Instituto Politécnico de Viseu

Escola Superior de Tecnologia e Gestão de Viseu

Departamento de Informática

Relatório de Projeto/Estágio

Plataforma de Gestão de *Logs*

Realizado em:

Grupo Visabeira

por:

António Jorge Ferreira Ramos

Orientadores:

ESTGV: Carlos Alberto Tomás Simões

Grupo Visabeira: Paulo Castanheira

Viseu, 2019

Instituto Politécnico de Viseu

Escola Superior de Tecnologia e Gestão de Viseu

Departamento de Informática

Relatório de Projeto/Estágio

Curso de Licenciatura em

Engenharia Informática

Plataforma de Gestão de Logs

Realizado em

Grupo Visabeira, Palácio do Gelo

de 18 de fevereiro de 2019 a 8 de junho de 2019

por

António Jorge Ferreira Ramos

Ano Letivo 2018/2019

Orientadores:

ESTGV: Carlos Alberto Tomás Simões

Grupo Visabeira: Paulo Castanheira

Viseu, 2019

Agradecimentos

Ao longo do desenvolvimento deste projeto tive ajuda fundamental para que este esteja funcional e objetivo, por isso não queria deixar de agradecer às pessoas que me ajudaram.

São eles Tiago Falé meu coordenador da Visabeira, Paulo Castanheira meu orientador na Visabeira, à minha família por me dar incentivos e em especial à minha irmã, que me ajudou a construir o relatório final, ao Professor Carlos Simões meu orientador da escola que se mostrou disponível para se reunir comigo e discutir o trabalho.

Queria agradecer aos meus colegas da Visabeira e do curso pelo apoio prestado ao longo deste projeto. A todos os professores que tive ao longo do meu percurso pelos conhecimentos que me transmitiram. A todos que me ajudaram digo vos simplesmente muito obrigado, pelo apoio prestado.

Índice

Índice de Figuras

1. Introdução

Este relatório descreve o trabalho elaborado no âmbito da unidade curricular "***Projeto***" do plano curricular da Licenciatura em Engenharia Informática, de acordo com o plano curricular da Escola Superior de Tecnologia e Gestão de Viseu. Este projeto decorreu durante o segundo semestre do ano letivo 2018/19 na empresa "***Grupo Visabeira***", sedeada no Palácio do Gelo Shopping, em Viseu.

No decorrer do relatório será exposto o tema do projeto e os seus respetivos objetivos, e um pequeno enquadramento sobre a entidade acolhedora. Posteriormente, serão explicadas e demonstradas as ferramentas utilizadas para a realização do projeto. De seguida será explicado ao pormenor o software utilizado e potencial deste.

Na respetiva conclusão, irão ser apresentados os obstáculos que surgiram, os testes do projeto, e as respetivas soluções dos mesmos.

1.1. Enquadramento

Como o "***Grupo Visabeira***" está cada em constante crescimento e tem muitos funcionários a usar os seus serviços internos (como *mail*), clientes a consultar os seus serviços, é necessário controlar os respetivos *logs* dos seus sistemas e consultar as informações.

Para solucionar este problema/necessidade a empresa apresentou ao Departamento de Informática uma proposta de projeto baseado em Gestão de Logs, que posteriormente foi disponibilizada no portal DAPE.

Ao longo do relatório vai ser explicado e detalhado todo o projeto por temas.

1.1.1. Empresa

"O ***Grupo Visabeira*** *tem 35 anos de atividade com a sua génese em Viseu, onde mantém a sede no Palácio do Gelo, nas áreas das Telecomunicações e Construção. Após tornar-se líder no mercado nacional, consolidou o seu core business e alargou a sua área de atuação a diversos setores, iniciando paralelamente um processo de internacionalização. Atualmente está presente em 15 países e comercializa os seus produtos e serviços para mais de 70 nações*" [1].

1.1.2. O projeto

Este projeto visa a implementação de uma plataforma ***opensource***, chamada *elk*, que permite centralizar *logs* produzidos pelos diversos sistemas ou serviços, tais como *apache*, *nginx*, *system*, etc. Esta plataforma vai ter a informação dos *logs* gerada pelos serviços presentes na empresa e vai apresentar visualizações das mesmas, bem como estatísticas (exemplos: falhas, *tops*, médias, contabilizar pedidos, etc). Esta plataforma vai ajudar o departamento de sistemas de informação a controlar os erros ocorridos dentro e fora da empresa.

