (Artigo 9º).

Por outro lado, o regulamento deixa a isenção de regulamentação a cargo dos Estados-Membros (”Deverá igualmente caber ao direito da União ou dos Estados-Membros determinar se o responsável pelo tratamento que exerce funções de interesse público ou prerrogativas de autoridade pública deverá ser uma autoridade pública ou outra pessoa singular ou coletiva de direito público, ou, caso tal seja do interesse público, incluindo por motivos de saúde, como motivos de saúde pública e proteção social e de gestão dos serviços de saúde, de direito privado, por exemplo uma associação profissional”[4]), o que pode levar a interpretações diferentes no que toca ao tratamento de dados, podendo impedir a troca de dados entre serviços de saúde em países diferentes.

Para a plataforma AIDA, o maior desafio relacionado com este regulamento é a necessidade de, segundo o artigo 30º, ser obrigatório existir um registo das operações de tratamento de dados realizadas na plataforma, sendo uma prioridade do grupo de investigação a criação de mecanismos para esse efeito.

2.4.2 *GDPR-Compliant Personal Data Management: A Blockchain-Based Solution*

Em [23] é descrito o desenvolvimento de uma plataforma de gestão de dados pessoais que atua em conformidade com o RGPD. Os objetivos da plataforma são providenciar mecanismos, tanto para os titulares da informação como para a plataforma, para processamento de dados, bem como de garantir a transparência dos dados oferencendo funcionalidades relacionadas com a tecnologia *blockchain*. A arquitetura proposta encontra-se ilustrada na Figura17.

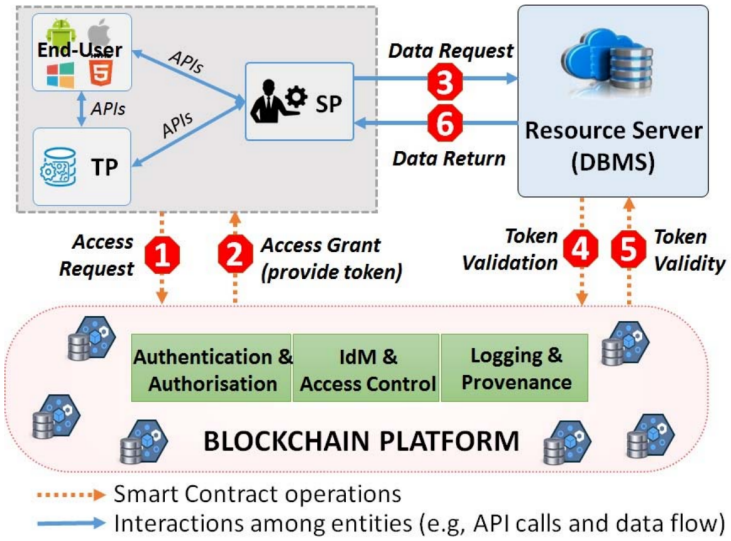


Figura 17: Arquitetura da plataforma de gestão de dados, como em [23]

Nesta plataforma, os mecanismos relacionados com o cumprimento do RGPD são incluidos numa rede de *blockchain*. Os componentes de *logging* e proveniencia dos dados são implementados também na rede *blockchain*. A Figura17 mostra também que todas as atividades relacionadas com os dados são autenticadas e autorizadas pela rede *blockchain* (pontos 1 e 2). A *blockchain*, servindo como um servidor de autenticação e autorização, emite um *token* de acesso, de modo a dar acesso a uma determinada informação. O providenciador do serviço recebe o token (ponto 2) e utiliza-o para requisitar a informação (ponto 3). Posteriormente, o servidor onde a informação está armazenada comunica com a plataforma *blockchain* para validar o acesso (passos 4 e 5), devolvendo posteriormente os dados (passo 6) [23]. Este procedimento garante que o acesso é válido e utilizado da forma correta pelo utilizador em questão.

3

METODOLOGIAS E TECNOLOGIAS

Neste capítulo são apresentadas as metodologias de investigação e de avaliação utilizadas nesta dissertação bem como as tecnologias que fazem parte da solução e análises comparativas entre essas tecnologias e tecnologias com os mesmos fins.

3.1 METODOLOGIA DE INVESTIGAÇÃO

No desenvolvimento da solução foi utilizada a metodologia Design Science Research (DSR) [24]. Esta metodologia divide-se em 6 fases:

- Identificação do problema e motivações - Nesta fase o objetivo é definir os problemas que necessitam de resolução e justificar o valor que teria o desenvolvimento de uma solução para esses problemas. Justificar o valor da solução tem diversos benefícios - motiva o investigador ou equipa de investigação a desenvolver a solução e a aceitar os resultados e ajuda a compreender o raciocínio associado à perceção do problema por parte da equipa de investigação;

- Definição dos objetivos para desenvolver a solução - Esta segunda fase consiste na inferência dos objetivos para o desevolvimento de uma solução, a partir dos problemas apontados na fase anterior e do que é possível de ser realizado. Os objetivos podem ser quantitativos - como identificar porque a solução pensada seria melhor do que soluções existentes - ou qualitativos - como uma descrição de como a solução suportaria a adição de novos componentes para solucionar problemas não identificados no desenvolvimento desta;

- Desenho e desenvolvimento - Na terceira fase é criada a solução. Conceptualmente, uma solução desenvolvida utilizando esta metodologia deve ser uma solução cujo desenho inclua as contribuições para a resolução dos problemas. Esta fase inclui também as decisões relativas ao funcionamento e arquitetura da solução;

- Demonstração - Nesta fase o objetivo é demonstrar que a solução desenvolvida resolve os problemas apresentados. Podem ser utilizados diversos métodos, tais como experiências, simulações, casos estudo, provas ou outras atividades apropriadas;
- Avaliação - Na penúltima fase o objetivo é avaliar quão bem a solução desenvolvida resolve os problemas apresentados. Isto envolve comparar os objetivos definidos na segunda fase da metodologia com os resultados obtidos na fase de demonstração;
- Comunicação - A última fase consiste na comunicação do problema, a sua importância, a solução e a sua utilidade e o rigor do seu desenvolvimento, através de publicações.

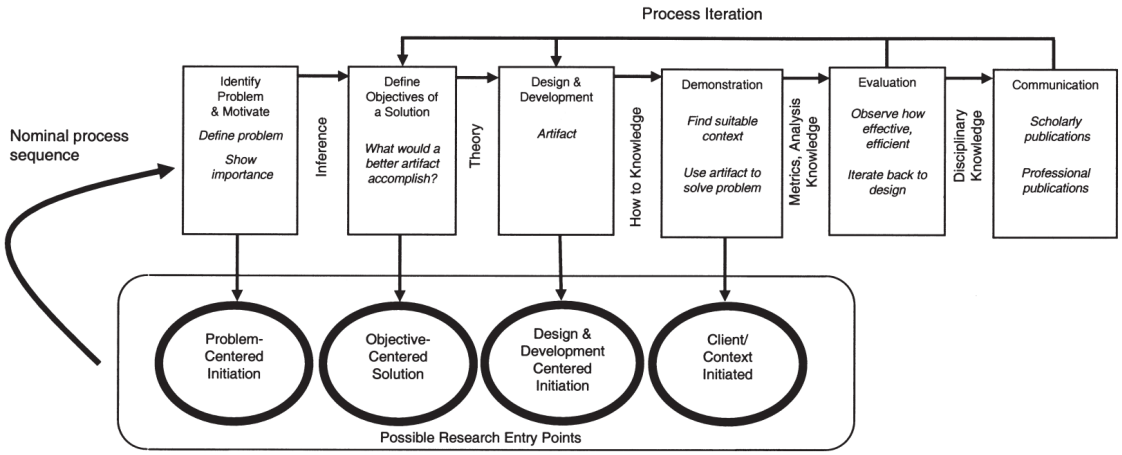


Figura 18: Esquema da aplicação da metodologia DSR, como proposta em [24]

Como ilustrado na Figura18, a metodologia pode ser aplicada de diversas formas: se for aplicada numa abordagem centrada no problema, esta é iniciada na primeira fase; se for aplicada numa abordagem centrada nos objetivos, é iniciada na segunda fase; se for aplicada num contexto centrado na solução, é iniciada na terceira fase; por último, se for centrada em analisar uma solução em funcionamento deve ser iniciada na fase 4. É de realçar que o processo associado à metodologia prevê que durante a aplicação desta os investigadores possam passar múltiplas vezes pelas diversas fases, à exceção da fase inicial, de identificação do problema.

Neste projeto, esta metodologia irá ser aplicada com uma abordagem centrada no problema, pelo que irão ser aplicadas as 6 fases da metodologia.

3.2 TECNOLOGIAS DE DESENVOLVIMENTO

3.2.1 *MirthConnect*

O Mirth Connect é um *engine open source* de integração de mensagens de saúde que utiliza uma arquitetura baseada em canais para interligar sistemas, cuja utilização permite facilitar a integração entre sistemas que utilizem formatos diferentes. Permite a filtragem, transformação e roteamento de mensagens, baseando-se em regras definidas pelo utilizador [25].

Possui um servidor de comunicação, que atua como um *container* de canais e um cliente, que atua através de uma interface, utilizada para criar, testar e monitorizar canais.

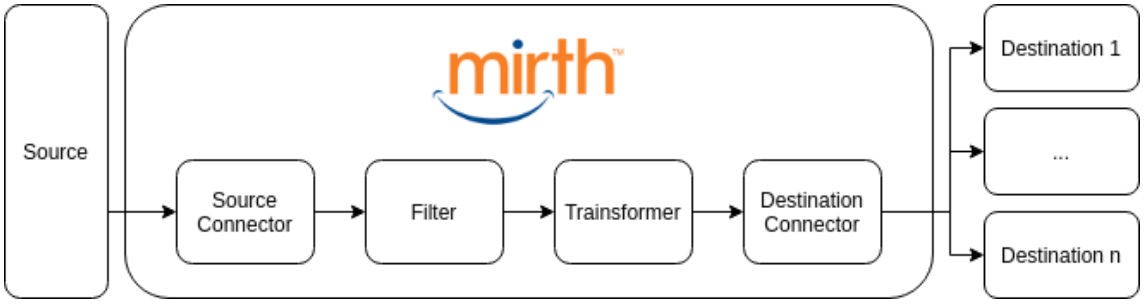


Figura 19: Arquitetura de um canal Mirth

Tal como ilustrado na Figura19, um canal Mirth é constituído por 4 componentes: *Source Connector*, *Filter*, *Transformer* e *Destination Connector*. O *source connector* recebe mensagens de um componente e pode ser configurado para ficar à escuta ou para recolher mensagens. O componente *Filter* determina quais as mensagens que devem ser aceites, baseando-se num conjunto de regras configuráveis. O componente *Transformer* extrai e processa os dados das mensagens para o formato no qual são enviadas para o destino. O componente *Destination Connecter* gere o roteamento das mensagens processadas, podendo enviar a mesma mensagem para diversos destinos. Atualmente, a plataforma AIDA está a ser preparada para que este *engine* seja utilizado para efetuar as conexões entre os seus componentes.

Em [26] é abordada a implementação do MirthConnect como um servidor de comunicação para converter mensagens HL7, provenientes de um sistema de gestão de informação de um laboratório, em dados para alimentar duas aplicações. Para esse efeito foram criados 3 *channels* Mirth com funções distintas:

- *Channel* 1: Este *channel* recolhe as mensagens enviadas pelo sistema de origem, mapeia os valores e, consoante o tipo de mensagem que chega, envia-a para o *channel* 2 ou 3.
- *Channel* 2: Este *channel* recebe as mensagens destinadas para a primeira aplicação. As mensagens são validadas, de modo a que todos os dados necessários estejam

completos. Posteriormente estas são transformadas para o formato no qual a aplicação trabalha e são enviadas para o seu destino.

- *Channel* 3: Este *channel* recebe as mensagens destinadas para a segunda aplicação. Tal como no channel anterior, as mensagens são validadas e posteriormente transformadas para o formato no qual esta aplicação trabalha. Após a transformação as mensagens são guardadas em ficheiros.

A utilização do MirthConnect como servidor de comunicação permitiu que a transferência de dados clínicos, através de mensagens HL7, entre o sistema de gestão e as duas aplicações fosse realizada de maneira eficiente e automatizada, reduzindo o tempo de operação substancialmente.

3.2.2 *Tecnologias de virtualização*

A virtualização de sistemas tem ganho cada vez mais importância nos dias de hoje, sendo utilizado em diversos contextos. Ao libertar o software das dependências ao nível da *interface* e de recursos, este género de tecnologias aumentam a interoperabilidade de sistemas e a versatilidade das plataformas [27].

A criação de tecnologias que tornassem a migração de aplicações entre máquinas rápida e fácil, bem como permitissem aplicar os pontos referidos no parágrafo anterior foi essencial para dar resposta ao mercado atual.

Docker

O Docker é uma tecnologia de virtualização que recorre ao uso de LinuxContainers [28] para padronizar uma aplicação ou parte de uma aplicação. Esta tecnologia aplica o conceito de virtualização ao nível do sistema operativo, simulando o seu funcionamento através de *containers*. Um *container* é uma unidade *standard* de *software* que encapsula o código e as suas dependências, permitindo assim que o código possa correr em qualquer ambiente computacional [29].

Como ilustrado na Figura20, aplicações encapsuladas em *containers* Docker atuam sobre o sistema operativo da máquina, cada uma com o seu próprio sistema operativo virtual. Com a aplicação encapsulada é possível migrá-la de uma máquina para outra sem ser necessário fazer configurações.