1.2. Objetivos a atingir

Este projeto tem como objetivo reunir todos os dados de tipos de ficheiros de *logs* de modo a potenciar uma gestão mais fácil desses mesmos dados. Do ponto de vista da entidade acolhedora, esta gestão é necessária para controlar o que acontece dentro e fora da empresa. O *software* vai ter a análise de dados, permitindo fazer pesquisas de dados, criar/modificar/apagar os *indexes* do *elasticsearch*, estatísticas dos dados, isto tudo num só site, levando assim a uma maior possibilidade de controlo sobre a empresa, quer internamente quer externamente.

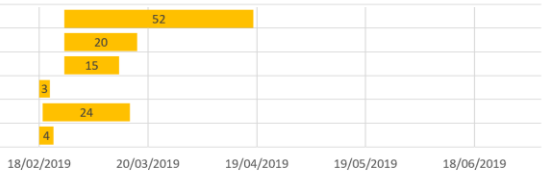
2. Análise de Sistemas

A entidade acolhedora **Grupo Visabeira** pretendia uma plataforma que gerisse os serviços deles tais como *haproxy*, *haproxymail* e *Web Exchange*, para isso foi proposto instalar e usar a ferramenta *elk*. Nesta plataforma vai ser possível pesquisar os *logs* que queremos, fazer estatísticas, criar gráficos e gerir os serviços.

Os passos que se seguem na imagem seguinte, gráfico de Gantt, representam as tarefas dadas pelo orientador

Cronograma de Tarefas - Gráfico de Gantt

Tarefa	Data Início	Duração (d)	Data Fim
Informação dos requisitos do trabalho	18/02/2019	4	21/02/2019
Pesquisa sobre o código opensource elk	19/02/2019	24	19/04/2019
Instalação da mv e respetivas conf do software	18/02/2019	3	20/02/2019
Construção de groks em ficheiros logs	25/02/2019	15	27/05/2019
Criação de dashboards e estatísticas	25/02/2019	20	05/06/2019
Realização do relatório do projeto	25/02/2019	52	12/06/2019



Legenda:
mv - máquina virtual
conf - configurações
d - dias

Figura 1 - Gráfico de Gantt

3. Ferramentas e Tecnologias

3.1. ELK.

ELK é o acrónimo para o conjunto de três projetos *opensource*: *Elasticsearch*, *Logstash* e *Kibana*. Nas secções seguintes descrevem-se, de forma sucinta, cada um destes três projetos.

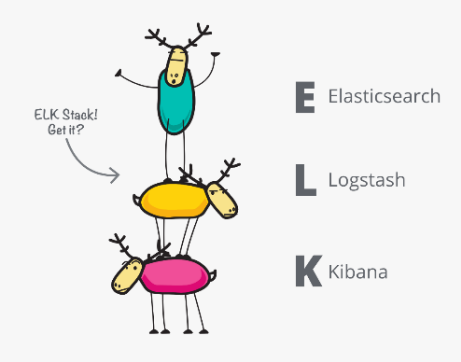


Figura 2 - Legenda elk

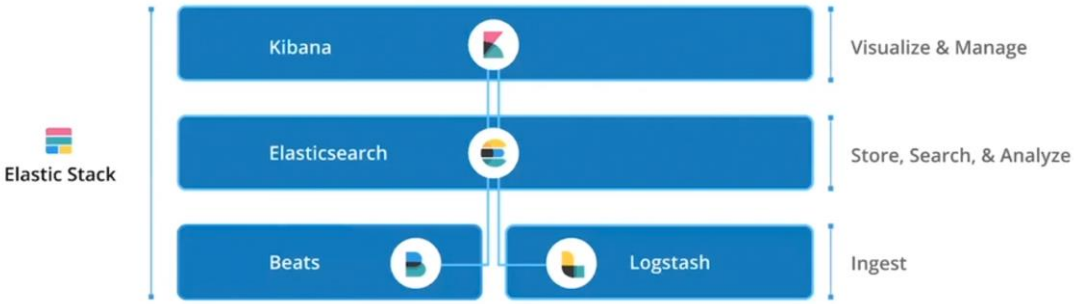


Figura 3 - Definições ELK

3.1.1. Elasticsearch.

O *ElasticSearh* é um mecanismo distribuído moderno de pesquisa e análise "***RESTful***" capaz de solucionar a maior parte dos casos de uso. Os *Web services RESTful* permitem que os sistemas solicitantes acedam e manipulem representações textuais de recursos da *Web* usando um conjunto uniforme e predefinido de operações sem estado. O *Elasticsearch* foi baseado no ***Apache Lucene***, para pesquisa e análise eficiente de grandes quantidades de dados. Foi desenvolvido por Shay Banon em 2010, em linguagem java, disponibilizado sob os termos ***Apache License***. Este vai ser o coração do ***Elastic Stack***, porque é ele que vai comunicar com o ***Kibana***, o ***Logstash*** e o ***Filebeat***. O ***Elasticsearch*** tem como principal função guardar, pesquisar e analisar dados provenientes do ***Logstash***.