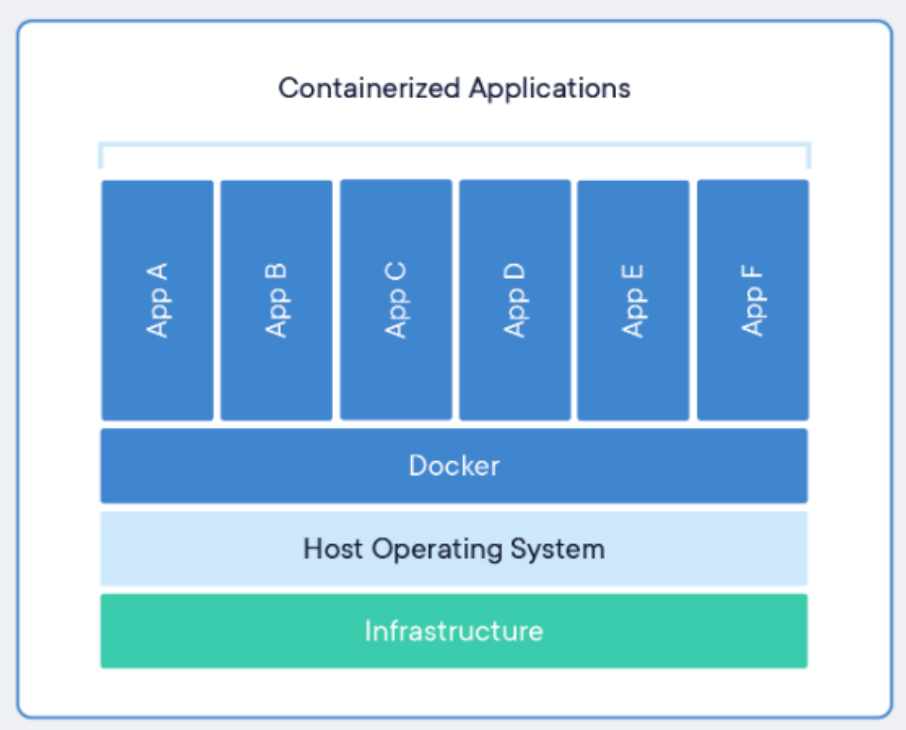


Figura 20: Esquema de um sistema baseado em *containers*

Máquinas Virtuais

O conceito de virtualização pode ser aplicado de forma diferente da abordagem utilizada pelo Docker. Ao invés de ser aplicado ao nível do sistema operativo - focando-se em simular o funcionamento deste - As tecnologias baseadas em máquinas virtuais, como são os casos do VMWarew [1] e da Oracle VirtualBox [2], aplicam-no no nível físico da máquina, focando-se em simular *hardware*. Desta forma este conceito pode ser aplicado não só a componentes de *hardware*, como discos, processadores ou memória, mas à máquina como um todo.

Existem 2 tipos de tecnologias baseadas em máquinas virtuais: as de tipo 1, que correm diretamente no *hardware* e as de tipo 2, que correm como uma camada adicional de *software* no sistema operativo.

Num sistema baseado em máquinas virtuais, como o ilustrado na Figura21, cada máquina virtual tem os seus próprios recursos de *hardware* virtuais, geridos pelo utilizador em cada uma delas. Desta forma é possível dar mais recursos a aplicações que necessitem de mais poder de computação e menos às aplicações que necessitem de menos.

1 https://www.vmware.com/
2 https://www.virtualbox.org/

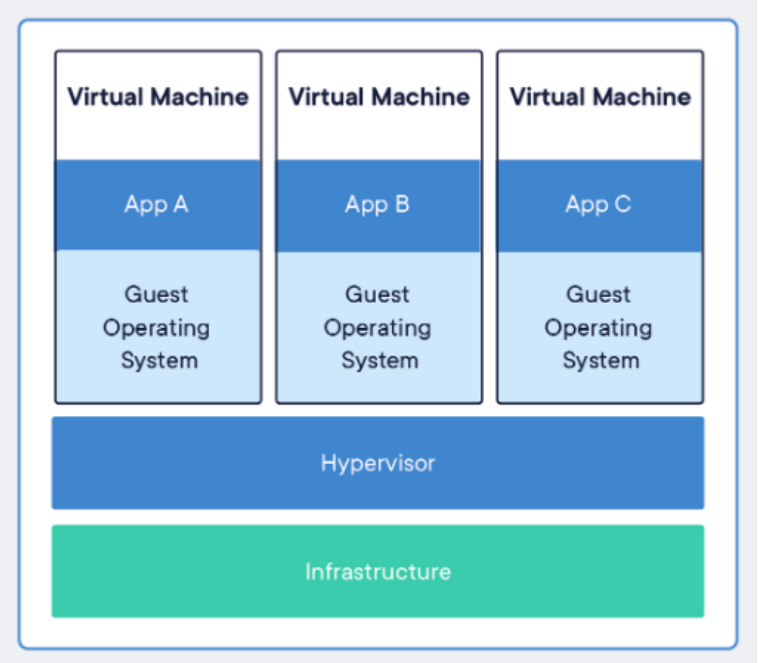


Figura 21: Esquema de um sistema baseado em máquinas virtuais

3.2.3 *Bases de Dados Não Relacionais*

Atualmente muitas aplicações utilizam o modelo relacional, através da linguagem SQL, como o método principal para o armazenamento de dados em bases de dados. Contudo, estão a ser gerados conjuntos cada vez maiores de dados, sendo importantes soluções que consigam lidar com tais números. Foi nesse sentido que apareceu o movimento *Not Only SQL (No SQL)*, pois as soluções relacionais não conseguem acompanhar a rapidez pedida para um bom funcionamento das aplicações, apesar de a qualidade e quantidade de hardware existente ser muito maior do que quando o modelo relacional foi introduzido.

Na área da saúde este modelo tornou-se importante, pois em ambiente hospitalar é gerada uma cada vez maior quantidade de informação, devido à manutenção de registos, existência de requisitos regulatórios e de conformidade e de modo a ser possível prestar um melhor serviço aos pacientes [30]. A quantidade de dados que circulam num sistema como a AIDA pode ser utilizada para identificar necessidades, providenciar serviços e prever e prevenir crises [31], ou então para monitorizar o funcionamento desse sistema, de modo a garantir segurança e previsão de eventos.

Este género de solução tornou-se importante nesse sentido, pois oferece melhor desempenho em termos de leitura e escrita, maior escalabilidade e capacidade de resposta a pedidos, quando comparado com o modelo relacional. Neste modelo a informação é guardada em estruturas de dados com formato conhecido, como JSON ou XML, trazendo assim maior capacidade de compreensão do utilizador.

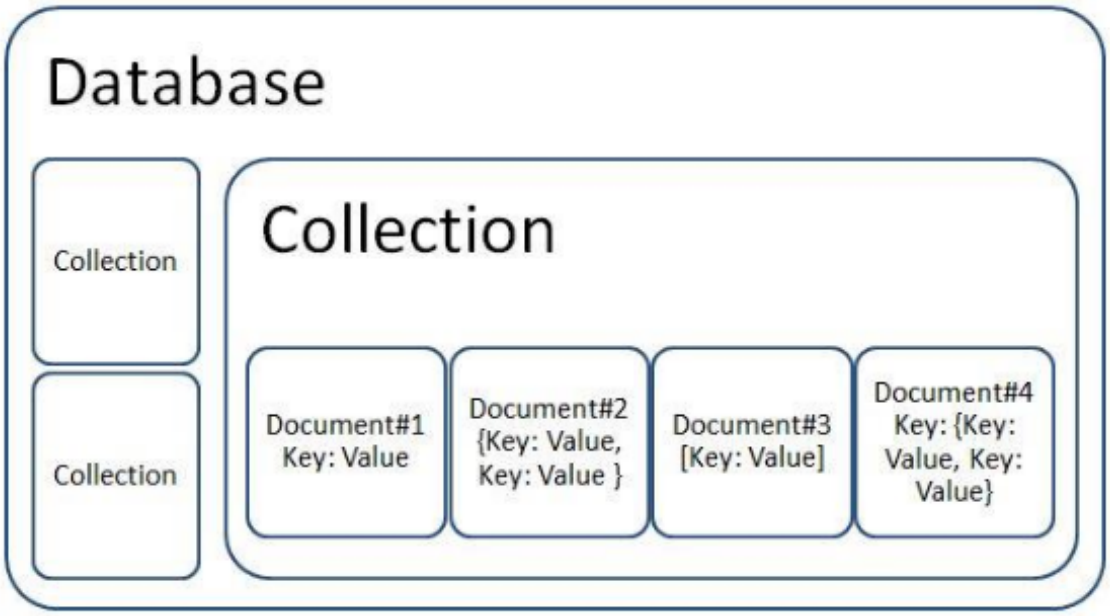


Figura 22: Exemplo da estrutura uma base de dados orientada a documentos

Uma base de dados orientada a documentos, tal como a ilustrada na Figura22 tem uma estrutura muito diferente de uma base de dados relacional. Neste tipo de base de dados a informação é armazenada em documentos, utilizando pares chave-valor, sendo os documentos divididos em *collections*. A utilização de pares chave-valor para armazenar a informação permite que os documentos não tenham uma estrutura definida.

ElasticSearch

ElasticSearch é uma base de dados orientada a documentos, RESTful e *open-source*, que surgiu do Apache Lucene [3], uma biblioteca de Java de pesquisa e indexação de texto. Nesta base de dados, cada documento armazenado tem um índice atribuído, para uma mais rápida pesquisa sobre os mesmos, e um *timestamp* associado. Foi construída para lidar com grandes volumes de dados, tendo sempre grande disponibilidade, a capacidade de ser distribuída por múltiplas máquinas, tolerante a falhas e altamente escalável. As suas valências permitem-lhe ser utilizado para monitorizar aplicações, através dos registos que são gerados durante a utilização destas ou para funções de pesquisa, devido à estrutura peculiar de armazenamento de informação que apresenta. É normalmente utilizado em conjunto com o Kibana, um plugin que permite a visualização da informação, através de *dashboards*.

Cada entidade ElasticSearch denomina-se *node*, sendo possível operar com apenas um *node*, em modo *single-server* ou com diversos, denominando-se o seu conjunto de *cluster*. O

3 https://lucene.apache.org/

ElasticSearch funciona de modo distribuído por natureza, atuando como uma rede *Peer-to-Peer (P2P)*. Uma rede P2P pode ser definida como um conjunto de entidades que partilham recursos de hardware (neste caso, partilham capacidade de armazenamento) e acessíveis entre si diretamente [32].

A divisão dos documentos é feita em índices estando cada índice dividido em *shards*, cujo número é definido aquando da criação do índice. Os documentos têm um tipo associado, que pode ser definido quando o índice é criado ou ser definido de forma dinâmica pelo ElasticSearch quando o documento é inserido [33]. É possível definir que parâmetros são filtrados ou removidos e a forma como os parâmetros de texto são armazenados, através do mapeamento do mesmo [34].

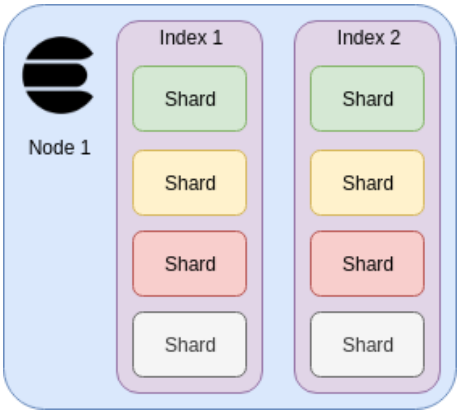


Figura 23: Modo *single-server*

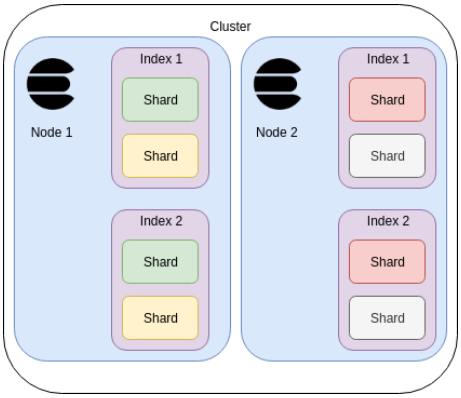


Figura 24: Modo distribuído

O exemplo exposto nas figuras 23 e 24 mostra um servidor ElasticSearch com 2 índices, cada um deles com 4 *shards*, primeiro em modo *single-server* (Figura23) e, posteriormente, ao ser adicionado um novo *node* ElasticSearch, em modo distribuído (Figura24). Os índices são replicados para o novo *node* e os *shards* são repartidos pelos 2 *nodes* existentes no *cluster*, de forma a balancear a carga dos servidores.

Na Figura23, os pedidos efetuados sobre um índice são enviados para todos os *shards* desse índice, executados em cada um deles, agregada a informação encontrada e enviada a resposta ao cliente. Se um pedido for sobre todos os índices então esse pedido é executado em todos os *shards* de ambos os índices.

No caso da Figura24, os pedidos são replicados pelos 2 *nodes*, sendo posteriormente tratados por cada um como no modo *single-server*. A inserção dos documentos funciona da mesma maneira em ambos os exemplos, com o algoritmo do ElasticSearch a definir em que *shard* o documento é armazenado e indexado [35], para ser possível repartir a quantidade de informação pelos *shards* em partes iguais.

Apesar das muitas diferenças entre ElasticSearch e o modelo relacional, existem algumas semelhanças entre alguns conceitos de ElasticSearch e conceitos do modelo relacional [36].

Tabela 1: ElasticSearch vs BD relacional

ElasticSearch	Relacional
Índice	Base de Dados
Tipo de documento	Tabela
Documento	Linhas
Elementos	Colunas

É possível observar algumas semelhanças entre os conceitos de ElasticSearch e de uma base de dados relacional mencionados na Tabela 1. Cada índice de ElasticSearch pode ser definido como uma base de dados num *engine* como Oracle ou MySQL. Também existem semelhanças entre o tipo dos documentos em ElasticSearch e o conceito de tabela do modelo relacional, bem como entre o conceito de documento (e os seus elementos) e os conceitos de linha e coluna de uma base de dados relacional. Tanto o tipo de documento de ElasticSearch como a tabela, no modelo relacional, definem a estrutura do que é armazenado. Cada documento corresponde a uma linha no modelo relacional, pois todos os documentos pertencentes a um tipo têm uma estrutura semelhante, tal como todas as linhas de uma tabela têm a mesma estrutura, podendo ou não ter elementos nulos.

A utilização desta base de dados, em conjunto com uma ferramenta de monitorização, bem como com uma ferramenta de processamento de dados pode permitir a monitorização de vários contextos relacionados com a plataforma AIDA, como o desempenho dos servidores e componentes da plataforma ou as operações de tratamento dos dados pessoais dos pacientes.