Figura 4 - Elasticsearch (Endereço: x.x.x.x:9200, onde x é o endereço da máquina)

O ***Elasticsearch*** permite realizar e combinar muitos tipos de pesquisas, nomeadamente estruturadas, não estruturadas, geo e métrica. Este mecanismo, denominado de ***query***, pode ser feito quer no ***elastic*** quer no ***kibana*** e permite analisar todos os documentos que tenham uma determinada localização, palavra, código, etc.

A implementação dos índices no ***elastic*** permitiu-lhe uma grande melhoria no desempenho.

O *elasticsearch é ainda* escalável, ou seja, pode ser utilizado por vários servidores desde que os identifiquemos na configuração, e permite detetar falhas nas configurações ou no sistema, protegendo assim o *cluster*, estando seguro e acessível.

3.1.2. Conceitos *Elasticsearch*

- ***Near Realtime (NRT):*** *Elasticsearch* é uma plataforma de pesquisa *near-realtime*. Isto significa que existe uma pequena latência do tempo que o *index* demora até o documento se torne pesquisável.
- ***Cluster:*** Um ***cluster*** é uma coleção de um ou mais nós (máquinas) que juntos mantêm os dados de forma consistente e fornecem recursos de indexação e pesquisa. Um *cluster* é identificado por um nome exclusivo que, por omissão, é ***Elasticsearch.*** Ter este nome é importante, pois um nó só pode fazer parte de um cluster;
- ***Node (Nó):*** Um **nó** é um servidor que faz parte do *cluster*, armazena os dados e participa no processo de indexação e pesquisa do *cluster*. Tal como um cluster, um nó é identificado por um nome. Por omissão o nome é um ***UUID (Universally Unique IDentifier)*** aleatório no arranque do servidor, mas o utilizador pode renomear o nó. Podemos juntar os nós para *clusters* específicos. Por omissão, cada *nó* é configurado para se juntar a um *cluster* denominado *elasticsearch*, isto é, podem ser inicializados vários nós em redes que, supondo que eles possam ser descobertos, todos se irão juntar num único cluster. Num único *cluster*, podemos ter quantos nós quisermos.
- ***Índices (Indexs), Tipos, documentos:*** Os ***índices*** são as bases de dados, conjunto de documentos com características semelhantes, do ***Elasticsearch***. Por exemplo, no Grupo Visabeira podemos ter um índice para os funcionários, outro para os fornecedores e outro para os clientes. O que define o índice é o nome. Os tipos são como as tabelas de uma base de dados. Os documentos são as informações dessas mesmas tabelas, onde a informação pode ser indexada. Podemos ter um documento para os funcionários, outro para os fornecedores e outro para os clientes.
- ***Shards e Replicas:*** Um *index* pode guardar uma quantidade de informação que exceda os limites do *hardware* para um só nó. Para resolver este problema, o *elasticsearch* subdivide o *index* em várias peças chamadas *shards.* Quando criamos o *index*, podemos definir o número de *shards* que queremos. Cada *shard* é *fully-funtional* e independente do respetivo "index" que pode ser alojado em qualquer nó do *cluster*.

  O *shard* é importante por duas razões:

  - Permite dividir/escalar horizontalmente o conteúdo;
  - Permite distribuir e paralelizar operações em *shards* (potencialmente em vários nós), aumentando assim o desempenho/rendimento.

A mecânica de como um pedaço do documento é distribuído e também como os seus documentos são agregados de volta às solicitações de pesquisa são tarefas completamente geridas pelo *Elasticsearch.*

Num ambiente de rede/nuvem em que as falhas podem ser esperadas a qualquer momento, é muito útil e altamente recomendado ter um mecanismo de *failover* no caso de um *shard/node* ficar *offline* ou desaparecer por qualquer motivo. Para esse fim, o *elasticsearch* permite efetuar uma ou várias cópias dos fragmentos do índice.