Apache Solr

O Apache Solr [4] é uma base de dados orientada a documentos, *open-source* que funciona no modelo *master-slave* e que, tal como o ElasticSearch, surgiu do Apache Lucene. Possui uma API REST e bibliotecas de diversas linguagens, como Java, Python, Ruby ou C#. Esta base de dados consegue armazenar dados com ou sem um esquema definido, de várias fontes. Tal como o ElasticSearch, funciona à base de índices e as suas principais valências incluem otimização para grandes volumes de dados, alta escalabilidade e tolerância a falhas [37].

4 https://lucene.apache.org/solr/

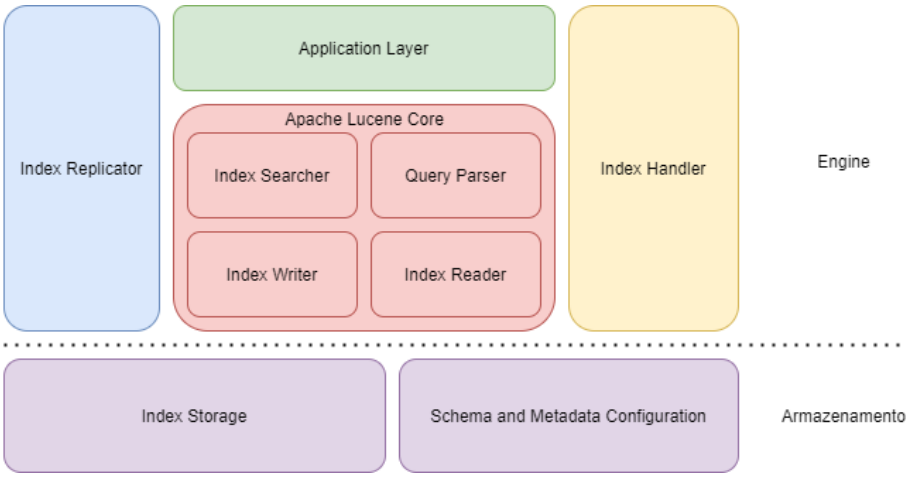


Figura 25: Arquitetura da base de dados Solr, como em [38]

Como ilustrado na Figura25, a base de dados Solr é constituida por múltiplos componentes. O *index replicator* é responsável pela replicação dos índices, do servidor mestre para os restantes. O servidor mestre mantém os índices atualizados e os restantes servidores são responsáveis por comunicar com este para que a informação seja replicada, garantindo desta forma alta disponibilidade.

A arquitetura da base de dados possui também um núcleo de componentes do Apache Lucene. O *query parser* trata do processamento das *queries* enviadas pelos clientes, utilizando classes da biblioteca do Apache Lucene. O *index reader* e o *index writer* providenciam, respetivamente, o acesso aos índices e a possibilidade de criar novos índices. Por seu lado, o *index searcher* é responsável por retornar os resultados das pesquisas enviadas pelo *query parser*.

O *index handler*, outro componente da base de dados, é o responsável pela adição, atualização ou eliminação dos documentos dos índices. Por último, a *aplication layer* é responsável pela receção e envio de pedidos das aplicações cliente.

Análise Comparativa

Tanto ElasticSearch como Solr são tecnologias baseadas no Apache Lucene. Por essa razão, têm bastantes semelhanças. Ambas guardam a informação em índices e as suas funcionalidades de pesquisa sobre a informação são eficientes. Contudo, existe uma diferença fulcral entre ambas: ElasticSearch é orientada a dados com registo temporal ou com registos textuais, enquanto que Solr é mais eficiente quando se trata de dados estáticos. Esta é uma diferença crucial na escolha da tecnologia a utilizar, pois sendo o objetivo da ferramenta a desenvolver a monitorização dos registos textuais das operações de tratamento de dados realizadas na plataforma AIDA, o ElasticSearch serve melhor o propósito da ferramenta.

3.2.4 *Logstash*

O Logstash [5] é uma ferramenta de processamento de dados que recolhe dados de múltiplas fontes, realiza operações de filtragem e processamento e encaminha-os para múltiplos destinos. Faz parte do ElasticStack (constituído pelo ElasticSearch, o Kibana e o Logstash), pelo que é benéfica a utilização destas 3 ferramentas na implementação, pois foram criadas para serem utilizadas em conjunto.

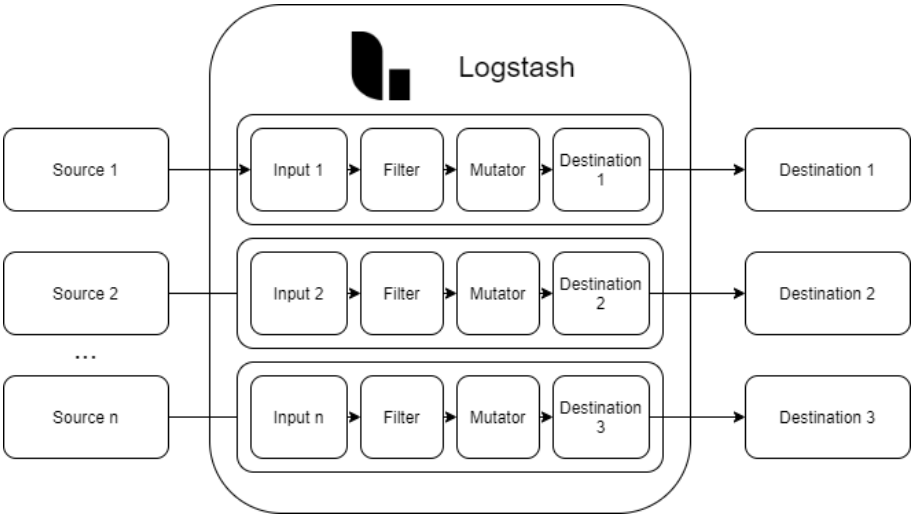


Figura 26: Exemplo do funcionamento do Logstash

O Logstash trabalha através de *pipelines*, utilizando plugins para a recolha e envio de dados, o que lhe confere uma grande variedade de fontes e destinos. Como ilustrado na Figura26, os dados são recolhidos pelo *plugin* de recolha de dados, podendo esses dados ser originários de bases de dados, tais como MySQL ou Oracle ou de ElasticSearch, docker ou simplesmente ficheiros. Posteriormente são filtrados, segundo os parâmetros definidos no *pipeline*. Depois, os dados são processados, para ficarem no formato pretendido pela aplicação de destino: podem ser transformados para o formato JSON, de modo a serem enviados para o ElasticSearch, ou para uma *query* SQL, para serem enviados para uma base de dados relacional. Por último, são enviados para um ou mais destinos, utilizando os *plugins* para integração com as tecnologias de destino.

A utilização desta ferramenta permite que a solução desenvolvida seja utilizada para diversos fins, pois permite recolher dados de fontes como os canais de Mirth utilizados na plataforma AIDA para a troca de mensagens HL7 - respondendo assim à legislação presente no RGPD - ou então dados relativos ao funcionamento dos servidores da plataforma, contribuindo para uma melhor monitorização do funcionamento desta.

5 https://www.elastic.co/logstash

3.2.5 *Ferramentas de Monitorização*

A monitorização aplicacional é uma área em constante crescimento. Hoje em dia é cada vez mais necessário contabilizar e estudar diversos aspetos relacionados com os serviços informáticos das mais diversas áreas, pois as falhas em qualquer sistema informático saem cada vez mais caras aos fornecedores dos serviços.

Kibana

Kibana é um *plugin* de ElasticSearch, que serve como camada de visualização para os dados armazenados nos índices desta base de dados, através de *dashboards* de vários tipos, permitindo também efetuar pesquisas e filtragens sobre os dados, utilizando a API do ElasticSearch para comunicar com o servidor.

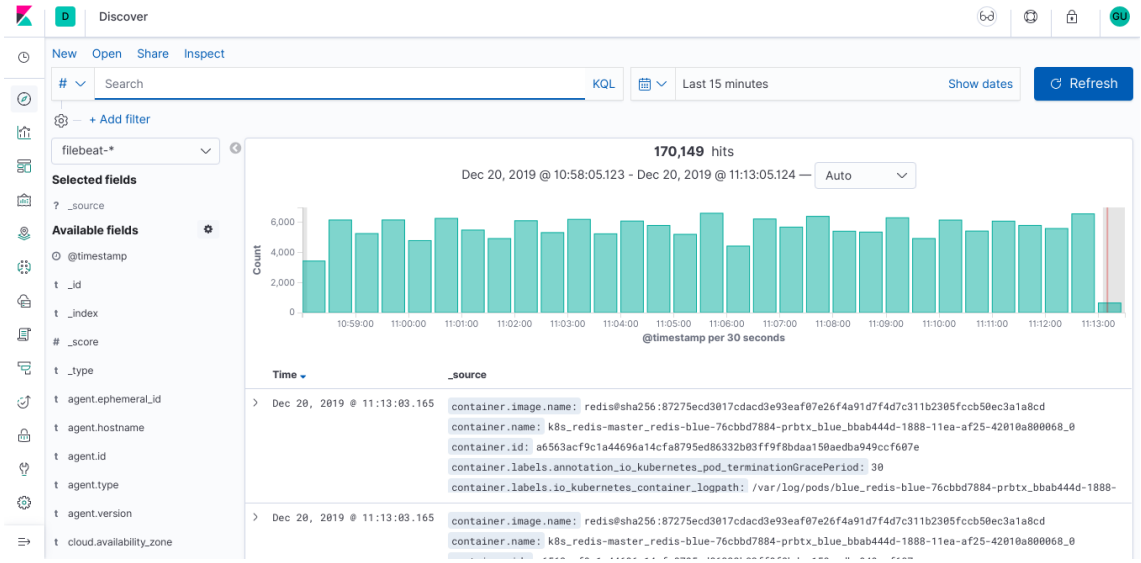


Figura 27: Página de pesquisa do Kibana

A página ilustrada na Figura27 permite diversas operações. Nesta página é possível listar os documentos armazenados num determinado índice, podendo incluir filtros por parâmetros. É também possível observar a quantidade total de documentos armazenados e a quantidade ao longo do tempo, sendo esse intervalo de tempo ajustável. É possível também guardar a listagem de documentos com filtros, para ser utilizada em outras páginas, para uma mais fácil configuração.

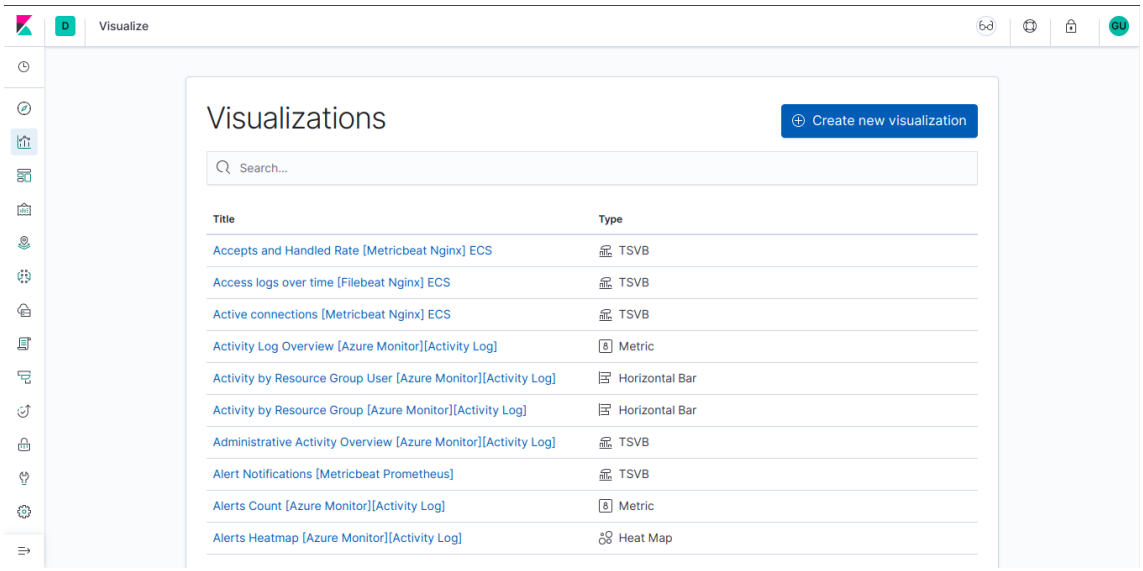


Figura 28: Página de visualizações do Kibana

Esta página permite criar e editar visualizações sobre os dados. É possível a criação de visualizações de diversos tipos, entre os quais métricas sobre um determinado parâmetro (por exemplo, a quantidade de mensagens relativas a um paciente armazenadas) ou a variação de um determinado parâmetro ao longo do tempo. É possível importar listagens guardadas na página de pesquisa.

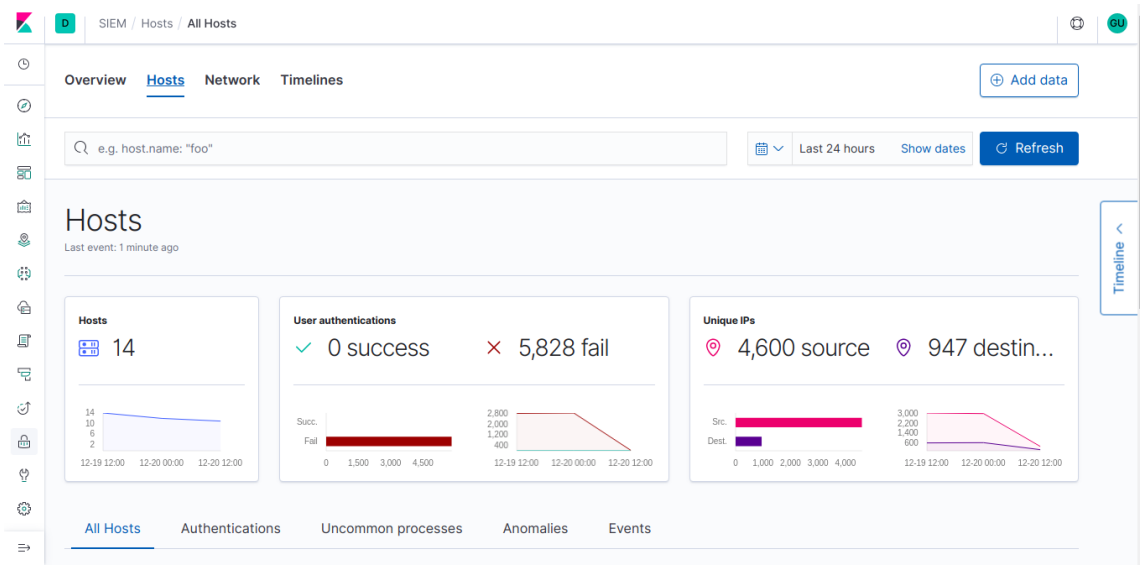


Figura 29: Página dos *dashboards* do Kibana

Esta página permite a criação, edição e visualização de *dashboards*, construídos utilizando os gráficos e contagens criados na página de visualizações. É também possível a exportação dos *dashboards*, permitindo a sua visualização em páginas *web*.

Grafana

Grafana [6] é uma ferramenta *open-source* que permite a criação, edição e visualização de *dashboards*, através de métricas. Possui diversos tipos de visualizações, tais como campos de textos, gráficos de barras ou de linhas, tabelas, entre outros, permitindo desta forma uma grande personalização da monitorização. Tem integração com múltiplas bases de dados, permitindo a utilização de várias para a monitorização. Possui também um sistema de autenticação e de gestão de permissões, com utilizadores e *roles*.

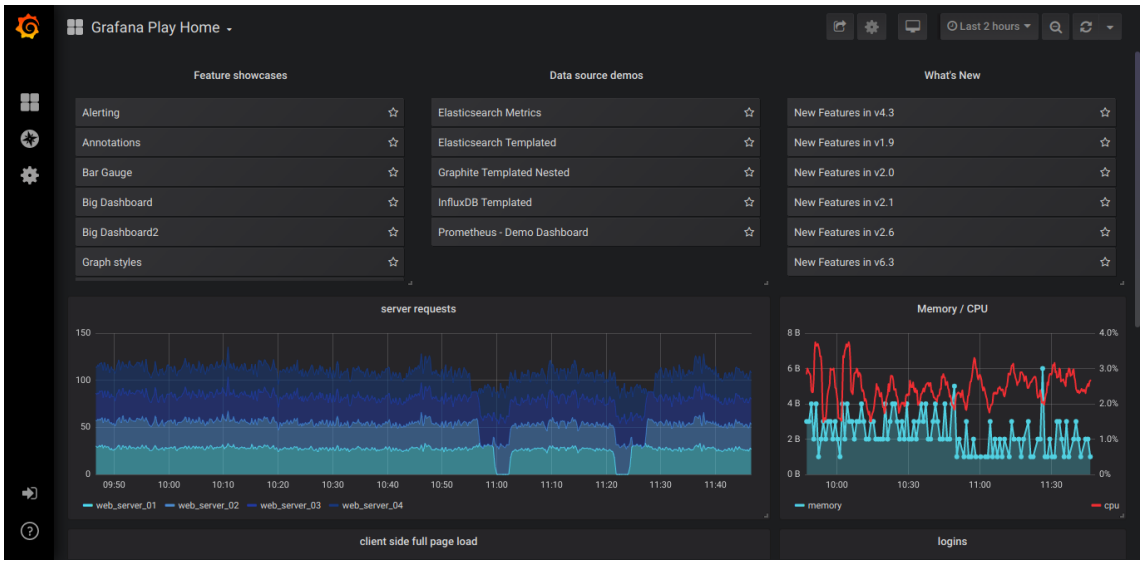


Figura 30: Página dos *dashboards* do Grafana

Nesta página é possível a criação, edição e visualização de *dashboards*, sendo possível definir um intervalo de tempo para a visualização. É possível a partilha dos *dashboards*, através de capturas de ecrã geradas pelo Grafana, com data de expiração ajustável, dando uma visualização dos gráficos presente nos dashboards num determinado intervalo de tempo, acessível por qualquer pessoa com acesso ao *link*. Esta funcionalidade permite a partilha dos *dashboards* sem incluir informação privada. É também possível partilhar os dashboards através de um URl, que permite o acesso a toda a informação do dashboard.

6 https://grafana.com/



Figura 31: Página das *queries* de uma visualização

Nesta página é possível a realização de queries sobre as métricas armazenadas, cujos resultados são exibidos em tempo real nos *dashboards*. É possível também efetuar configurações sobre as visualizações, como a legenda que é apresentada ou a denominação dos dados.

Análise Comparativa

Tabela 2: Análise comparativa entre Grafana e Kibana

	Kibana (open-source)	Grafana
Autenticação e permissões	Sim (com o *plugin SearchGuard*)	Sim
Bases de Dados com integração	Apenas ElasticSearch	Prometheus, ElasticSearch, InfluxDB, Graphite, Postgres, MySQL
Funcionalidades relevantes	Pesquisa de texto	Incluir múltiplas fontes de dados nas visualizações

Estas duas ferramentas de monitorização têm bastantes semelhanças. Ambas possuem uma grande diversidade de visualizações e permitem a partilha de *dashboards*. Por outro lado, existem algumas diferenças fundamentais. A versão *open-source* do Kibana não possui autenticação nem sistema de permissões por defeito, enquanto que o Grafana possui ambos, permitindo maior segurança. Este problema é facilmente contornável, pois o Kibana possui um plugin que oferece estas funcionalidades - o *SearchGuard*. O Grafana possui integração com diversas Bases de Dados, permitindo a utilização de múltiplas na monitorização, enquanto que o Kibana é um *plugin* de ElasticSearch, pelo que não possui integração com

outros sistemas. Desta forma, o Grafana é uma ferramenta orientada a métricas, tirando partido das múltiplas integrações que tem para obter uma maior quantidade de dados. O Kibana, por seu lado, é orientada a registos textuais, devido a ser um *plugin* de ElasticSearch. Como a monitorização incidirá sobre registos textuais, a escolha recaiu sobre Kibana.

3.3 METODOLOGIA DE AVALIAÇÃO

Aqui serão expostas as metodologias que irão ser utilizadas para avaliar a solução desenvolvida, de forma a perceber se esta resolve os problemas apontados no capítulo introdutório (1).

3.3.1 *Avaliação de Performance*

Para avaliar a ferramenta de monitorização foram definidos parâmetros concretos de ações que esta deverá conseguir realizar, tanto no ambiente de desenvolvimento - diga-se, a simulação local - como no ambiente de produção - a solução implementada na unidade de saúde.

Ambiente de Desenvolvimento

No ambiente de desenvolvimento pretende-se avaliar se a solução desenhada pode monitorizar com eficácia a troca de informação na plataforma AIDA e os componentes desta, pelo que foram definidos os seguintes parâmetros a serem avaliados:

- A simulação deverá conseguir armazenar e monitorizar mensagens com parâmetros nulos - desta forma saber-se-á que, mesmo que hajam mensagens com parâmetros nulos a serem enviadas para a ferramenta de monitorização, esta consegue armazenar e monitorizar essas mensagens;
- A simulação deverá apresentar as mensagens com um atraso inferior a 30 segundos - desta forma é possível perceber se a informação visualizada nos *dashboards* é exata;
- A simulação deverá armazenar mensagens relativas ao funcionamento da base de dados simulada;

Ambiente de Produção

No ambiente de produção pretende-se avaliar o desempenho da solução quando colocada perante desafios que não tenham sido simulados no ambiente de desenvolvimento, tais como a atualização das mensagens que têm parâmetros nulos ou um pico na quantidade de informação trocada, bem como perceber se a solução consegue monitorizar o funcionamento

de alguns componentes da plataforma AIDA. Para esse efeito, foram definidos os seguintes parâmetros a serem avaliados:

- A solução deverá conseguir armazenar 90% de toda a troca de informação realizada durante 5 dias;
- A solução deverá conseguir atualizar as mensagens que forem armazenadas com parâmetros nulos;
- A solução deverá conseguir recolher métricas da máquina onde está alojada a ferramenta de monitorização;

3.3.2 *Análise SWOT*

A análise *Strengths, Weaknesses, Opportunities & Threats (SWOT)* é um método utilizado para avaliar as valências (Strengths), fraquezas (Weaknesses), oportunidades (Opportunities) e ameaças (Threats) associadas a um projeto. Envolve especificar o objetivo de um projeto e identificar os fatores internos e externos que são favoráveis e desfavoráveis para alcançar o objetivo [39]. Para identificar cada um desses aspetos é necessário responder a algumas perguntas:

- Valências
  - O que é que o sistema faz bem?
  - Que propriedades diferenciadas possui?
- Fraquezas
  - O que pode ser melhorado?
  - Onde existem menos recursos que em outros projetos?
- Oportunidades
  - Que oportunidades existem para o sistema?
  - De que tendências o sistema pode tirar partido?
  - Que valências do sistema podem ser tornadas oportunidades?
- Ameaças
  - Que ameaças podem marcar negativamente a solução?
  - Que ameaças podem ser expostas pelas fraquezas?

Ao utilizar este método para avaliar a solução desenvolvida é possível compreender que aspetos desta podem ser melhorados, criando novos problemas para serem resolvidos e novas oportunidades para serem exploradas em projetos futuros.

4

PLATAFORMA DE MONITORIZAÇÃO

Este capítulo do presente documento expõe o problema que levou ao desenvolvimento desta solução, os requisitos funcionais e técnicos a que esta terá de obedecer e a arquitetura tanto da simulação local como da solução implementada em ambiente real.

4.1 IDENTIFICAÇÃO DO PROBLEMA

Hoje em dia, os sistemas informáticos são, cada vez mais, parte integrante dos serviços das mais variadas áreas. Desde bancos a agências de seguros, hospitais ou clínicas, todos estes serviços possuem sistemas informáticos dos quais dependem para realizarem a sua atividade. Esses sistemas informáticos contêm no seu seio dados pessoais de todas as pessoas que deles usufruem. Por exemplo, diariamente circulam na plataforma AIDA milhares de mensagens HL7 que contêm dados pessoais dos intervenientes do universo das unidades de saúde nas quais esta plataforma atua.

O surgimento de legislação na área da proteção de dados pessoais, que emergiu dessa crescente utilização de sistemas informáticos no nosso dia-a-dia, trouxe a necessidade de esses sistemas alterarem a sua forma de atuar de modo a poderem cumpri-la. É agora uma prioridade das plataformas das mais diversas áreas a criação de mecanismos para que possam atuar em conformidade com essa legislação. Na área da saúde, é necessária a existência nos sistemas informáticos de um registo das operações de tratamento de dados pessoais realizadas no seu seio. A plataforma AIDA não possui mecanismos nesse sentido, pelo que ainda não se encontra a atuar em conformidade com essa legislação.

Sendo os sistemas informáticos uma parte integrante dos serviços indispensáveis ao correto funcionamento de uma sociedade, é importante que estes estejam disponíveis para utilização o máximo de tempo possível, de modo a diminuir a ocorrência de problemas que possam advir da sua indisponibilidade. Quando se trata de sistemas informáticos na área da saúde, é importante que estes estejam sempre disponíveis de modo a que não exista perda de informação dos pacientes, pois isso iria implicar a perda de informação clínica que pode ser essencial para se compreender o problema de saúde que os afeta.

A plataforma AIDA não possui uma ferramenta para monitorizar o estado dos seus componentes, pelo que é também essencial o desenvolvimento de ferramentas das quais esta plataforma pode beneficiar para analisar o estado dos seus componentes, podendo utilizar as métricas relativas aos seus componentes para antecipar possíveis falhas no seu funcionamento, contribuindo desta forma para um melhor funcionamento.

4.2 REQUISITOS FUNCIONAIS

Os requisitos funcionais dizem respeito às funcionalidades que a ferramenta terá de cumprir e que descrevem um seu componente ou um comportamento que este deverá conseguir realizar. Esse comportamento pode ser um serviço, uma função ou tarefa. No desenvolvimento de um produto, é útil distinguir entre funcionalidades necessárias para o produto a desenvolver competir com potenciais concorrentes e funcionalidades que diferenciam o sistema desses concorrentes [40].

Neste sentido, foram definidos os seguintes requisitos funcionais:

- A solução deverá incluir um sistema de autenticação, tanto na base de dados como na ferramenta de visualização;
- A solução deverá possuir uma página de pesquisa;
- A solução deverá permitir a construção, edição e eliminação de *dashboards*;
- A solução deverá possuir uma página de visualização de *dashboards*;
- A solução deverá permitir a partilha de *dashboards*, através de iFrames;
- A solução deverá permitir a filtragem de dados;
- A solução deverá permitir a importação e exportação de *dashboards*, no formato JSON;
- O utilizador deverá conseguir fazer pesquisas sobre a informação armazenada de forma eficiente;

4.3 REQUISITOS TÉCNICOS

Os requisitos técnicos são restrições técnicas que o sistema deve cumprir, estando relacionados com o desempenho da solução, a usabilidade ou a fiabilidade deste. Ao contrário dos requisitos funcionais, aplicam-se ao sistema como um todo e não apenas a uma funcionalidade. No desenvolvimento de um projeto este tipo de requisitos são essenciais, pois balizam os objetivos propostos para a solução e ajudam a decidir a arquitetura do sistema. Nesse sentido, foram definidos os seguintes requisitos técnicos:

- O utilizador deverá conseguir visualizar a informação relativa às operações de tratamento de dados efetuadas na plataforma AIDA;
- O utilizador deverá conseguir construir *dashboards* após utilizar a plataforma durante 10 minutos;
- O utilizador deverá conseguir monitorizar os componentes da plataforma AIDA;
- O utilizador deverá autenticar-se com um *login* antes de utilizar o sistema;
- Apenas o administrador deverá conseguir aceder à informação dos índices da base de dados;
- Apenas o administrador deverá adicionar, editar e eliminar índices;

4.4 Arquitetura da Solução

Tendo em conta os requisitos funcionais e técnicos expostos anteriormente, é proposta uma solução que recorre às tecnologias estudadas no capítulo 3 - ElasticSearch, Kibana, Logstash, MirthConnect e Docker - para dar resposta aos problemas expostos no capítulo introdutório. Primeiramente, foi proposta a arquitetura que se encontra ilustrada na Figura32.