A replicação é importante por duas razões:

- Fornece alta disponibilidade no caso de um *shard/node* falhar. Por esse motivo, é importante observar que um *shard* original/principal do qual foi copiado.
- Permite aumentar a velocidade de pesquisa/taxa de transferência, uma vez que a pesquisa pode ser executada nas várias réplicas em paralelo.

3.1.3. *Logstash*

O *Logstash* centraliza, transforma e armazena os dados. *Logstash* é um *pipeline* de processamento de dados do lado do servidor, *Filebeat* aberto, que insere dados de várias fontes simultaneamente, transforma e envia para o *stash*.

O ***Logstash*** divide-se em três partes:

- ***Input*:** Onde se define os caminhos dos ficheiros de logs (tanto pode ser no ***Logstash*** como no ***filebeat***). Os dados normalmente são dispersos ou isolados através de vários sistemas nos diversos formatos. O ***Logstash*** tem uma variedade de ***inputs*** que são em eventos de uma multitude de fontes comuns, todos ao mesmo tempo. Ingere facilmente a partir dos *logs*, métricas, aplicações *web*, *data stores*, e variados serviços *AWS (Amazon Web Services)*, suportando diversos tipos de inputs que leem o conteúdo do ficheiro. A lista dos vários inputs disponíveis pode ser consultada em [2].

- ***Filter*:** Onde se filtra os *logs* para a informação e campos que queremos. Os filtros do ***Logstash*** analisam cada evento, identificando os campos nomeados para criar a estrutura e os transformam para convergirem em um formato comum para uma análise mais poderosa. O ***Logstash*** transforma dinamicamente e prepara os dados, independentemente do formato ou da complexidade:

  - Derivar estruturas de dados não estruturados com *grok* (ver Nota abaixo);
  - Decifrar as coordenadas geográficas dos endereços IP;
  - Anonimizar os dados PII (***Personally Identifiable Information)***, exclui completamente os campos sensíveis;

- Facilitar o processamento geral, independente da origem de dados, formato ou esquema.

Nota: *Grok* é um *plugin* do filtro do *logstash* e é uma ótima maneira de transformar dados de *logs* não estruturados em algo estruturado e consultável. A maneira de trabalhar do *grok* é analisando o texto *pattern* com o texto do ficheiro de *log*. A sintaxe para o *grok pattern* é "%{SYNTAX:SEMANTIC}". A *Syntax* é o nome do *pattern* que vai fazer *match* ao texto do ficheiro de *log*, a semântica vai ser o nome do identificador que foi encontrada.

- ***Output*:** para onde vai a informação tratada pelo ***Logstash***.

O ***Logstash*** tem uma variedade de saídas que permitem *tracking* dos dados onde quisermos, dando nos assim uma enorme flexibilidade de desbloquear uma grande quantidade de casos de uso *downstream*.

3.1.4. Kibana

O ***Kibana*** dá forma aos dados e é a interface extensível do utilizador para configurar e gerir todos os aspetos do ***Elastic Stack***.

Como se mostra na figura 5, o *Kibana* divide-se em várias subsecções, nomeadamente, *Discover, Visualize, Dashboard, Timelion, Canvas, Machine Learning, Maps, Infrastructure, Logs, APM (Application Performance Monitoring), Uptime, Dev tools, Monitoring, Management.* Descrevem-se abaixo as funcionalidades de cada uma delas:

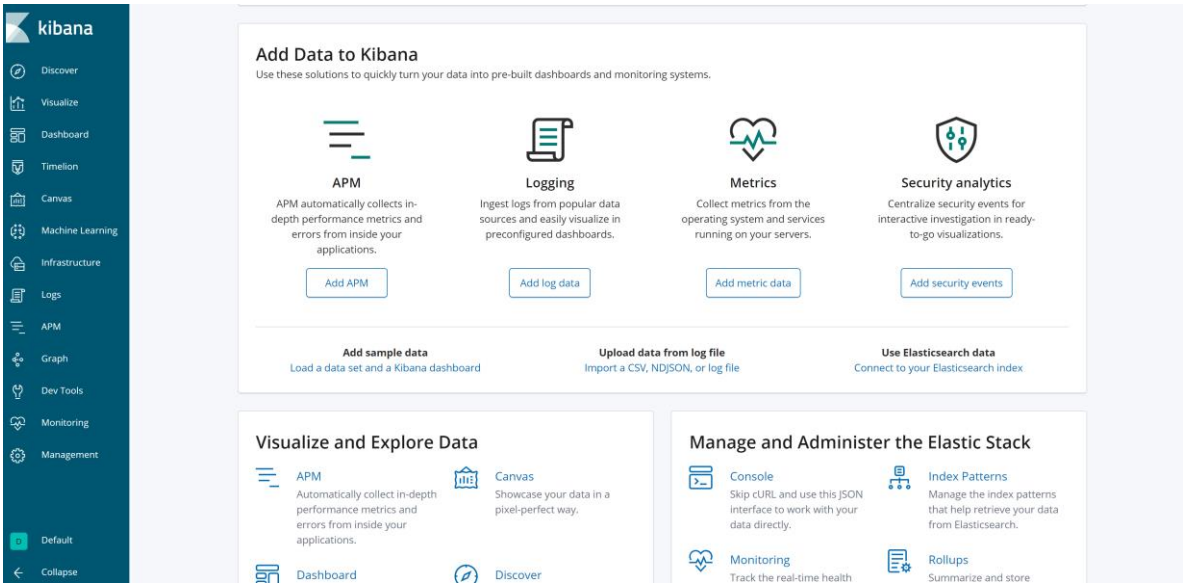


Figura 5 - Ilustração do kibana (Endereço: x.x.x.x:5601, onde x é o endereço da máquina)

- ***Discover***: Que permite a pesquisa de ***querie***, filtrar os resultados por pesquisa, e ver os dados do documento. Permite ainda ver o resultado do número de documentos que se encaixam na pesquisa e fornece estatísticas dos campos escolhidos (basta clicar em cima do campo – ver exemplo na Figura 6);

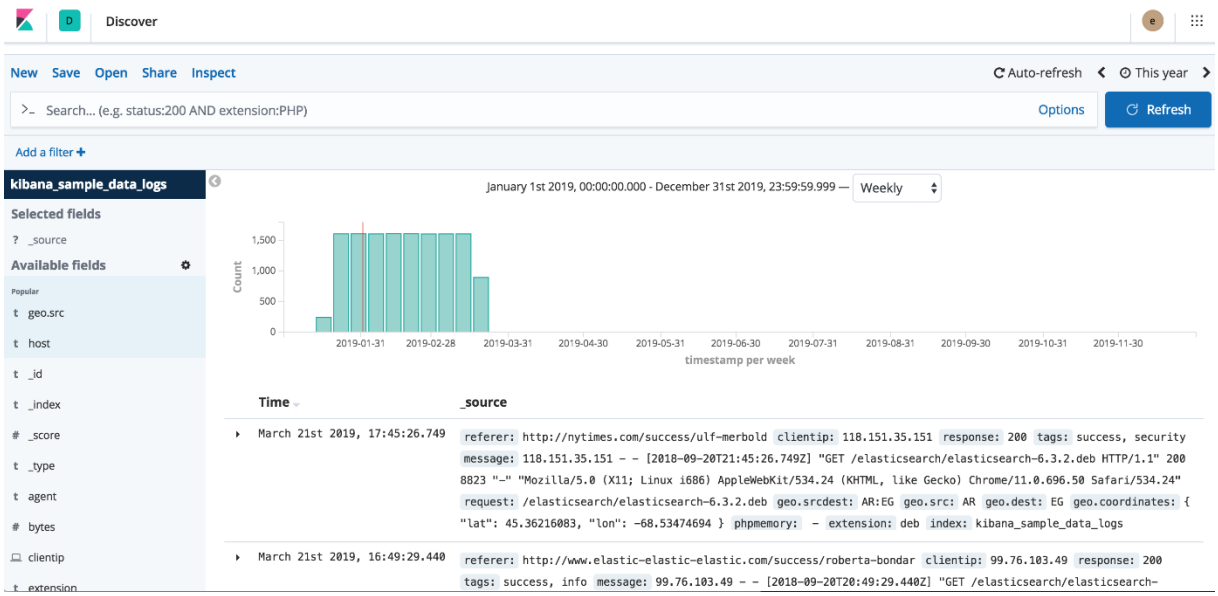


Figura 6 - Exemplo discover

- ***Visualize***: Permite criar visualizações dos dados com base no ***index*** do ***Elasticsearch***. É também possível criar ***dashboards*** que relacionem essas mesmas visualizações. As visualizações são baseadas em ***queries*** do ***Elasticsearch***. Usando séries de agregação (ver Nota) para extrair e processar dados, podem ser gerados gráficos que mostrem as tendências, picos e decaimento dos parâmetros pretendidos. Podemos também criar visualizações da pesquisa feita no ***discover*** ou começar uma por nós mesmos;
- ***Dashboard***: Permite mostrar uma coleção de visualizações e pesquisas. Podemos organizar, redimensionar e editar o conteúdo do painel e em seguida, guardar o ***dashboard***. (ver exemplo na Figura 7);