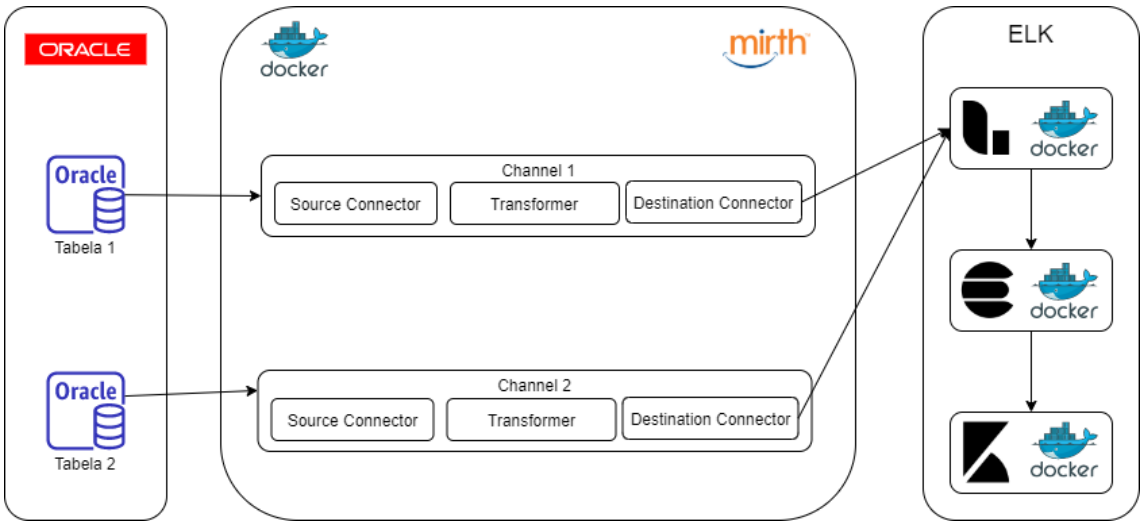


Figura 32: Primeira proposta da arquitetura da solução

Inicialmente, a solução consistia em *channels* MirthConnect, para extrair a cada minuto e transformar as mensagens armazenadas na base de dados Oracle da plataforma AIDA, enviando-as para o Logstash, que as filtra consoante a tabela de origem e envia para o índice ElasticSearch correspondente, sendo as mensagens mostradas no Kibana. As tecnologias são encapsuladas em *containers* Docker, para que seja mais fácil a sua migração.

Posteriormente foi proposta uma nova solução, para possibilitar a integração da monitorização dos componentes da plataforma, tornando ao mesmo tempo o Logstash a ferramenta de recolha das mensagens HL7. Foi adicionado um *plugin* de agentes de recolha e envio de dados, o Beats [1], para possibilitar a monitorização dos componentes da plataforma AIDA. A arquitetura encontra-se exposta na Figura33

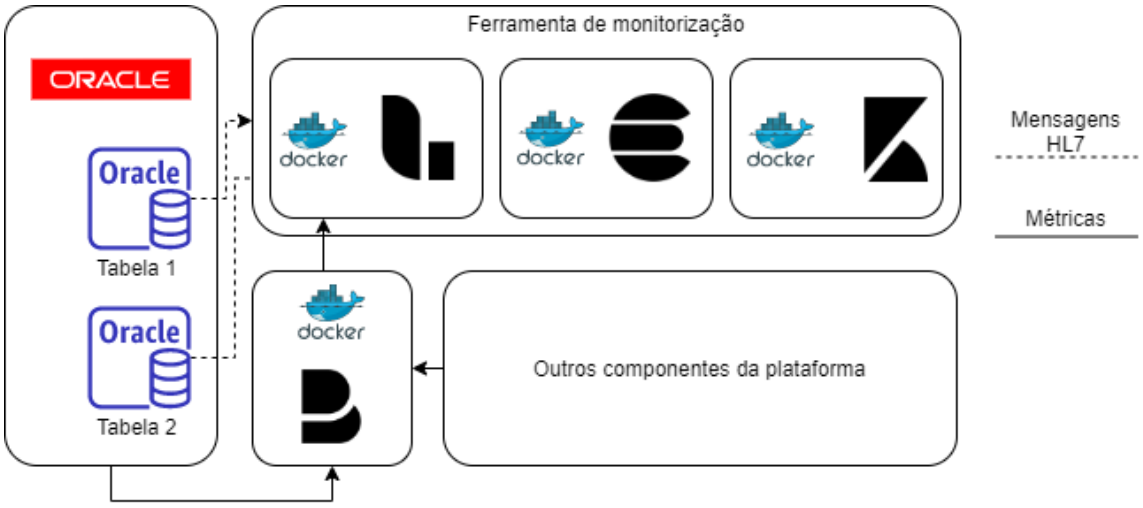


Figura 33: Arquitetura da solução

Nesta arquitetura as mensagens HL7 são recolhidas pelo Logstash, através de uma *query* à base de dados, sendo postetiormente enviadas para o ElasticSearch. O Beats recolhe métricas sobre o funcionamento dos diversos componentes da plataforma AIDA e envia-as para o Logstash, para que sejam processadas e armazenadas no ElasticSearch. Esta arquitetura permite que a recolha e processamento de mensagens e métricas sejam uniformizados, ficando ao cargo de apenas uma tecnologia, contrariando a arquitetura proposta inicialmente, em que as mensagens eram processadas no MirthConnect e depois processadas novamente no Logstash.

4.4.1 *Ambiente de Desenvolvimento*

Numa primeira fase da implementação foram realizados diversos testes às ferramentas da solução. O primeiro teste consistiu numa simulação local do ambiente no qual iria ser integrada a solução proposta e na construção de alguns dashboards relativos à informação simulada, utilizando uma base de dados - PostgreSQL [2] - com duas tabelas, um script feito em NodeJS [3] para gerar mensagens HL7, gerando uma mensagem a cada 5 segundos, o

1 https://www.elastic.co/pt/beats/
2 https://www.postgresql.org/
3 https://nodejs.org/

engine MirthConnect e as ferramentas do ELK, todas encapsuladas em *containers* Docker. A arquitetura da simulação encontra-se exposta na Figura34.

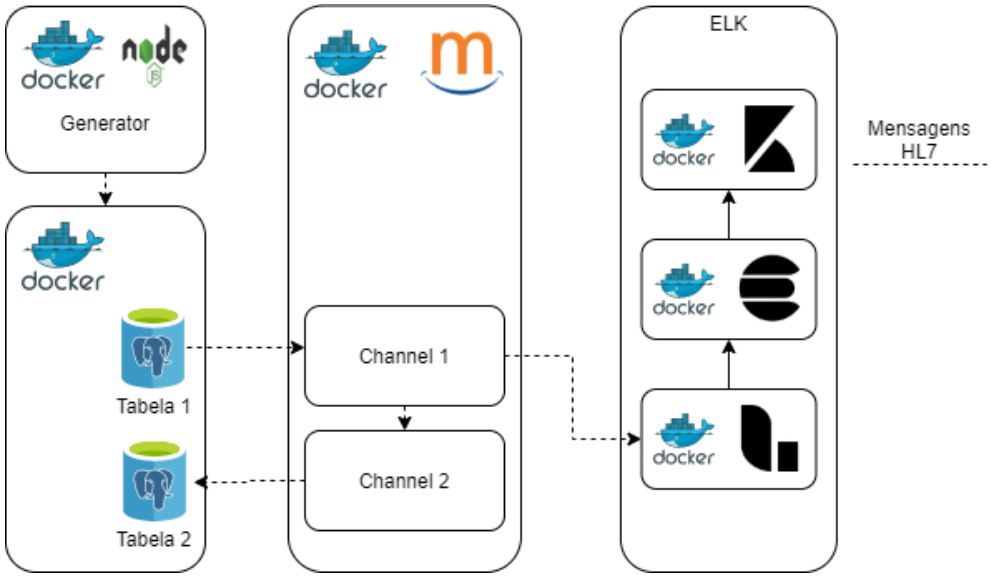


Figura 34: Arquitetura da primeira solução local

Como é possível observar na Figura34, as mensagens geradas pelo *script* são armazenadas na base de dados PostgreSQL. O *channel* 1 do MirthConnect efetua uma *query* periódica (de minuto a minuto) à tabela 1 da base de dados PostgreSQL, recolhendo as mensagens armazenadas nesse minuto. Posteriormente processa-as para ficarem no formato JSON e envia-as para o Logstash, para serem armazenadas e visualizadas na ferramenta de monitorização, e também para o *channel* 2, para serem armazenadas na tabela 2 da base de dados. Na base de dados Postgres foi simulada a estrutura da base de dados do Centro Hospitalar do Tâmega e do Sousa, a unidade de saúde na qual a solução iria ser implementada.

De modo a simular o ambiente no qual a solução iria atuar, foram criados *dashboards* relativos às operações de troca de dados. Esses *dashboards* encontram-se ilustrados nas figuras 35, 36 e 37.

O *dashboard* da Figura35 dá uma vista geral sobre as mensagens HL7 trocadas num dado período de tempo. Na área "contagens"são mostrados os 5 mais relevantes tipos de mensagens trocadas no dado período de tempo, bem como o número de mensagens desse tipo trocadas. Na área "nulos"são apresentadas as mensagens que são trocadas sem tipo. Por baixo é dada uma lista com todas as mensagens trocadas no espaço de tempo dado, com todos os parâmetros de informação que essas mensagens trazem. É também possível realizar uma filtragem sobre qualquer desses parâmetros.

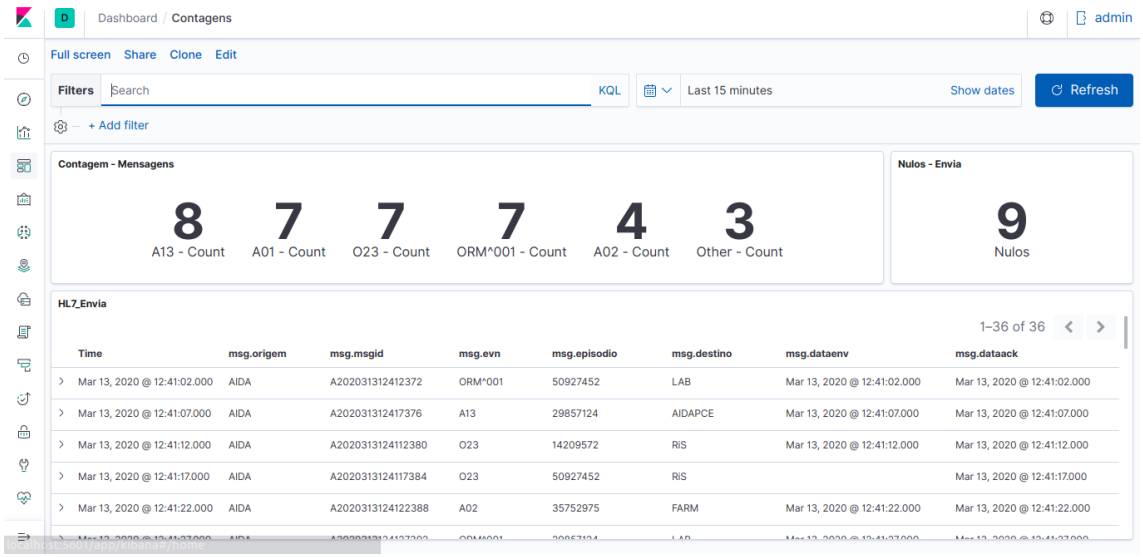


Figura 35: *Dashboard* da contagem e listagem de mensagens

A utilização deste *dashboard* permite monitorizar se os eventos mais utilizados no funcionamento da plataforma AIDA são os habituais e se não existem mais mensagens com campos nulos do que o razoável, ajudando a que possíveis falhas relacionadas com esses aspetos sejam corrigidas.

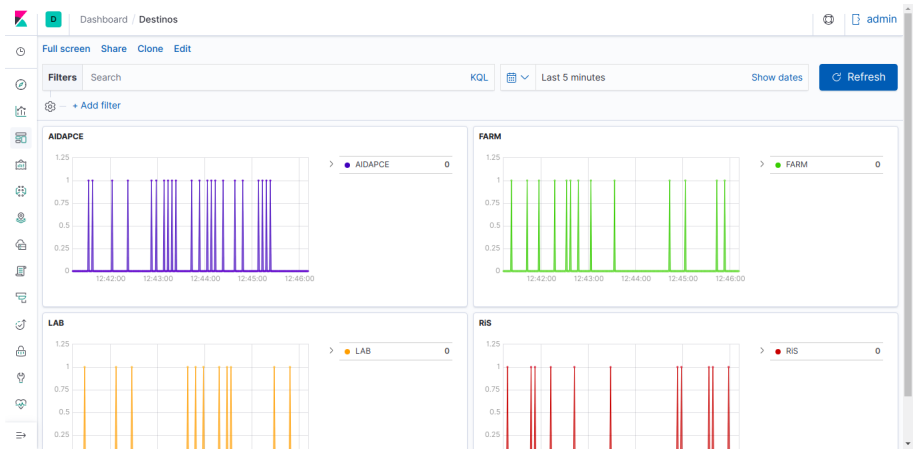


Figura 36: *Dashboard* dos destinos das mensagens

O *dashboard* da Figura36 apresenta o número de mensagens enviadas para cada destino num determinado espaço de tempo. Desta forma é possível saber o tráfego de cada departamento e encontrar eventuais anomalias no funcionamento do sistema, tais como o não envio ou receção de mensagens ou uma sobrecarga do departamento.

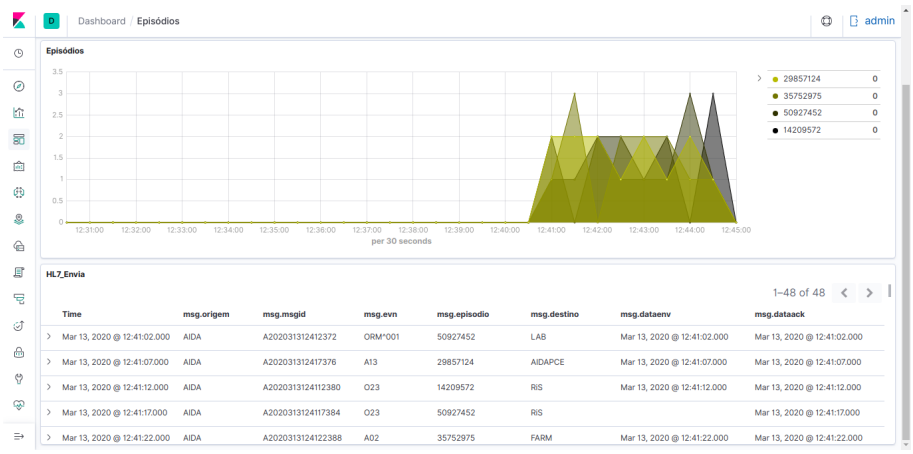


Figura 37: *Dashboard* dos episódios

O *dashboard* da Figura37 apresenta um gráfico com o número de mensagens relativas a um determinado episódio trocadas nos últimos 15 minutos. Por baixo, apresenta a lista de mensagens trocadas. Através deste dashboard é possível monitorizar a informação sobre um episódio específico trocada na plataforma, desta forma contribuindo para que a plataforma AIDA opere em conformidade com o RGPD.

As mensagens utilizadas nesta simulação têm alguns dos parâmetros iguais às mensagens trocadas diariamente na unidade de saúde na qual a solução foi implementada, na fase final do projeto, o Centro Hospitalar do Tâmega e do Sousa. São constituídas por sete parâmetros, simulados no *script* NodeJS:

- msgid - O ID único da mensagem, gerado a partir da data em que a mensagem é enviada;
- origem - A origem da mensagem. Nesta simulação as mensagens geradas têm sempre a mesma origem;
- evn - O evento HL7 tratado pela mensagem. Depende do destino da mensagem, pois alguns eventos são específicos de determinados destinos;
- destino - O destino da mensagem. Pode ser um de quatro destinos: RiS (), LAB(), AIDAPCE() e FARM(); Mensagens que tenham como destino o RiS têm como evento o evento O23 (). Mensagens cujo destino seja o LAB têm como evento o ORMˆ001 (). Mensagens cujo destino seja o AIDAPCE ou o FARM podem corresponder a 1 de 4 eventos: A01 (), A02 (), A03 () e A13 ();
- dataenv - A data e hora em que a mensagem foi enviada;
- dataack - A data e hora em que é recebida a resposta ao envio da mensagem - Pode ser nulo;

- episodio - O episódio a que corresponde a mensagem.

Após a primeira simulação foi efetuada uma segunda. Nesta, as mensagens são geradas por 2 *scripts* NodeJS (um para cada tabela) e dizem respeito às duas tabelas que viriam a ser monitorizadas em ambiente real. A arquitetura da simulação está ilustrada na Figura38.

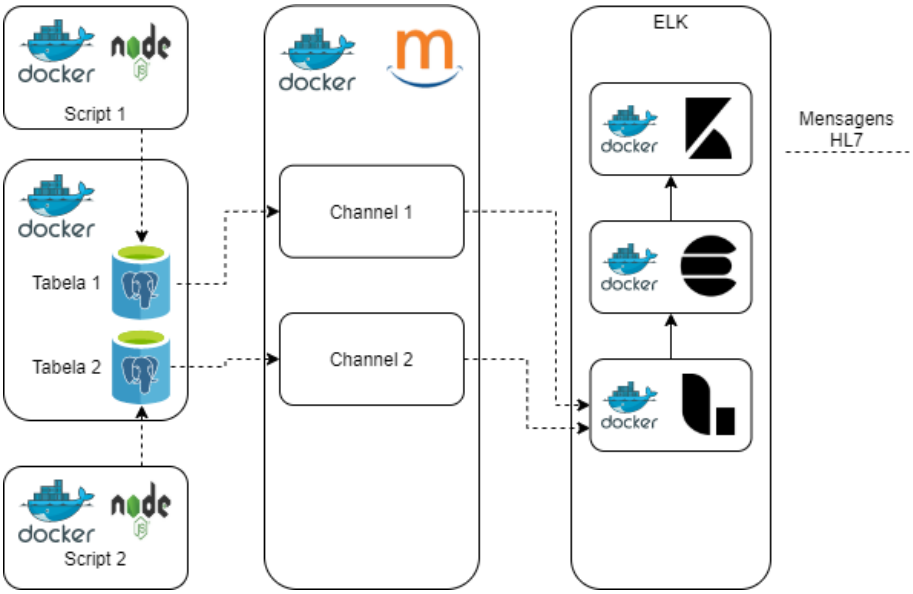


Figura 38: Arquitetura da segunda simulação local

Como ilustrado na Figura38, as mensagens são geradas pelos *scripts*, tal como na primeira simulação, e são armazenadas na base de dados PostgreSQL, na tabela respetiva. O *channel* 1 do MirthConnect efetua uma *query* periódica (de minuto a minuto) à tabela 1 da base de dados PostgreSQL, recolhendo as mensagens armazenadas nesse minuto, enquanto que o *channel* 2 utiliza o mesmo processo para a tabela 2. Posteriormente, as mensagens são processadas para ficarem no formato JSON e enviadas para o Logstash, para serem armazenadas e visualizadas na ferramenta de monitorização. Nesta simulação foram utilizados os mesmos campos que na simulação anterior, acrescentando um campo, para armazenar a data de receção da mensagem, um campo específico da segunda tabela.

Por último, foi efetuada uma última simulação, de modo a testar a viabilidade da monitorização dos componentes da plataforma AIDA. Isto foi motivado pela vontade do grupo de investigação de tornar a ferramenta de monitorização de dados uma ferramenta de monitorização de toda a plataforma AIDA, aproveitando as valências das tecnologias que fazem parte da ferramenta. Sendo ElasticSearch uma base de dados orientada a documentos, é possível armazenar nesta os vários tipos de dados necessários para monitorizar a plataforma, tais como métricas de sistema ou *logs* dos vários componentes, sem comprometer a *performance* desta na pesquisa de documentos. Foram utilizadas as mesmas tecnologias que na segunda implementação e adicionado o *Beats*, um *plugin* de agentes de recolha e envio de

dados, para recolher métricas relativas ao *container* Docker da base de dados PostgreSQL. A arquitetura da simulação encontra-se ilustrada na Figura39.

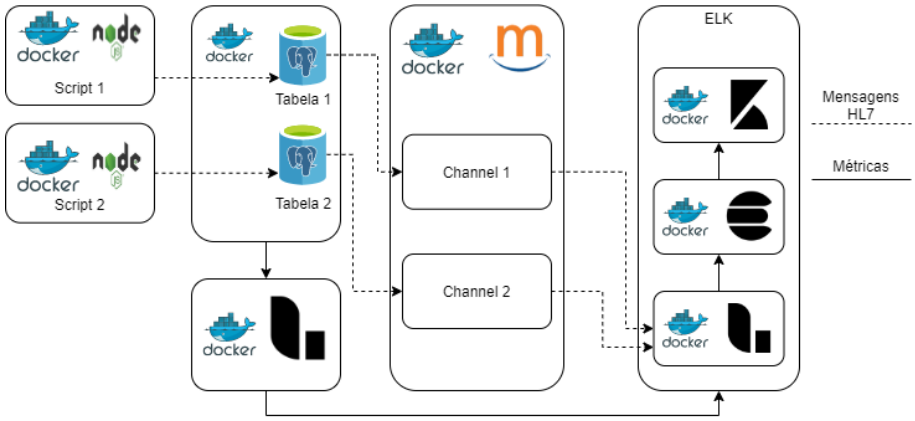


Figura 39: Arquitetura da terceira simulação local

Como ilustrado na Figura39, no geral esta simulação funciona da mesma forma que a solução anteriormente exposta, com uma funcionalidade adicional. O Beats recolhe métricas sobre o funcionamento do *container* docker, enviando-as para o Logstash. As métricas recolhidas são as seguintes:

- Estado do container (*up*, *down*);
- Percentagem de processador utilizada;
- Utilização de memória;
- Tráfego de rede;

4.4.2 *Ambiente de Produção*

Após ter sido feita a primeira simulação local da solução, esta foi implementada em ambiente real, no Centro Hospitalar do Tâmega e do Sousa, numa máquina com as seguintes especificações:

- 100 GB de disco
- 2 Processadores Intel(R) Xeon(R) CPU E5-2640 v3 @ 2.60GHz de 8 *cores*, 16 *threads* e 20MB de cache
- 4 GB de Memória RAM

A solução implementada passou por diversas fases. Primeiramente foi implementada uma solução similar à solução testada inicialmente, com 2 *channels* MirthConnect para a extração

e transformação a cada minuto das mensagens armazenadas na base de dados Oracle e com 2 *channels* adicionais, para atualização das mensagens que tiverem campos como o dataack vazios. A primeira solução implementada encontra-se ilustrada na Figura40.

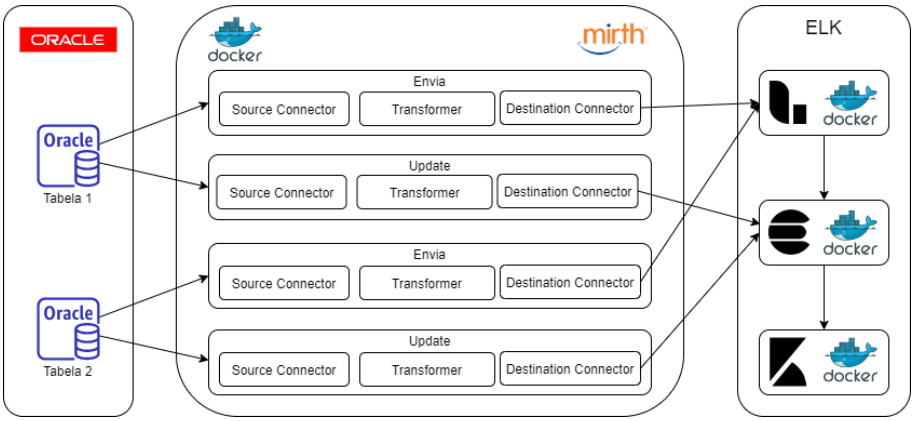


Figura 40: Arquitetura da primeira solução implementada no Centro Hospitalar do Tâmega e do Sousa

A implementação ficou marcada por alguns problemas relativos à mudança da hora, de *Coordinated Universal Time (UTC)* para a hora de verão (*European Summer Time (EST)*), pois essa mudança trouxe uma incongruência nas datas armazenadas na base de dados da unidade de saúde e as datas apresentadas na plataforma de monitorização. As datas armazenadas na base de dados são extraídas na hora de verão e o ElasticSearch interpreta-as como sendo UTC, fazendo com que o Kibana acrescente uma hora às datas, ou seja, os dados aparecem nos dashboard com uma hora de atraso. Para corrigir este problema foi adicionado um passo antes do envio das mensagens para o ElasticSearch. No Mirth, quando a mensagem é transformada para o formato em que é enviada (JSON), as datas são convertidas da hora de verão para a hora UTC, utilizando um *script*, permitindo assim que sejam visualizadas da maneira correta no Kibana. Foi também adicionado um utilizador à plataforma Kibana que possui permissões de leitura sobre os índices de ElasticSearch e sobre os dashboards do Kibana. Foram construídos *dashboards*, ilustrados nas figuras 41, 42 e 43, relativos a diversos aspetos importantes:

- Destinos das mensagens;
- Episódios das mensagens;
- Contagem do número de mensagens cujo dataack é nulo;
- Contagem dos eventos mais utilizados;

O *dashboard* da Figura41 dá uma vista geral sobre as mensagens HL7 trocadas num dado período de tempo. Na área "contagens"são mostrados os 5 mais relevantes tipos de

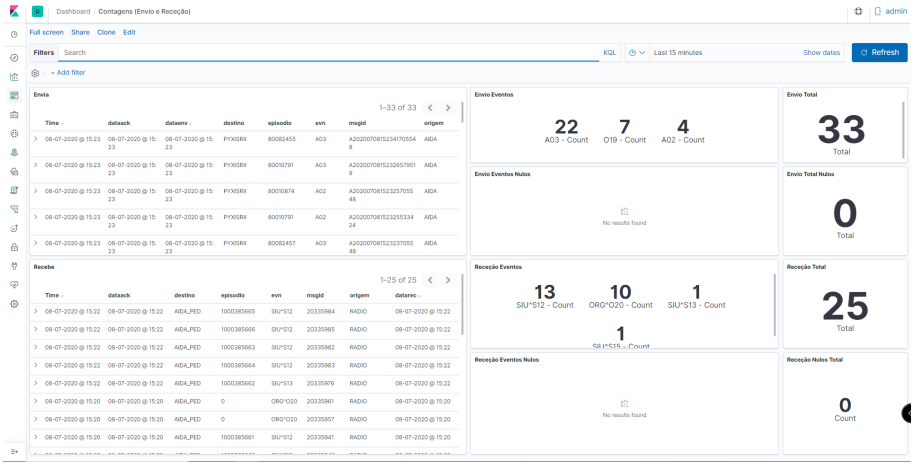


Figura 41: *Dashboard* da contagem e listagem de mensagens

mensagens trocadas no dado período de tempo, bem como o número de mensagens desse tipo trocadas. Na área ”nulos”são apresentadas as mensagens que são trocadas sem tipo. Por baixo é dada uma lista com todas as mensagens trocadas no espaço de tempo dado, com todos os parâmetros de informação que essas mensagens trazem. É também possível realizar uma filtragem sobre qualquer um desses parâmetros.

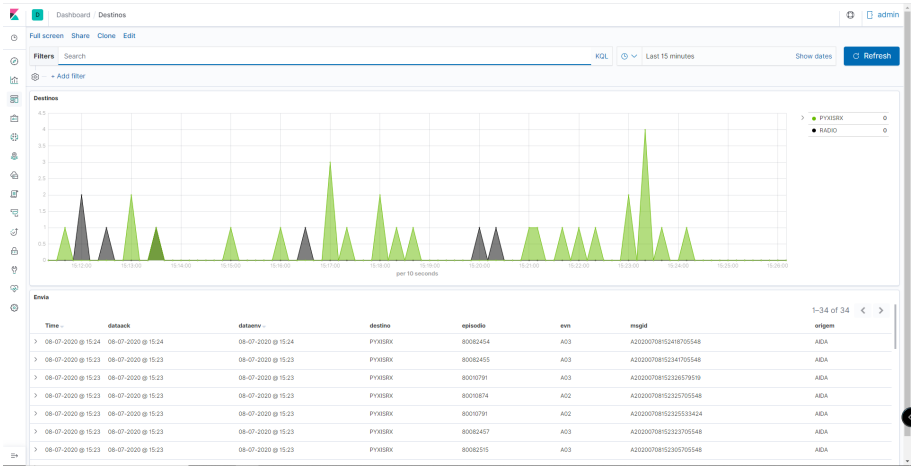


Figura 42: *Dashboard* dos destinos das mensagens

O *dashboard* da Figura42 apresenta o número de mensagens enviadas para cada destino num determinado espaço de tempo. Desta forma é possível saber o tráfego de cada departamento e encontrar eventuais anomalias no funcionamento do sistema, tais como o não envio ou receção de mensagens ou uma sobrecarga do departamento.

O *dashboard* da Figura43 apresenta um gráfico com o número de mensagens relativas a um determinado episódio trocadas nos últimos 15 minutos. Por baixo, apresenta a lista de

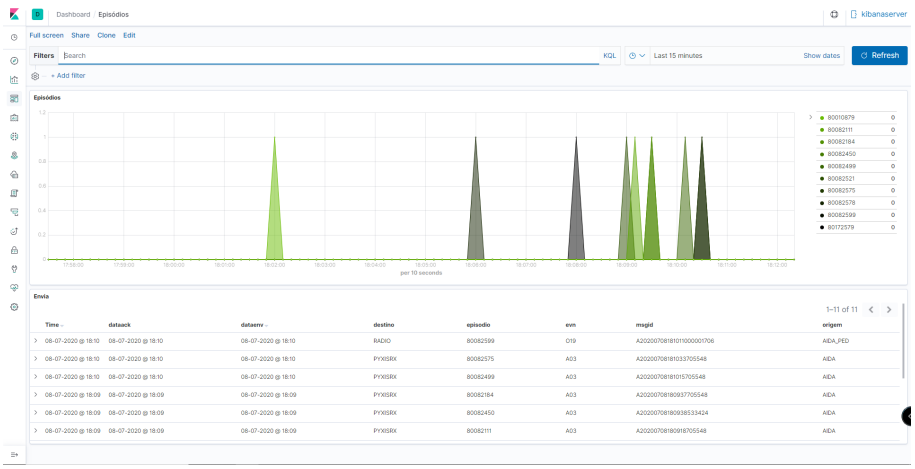


Figura 43: *Dashboard* dos episódios

mensagens trocadas. Através deste *dashboard* é possível monitorizar a informação sobre um episódio específico trocada na plataforma.

Posteriormente, no seguimento dos testes efetuados em ambiente de desenvolvimento, e com a alteração da arquitetura da ferramenta de monitorização, foi mudada a arquitetura da solução implementada no Centro Hospitalar.

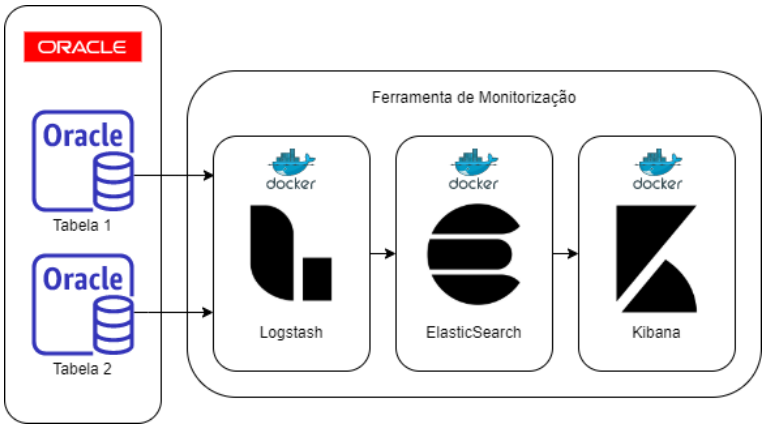


Figura 44: Arquitetura da segunda solução implementada no Centro Hospitalar do Tâmega e do Sousa

Nessa solução, cuja arquitetura está representada na Figura44, as mensagens são recolhidas pelo Logstash, utilizando uma *query* que converte as horas em UTC, que posteriormente, consoante a origem da mensagem, a armazena no índice de ElasticSearch correspondente, num índice diferente por dia. A opção por esta arquitetura constituiu uma uniformização das operações de recolha e processamento de dados, garantindo também a possibilidade de acrescentar integrações com outros componentes da plataforma, de modo a monitorizar o funcionamento desta, retirando, no percurso, complexidade e dependências ao projeto.

Após ter sido configurada a solução final de monitorização foram também realizados testes para uma possível integração do Logstash com os componentes da plataforma, para averiguar a possibilidade de acrescentar mais funcionalidades de monitorização do funcionamento da mesma, em ambiente de desenvolvimento.

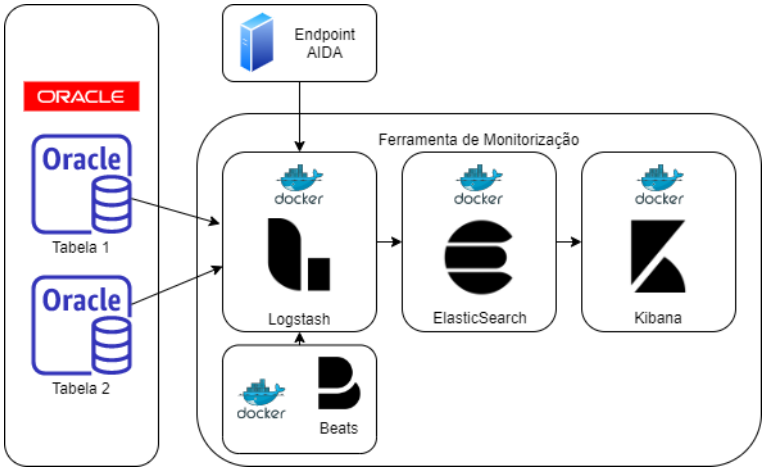


Figura 45: Arquitetura da solução final de monitorização

Como ilustrado na Figura45, foi integrada com o Logstash uma ferramenta de recolha de dados, o Beats [4], que, através dos seus plugins, recolhe métricas de sistema, dados de rede, monitorização de disponibilidade ou *logs*. Foram também criados *pipelines* adicionais no Logstash, quer para recolher as métricas enviadas pelo Beats, quer para fazer os *pings* ao endpoint da plataforma AIDA. Foram monitorizados a máquina virtual onde está instalada a ferramenta de monitorização, através de métricas de sistema, e um endpoint da plataforma, através de *pings* de minuto a minuto.

4 https://www.elastic.co/pt/beats/

5

RESULTADOS, DISCUSSÃO E AVALIAÇÃO

Neste capítulo será avaliada a solução desenvolvida, utilizando os métodos mencionados no capítulo 3, sendo discutidos os resultados obtidos. Serão avaliados os testes realizados localmente e as implementações em ambiente real.

5.1 AVALIAÇÃO DE PERFORMANCE

Foram realizadas três simulações em ambiente de desenvolvimento, com objetivos diferentes: Nas primeiras duas simulações o objetivo foi avaliar se as mensagens com parâmetros nulos eram efetivamente armazenadas e se não existia atraso superior a 30 segundos na visualização das mensagens. Na terceira simulação, para além de se verificar se os parâmetros anteriores eram cumpridos, foi também intenção verificar se a solução conseguia armazenar com sucesso as mensagens relativas ao funcionamento da base de dados PostgreSQL.

Em ambiente de produção, era necessário avaliar o desempenho da solução quando colocada perante os desafios diários que no ambiente de desenvolvimento não foram simulados. Nas duas primeiras implementações, o objetivo foi perceber se a solução conseguiria armazenar 90% de toda a troca de informação durante 5 dias e se as mensagens com parâmetros nulos eram atualizadas. Na última implementação, para além de cumprir os mesmos objetivos que as implementações anteriores, foi também verificado se a solução conseguia monitorizar o funcionamento da máquina onde esta foi implementada. A comparação entre o que se pretendia verificar e o que foi obtido tanto em ambiente de desenvolvimento como em ambiente de produção está descrito nas Figura46.

Objetivos	Simulação 1	Simulação 2	Simulação 3
Armazenar e monitorizar mensagens com parâmetros nulos	Foram geradas de modo aleatório algumas mensagens com parâmetros nulos para testar o cumprimento deste parâmetro. Como é possível observar na figura 41, as mensagens com parâmetros nulos não só encontravam-se a ser armazenadas como a ser monitorizados.		
Apresentar as mensagens com atraso inferior a 30 segundos	O MirthConnect efetuava a query à base de dados de 5 em 5 segundos. O Kibana atualizava os dashboards de 15 em 15 segundos e as mensagens dos últimos 30 segundos apareciam sempre.	O Logstash efetuava a query à base de dados todos os segundos e o Kibana atualizava os dashboards com o mesmo espaçamento temporal que no teste anterior.	
Armazenar mensagens sobre o funcionamento do container Docker da base de dados			Na terceira simulação foram armazenadas mensagens sobre os aspetos do funcionamento do container Docker da base de dados descritos no capítulo anterior. Apesar disso, não foi possível testar esta funcionalidade de forma alargada devido ao reduzido poder computacional do dispositivo disponível, o que provocava bloqueios frequentes quando era monitorizado o container.

Objetivos	Implementação 1	Implementação 2	Implementação 3
Armazenar 90% de toda a troca de informação realizada durante 5 dias	A solução monitoriza de forma eficaz a troca de informação na plataforma, tendo funcionado ininterruptamente durante 1 mês.		
Atualizar as mensagens com parâmetros nulos	Os 2 channels de MirthConnect criados para atualizar os parâmetros nulos efetuavam queries às tabelas da base de dados todos os minutos. O Kibana atualizava os dashboards de 15 em 15 segundos e as mensagens eram atualizadas.	O Logstash escutava por atualizações na base de dados e atualizava os documentos no ElasticSearch sempre que existia alguma alteração. Por esta razão, a opção por utilizar apenas o Logstash para recolher os dados das tabelas da base de dados acabou por se revelar muito positiva, pois retirou alguma complexidade à ferramenta.	
Recolher métricas sobre o funcionamento da máquina onde a ferramenta está alojada			As métricas recolhidas sobre o funcionamento da máquina onde a ferramenta se encontrava alojada estão descritos no capítulo anterior. Foram armazenados com sucesso vários indicadores sobre o desempenho da máquina, tendo sido recolhidos durante uma semana ininterruptamente.

Figura 46: Comparação entre os parâmetros de avaliação propostos no capítulo 3 e os resultados obtidos

Os resultados retirados do desenvolvimento da solução foram bastante positivos. Tanto em ambiente de desenvolvimento como em ambiente de produção, estes permitiram cumprir os objetivos delineados. Após os testes no ambiente de desenvolvimento foi possível perceber que a monitorização dos componentes da plataforma podia ser realizada pela solução desenvolvida, pela facilidade com que foi integrada nas simulações.

Em ambiente de produção foi possível perceber que a solução cumpria as exigências do ambiente em que estava a atuar e que eram viáveis os mecanismos de monitorização. Os resultados obtidos demonstram também a viabilidade da arquitetura da solução, pois esta permite a inclusão de mais mecanismos de monitorização e o armazenamento de grandes quantidades de dados sem necessidade de efetuar grandes alterações, nem ao nível das tecnologias de recolha de métricas nem ao nível da base de dados, pois as operações relacionadas com a criação de índices ou a inserção/atualização de documentos são realizadas por uma API e tanto o Logstash como o Beats permitem a criação de *pipelines* sem necessidade de parar todo o sistema.

5.2 ANÁLISE SWOT

Após a avaliação da *performance* da solução tanto em ambiente de desenvolvimento como de produção, foi realizada uma análise SWOT, que se encontra exposta na Figura47. Através desta análise foi possível compreender melhor os diversos aspetos positivos e negativos aliados à solução desenvolvida, quer internos quer externos à plataforma.

	Fatores Positivos	Fatores Negativos
Fatores Internos	Forças: - Facilidade de integração de mecanismos de monitorização; - Importação e exportação de dashboards; - Pesquisa e filtragem de mensagens; - Garante a conformidade com o RGPD;	Fraquezas: - Ausência de funcionalidades para criar workspaces no Kibana (apenas disponível na versão paga); - Ausência de uma base de dados de backup para prevenir eventuais falhas;
Fatores Externos	Oportunidades: - O armazenamento de grandes quantidades de dados permite a utilização de algoritmos de mineração de dados para encontrar padrões quer no funcionamento dos componentes da plataforma quer na quantidade e no tipo de mensagens trocadas; - Integração de mais mecanismos de monitorização	Ameaças: - Alteração da lei de proteção de dados; - Falhas de comunicação entre tecnologias;

Figura 47: Análise SWOT

Desta análise podem-se retirar resultados positivos, pois a solução possui diversas valências que apetrecham a plataforma AIDA de novos componentes essenciais para garantir

o cumprimento do RGPD e para monitorizar de maneira mais eficaz o funcionamento dos seus componentes. A facilidade de inclusão de novos mecanismos de monitorização e a importação e exportação de dashboards tornam a plataforma mais abrangente e mais simples de utilizar. Outro aspeto importante prende-se com a capacidade de pesquisar e filtrar mensagens, que garante a conformidade com o RGPD. No entanto, foram apontadas também algumas fraquezas da solução. A impossibilidade de criar *workspaces* no Kibana constitui um problema para a ferramenta de monitorização, pois existindo *workspaces* diferentes para os *dashboards* relativos à monitorização das operações de trocas de dados e para os *dashboards* para a monitorização dos componentes da plataforma poderiam existir 2 grupos de utilizadores da ferramenta. Outro problema deve-se à ausência de uma base de dados de *backup* dos dados armazenados, não prevenindo assim potenciais perdas de informação no ElasticSearch.

No que diz respeito aos fatores externos, as oportunidades constituem possíveis pontos para serem explorados em trabalhos futuros. A grande quantidade de informação, quer relativa às operações de tratamento de dados quer ao funcionamento dos diversos componentes da plataforma AIDA pode ser utilizada para alimentar algoritmos de mineração de dados, de modo a descobrir potenciais padrões ou falhas no funcionamento da plataforma (dos seus componentes e da quantidade e tipo de mensagens trocadas no seu seio). A introdução de mais mecanismos de monitorização pode ajudar também nesse aspeto. No entanto, existem ameaças externas ao correto funcionamento da ferramenta de monitorização. Uma alteração da lei relativa à proteção de dados em vigor pode fazer com que tenha de ser alterada a forma de a ferramenta de monitorização operar. Aliada a uma das fraquezas detetadas, falhas de comunicação entre as tecnologias podem provocar perdas de dados na base de dados.

6

CONCLUSÃO E TRABALHO FUTURO

Este projeto está inserido no tema da proteção de dados, um tema importante na sociedade dos dias de hoje. Com a aprovação do RGPD, a legislação em vigor atualmente, todos os serviços de saúde devem possuir ferramentas que assegurem a privacidade e segurança dos dados pessoais dos intervenientes do seu universo, bem como garantir que existe um registo de todas as atividades de tratamento de dados realizadas. Também é importante que serviços como a plataforma AIDA possuam mecanismos de monitorização dos seus serviços, de modo a minimizar e prevenir possíveis falhas no seu funcionameno. Era, então, importante perceber se era possível garantir a monitorização da troca de dados na plataforma AIDA, se essa monitorização garantia o alinhamento com o RGPD e, por último, se era possível integrar mecanismos de monitorização aplicacional, não descurando nunca a monitorização da troca de informação. De modo a responder a estas questões foi criada uma ferramenta de monitorização das operações de tratamento de dados realizados no seio da plataforma AIDA, com funcionalidades que lhe permitiam a monitorização dos componentes da plataforma. Foi realizado um estudo sobre conceitos importantes para o seu desenvolvimento e sobre várias tecnologias que poderiam ser uma mais valia para o desenvolvimento da solução.

Após a realização desse estudo, foram definidas as metodologias de investigação e avaliação a utilizar durante o desenvolvimento da solução e, por fim, definida a arquitetura da solução, realizando vários testes para perceber se a solução encontrada respondia às questões expostas e aos objetivos traçados aquando da apresentação das metodologias. A solução foi testada localmente, tendo sido obtidos resultados favoráveis, que permitiram que os *dashboards* fossem mais facilmente idealizados e comprovada tanto a sua utilidade como a dos mecanismos de monitorização aplicacional.

Após os testes locais a solução foi implementada no Centro Hospitalar do Tâmega e do Sousa, utilizando uma arquitetura inicial, utilizando Docker para encapsular as tecnologias utilizadas e que consistia em *channels* MirthConnect, para extrair e transformar as mensagens armazenadas na base de dados da plataforma, enviando-as para o Logstash, que filtrava consoante a tabela de origem e enviava para o índice ElasticSearch correspondente, sendo as mensagens mostradas no Kibana. Esta solução teve um obstáculo importante na sua

implementação: a diferença horária das datas guardadas na base de dados da plataforma e a interpretação realizada pelas tecnologias da ferramenta. Enquanto que as datas eram extraídas da base de dados na hora de verão, no ElasticSearch eram interpretadas como UTC, ou seja, como tendo uma hora a menos, criando um atraso de 1 hora nas mensagens armazenadas. A solução foi adicionar um *script* nos *channels* MirthConnect para converter a data em UTC. Depois de implementada a solução foram construídos os *dashboards* idealizados na fase anterior. Os resultados retirados da implementação em ambiente real foram extremamente positivos, pois a ferramenta desenvolvida permite que seja cumprida a legislação sobre proteção dos dados pessoais, servindo-se das valências da base de dados ElasticSearch para armazenar todas as interações realizadas entre a plataforma AIDA e os demais intervenientes no ambiente hospitalar em que esta se insere.

Posteriormente, foi alterada a arquitetura da ferramenta, tendo sido removido o MirthConnect para a recolha e tratamento das mensagens ser efetuado ser apenas efetuado pelo Logstash. Esta alteração trouxe uma maior uniformização das operações de recolha e processamento de dados, garantindo também a possibilidade de acrescentar integrações com outros componentes da plataforma, de modo a monitorizar o funcionamento desta, retirando, no percurso, complexidade e dependências ao projeto. Nesta solução o problema do atraso das datas armazenadas no ElasticSearch foi solucionado ao mudar a *query* feita à base de dados da plataforma, sendo as datas extraídas em UTC.

Por fim, numa fase final do projeto, foi integrada na solução uma ferramenta para recolha de métricas de sistema, o Beats, que permitiu que fosse testada a monitorização da máquina em que a ferramenta está alojada, tendo sido obtidos resultados favoráveis à utilização desta ferramenta para a monitorização dos componentes da plataforma. A inclusão dos mecanismos adicionais de monitorização trouxe também resultados favoráveis devido à quantidade e qualidade das métricas produzidas e à facilidade de incluir novos módulos na ferramenta, por via das tecnologias utilizadas nesta serem facilmente customizáveis.

A solução abordada nesta dissertação ajuda à implementação desta legislação na plataforma AIDA, através dos registos acerca da informação trocada no seu seio. Deste modo é possível cumprir um dos desafios trazidos pelo RGPD, garantindo também o cumprimento do regulamento da interoperabilidade da saúde. Além disso, este sistema possibilita também a oferta à plataforma AIDA de funcionalidades para ajudar a deteção de erros e anomalias, através dos seus *dashboards* e dos mecanismos de monitorização aplicacional.

As últimas adições feitas na ferramenta de monitorização levantaram mais questões que irão necessitar de ser investigadas em trabalho futuro. Questões como que métricas são mais essenciais quando se trata de monitorizar os componentes da plataforma AIDA ou que *dashboards* relativos a essas métricas são necessários para uma monitorização eficiente são questões que irão certamente precisar de resposta.

O sistema irá ser implementado nas restantes unidades de saúde em que a plataforma AIDA esteja presente, sendo no futuro utilizado para monitorizar a saúde de todas as instâncias da plataforma AIDA e as operações de tratamento de dados realizadas em todas elas. É também necessário resolver algumas fraquezas e aproveitar algumas oportunidades expostas na análise SWOT realizada, tais como adicionar uma base de dados para *backup* da informação armazenada no ElasticSearch, de modo a prevenir eventuais falhas ou aproveitar para integrar mais mecanismos de monitorização. Por último, poderão também ser armazenadas no ElasticSearch as mensagens HL7 criadas nas unidades de saúde, permitindo a adição de *dashboards* relativos a essas mensagens não diretamente relacionados com as operações de tratamento de dados. Isto poderá também permitir que, em trabalho futuro os dados armazenados na base de dados sirvam para detetar padrões no funcionamento das unidades de saúde, através do número de mensagens trocadas diariamente ou o tipo de mensagens mais frequentes a certa hora do dia, utilizando algoritmos de mineração de dados.

BIBLIOGRAFIA

[1] Iakovidis I. Towards personal health record: current situation, obstacles and trends in implementation of electronic healthcare record in europe. *International Journal of Medical Informatics*, 1998.

[2] Cardoso L., Marins F., Portela F., Santos M., Abelha A., and Machado J. The next generation of interoperability agents in healthcare. *International Journal of Environmental Research and Public Health*, 2014.

[3] Duarte J., Portela F., Santos M., Abelha A., and Machado J. Electronic health record in dermatology service. *Enterprise Information Systems*, 2011.

[4] Regulamento (ue) 2016/679 do parlamento europeu e do conselho, de 27 de abril de 2016, relativo à proteção das pessoas singulares no que diz respeito ao tratamento de dados pessoais e à livre circulação desses dados e que revoga a diretiva 95/46/ce (regulamento geral sobre a proteção de dados). *Jornal Oficial da União Europeia*, L 119, 2016.

[5] Mead C. N. Data interchange standards in healthcare it—computable semantic interoperability: Now possible but still difficult, do we really need a better mousetrap? *J. Healthcare Inf. Manag.*, 2006.

[6] Rezaei R., Chiew T. K., Lee S. P., and Aliee Z. S. Interoperability evaluation models: A systematic review. *Computers in Industry*, 2014.

[7] Tolk A., Diallo S. Y., and Turnitsa C. D. Applying the levels of conceptual interoperability model in support of integratability, interoperability, and composability for system-of-systems engineering. *J. Syst. Cybernet. Inf*, 2007.

[8] Machado J., Abelha A., Neves J., and Santos M. Ambient intelligence in medicine. *Proceedings of the IEEE-Biocas*, 2006.

[9] Iroju O., Soriyan A., Gambo I., and Olaleke J. Interoperability in healthcare: benefits, challenges and resolutions. *International Journal of Innovation and Applied Studies*, 2013.

[10] Hl7 international, 2019. URL https://www.hl7.org/index.cfm?ref=nav.

[11] Peixoto H., Santos M., Abelha A., and Machado J. Intelligence in interoperability with aida. *In Foundations of Intelligent Systems*, 2012.

[12] Hl7 standards v2 product suite, 2019. URL http://www.hl7.org/implement/standards/product_brief.cfm?product_id=185.

[13] Saripalle R., Runyan C., and Russell M. Using hl7 fhir to achieve interoperability in patient health record. *Journal of biomedical informatics*, 2019.

[14] Choudhary V. Software as a service: Implications for investment in software development. *40th Annual Hawaii International Conference on System Sciences (HICSS'07)*, 2007.

[15] Monitoring that scales with a decentralizing and growing development team, 2020. URL https://www.datadoghq.com/pdf/Bazaarvoice_Datadog_Case_Study.pdf.

[16] Industrial solutions, 2020. URL https://www.stanleyblackanddecker.com/our-businesses/industrial#Engineered-Fastening-Solutions.

[17] Stanley black & decker transforms work with support from loggly, 2020. URL https://www.loggly.com/resource/stanley-black-decker-transforms-work-with-support-from-loggly/.

[18] Kufel Ł. Tools for distributed systems monitoring. *Foundations of computing and decision sciences*, 2016.

[19] Renita J. and Elizabethh N. E. Network's server monitoring and analysis using nagios. *International Conference on Wireless Communications, Signal Processing and Networking*, 2017.

[20] Agrawal V., Kotia D., Moshirian K., and Kim M. Log-based cloud monitoring system for openstack. *2018 IEEE Fourth International Conference on Big Data Computing Service and Applications (BigDataService)*, 2018.

[21] Betke E. and Kunkel J. Real-time i/o-monitoring of hpc applications with siox, elasticsearch, grafana and fuse. *International Conference on High Performance Computing*, 2017.

[22] Oliveira R., Ferreira D., Ferreira R., and Cruz-Correia R. Open-source based integration solution for hospitals. *IEEE 29th International Symposium on Computer-Based Medical Systems*, 2016.

[23] Truong N. B., Sun K., Lee G. M., and Guo Y. Gdpr-compliant personal data management: A blockchain-based solution. *IEEE Transactions on Information Forensics and Security*, 2019.

[24] Peffers K., Tuunanen T., Rothenberger M. A., and Chatterjee S. A design science research methodology for information systems research. *Journal of Management Information Systems*, 2007.

[25] Bortis G. Experiences with mirth: An open source health care integration engine. *Proceedings of the 30th international conference on Software engineering*, 2008.

[26] Rodriguez J. C. C., Stäubert S., and Löbe M. Automated import of clinical data from hl7 messages into openclinica and transmart using mirth connect. *Studies in health technology and informatics*, 2016.

[27] Smith J. E. and Nair R. The architecture of virtual machines. *Computer*, 2005.

[28] Linux containers, 2019. URL https://linuxcontainers.org/lxc/introduction/.

[29] What is a container, 2019. URL https://www.docker.com/resources/what-container.

[30] Kudyba S. P. Healthcare informatics: Improving efficiency and productivity. *Data Mining in Health Care*, 2010.

[31] Manyika J. Big data: The next frontier for innovation, competition, and productivity, 2011. URL https://www.mckinsey.com/business-functions/mckinsey-digital/our-insights/big-data-the-next-frontier-for-innovation.

[32] Schollmeier R. A definition of peer-to-peer networking for the classification of peer-topeer architectures and applications. *Proceedings First International Conference on Peer-to-Peer Computing*, 2001.

[33] Divya M. S. and Goyal S. K. Elasticsearch: An advanced and quick search technique to handle voluminous data. *Compusoft 2.6*, 2013.

[34] Kuć R. and Rogoziński M. *Mastering ElasticSearch*. Packt Publishing Ltd, 2017.

[35] Dixit B., Kuc R., Rogozinski M., and Chhajed S. *Elasticsearch: A Complete Guide*. Packt Publishing Ltd, 2017.

[36] Kononenko O., Baysal O., Holmes R., and Godfrey M. W. Mining modern repositories with elasticsearch. *Proceedings of the 11th Working Conference on Mining Software Repositories*, 2014.

[37] Apache solr features, 2020. URL https://lucene.apache.org/solr/features.html.

[38] Karambelkar H. V. *Scaling Big Data with Hadoop and Solr*. 2015.

[39] Pandya M. Cloud computing for libraries: A swot analysis. 2012.

[40] Malan R. and Bredemeyer D. Functional requirements and use cases. *Bredemeyer Consulting*, 2001.

7

PUBLICAÇÃO

7.1 OVERCOMING CHALLENGES IN HEALTHCARE INTEROPERABILITY REGULATORY COMPLIANCE

Autores: António Castanheira, Hugo Peixoto e José Machado
Título: Overcoming Challenges in Healthcare Interoperability Regulatory Compliance
Conferência: Advances in Intelligent Systems and Computing
Ano de Publicação: 2020

Resumo: There has been a significant increase in the quantity of information stored digitally by health institutions. Such information contains personal data from the actors in their universe. Thus, is crucial that it is governed by a set of rules, in order to allow it to be understood without losing important data.

With the increased use of digital tools for storing and exchanging information, ethical issues began to arise in the context of the privacy of personal data. Questions about access, processing, treatment and storage of personal data became increasingly important in society, leading to the creation of the General Data Protection Regulation (GDPR) in force at European level.

GDPR is one of the main challenges in healthcare interoperability regulatory compliance, therefore the proposed architecture shows an approach to enforce GDPR compliance into Agency for Integration, Diffusion and Archive Platform (AIDA), which is held by several healthcare unities in Portugal, using technologies like ElasticSearch and Kibana.

Palavras-Chave: GDPR, ElasticSearch, Interoperability, Electronic Health Record
Estado de Publicação: Publicado