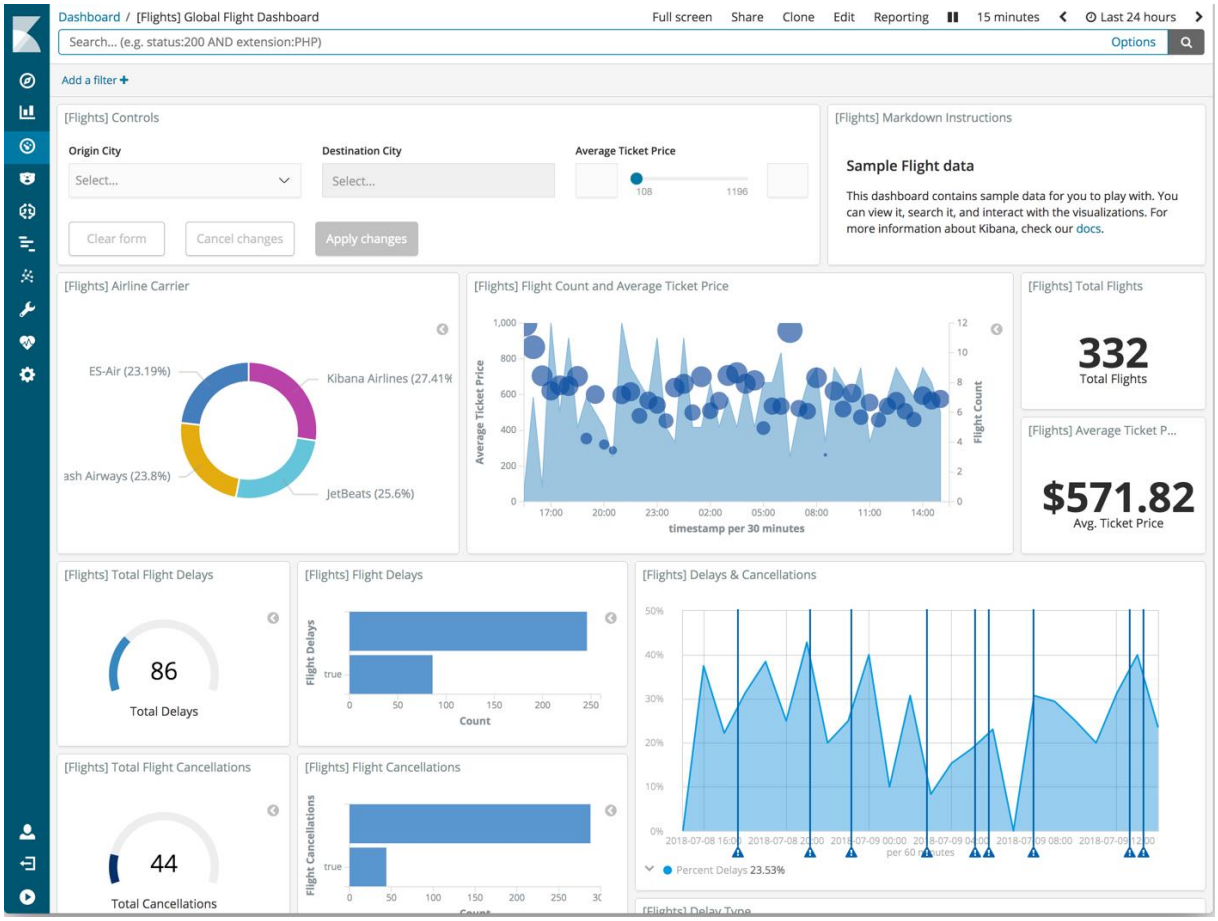


Figura 7 - Exemplo dashboard

- ***Timelion***: Trata-se de um visualizador temporal que permite combinar fontes de dados totalmente independentes em uma única visualização. É orientado por uma linguagem de expressão simples usada para recuperar dados de séries temporais, realiza cálculos para descobrir as respostas a perguntas complexas e visualizar os resultados;
- ***Canvas***: Serve para representar dados de maneiras como queremos. (ver exemplo na Figura 8);

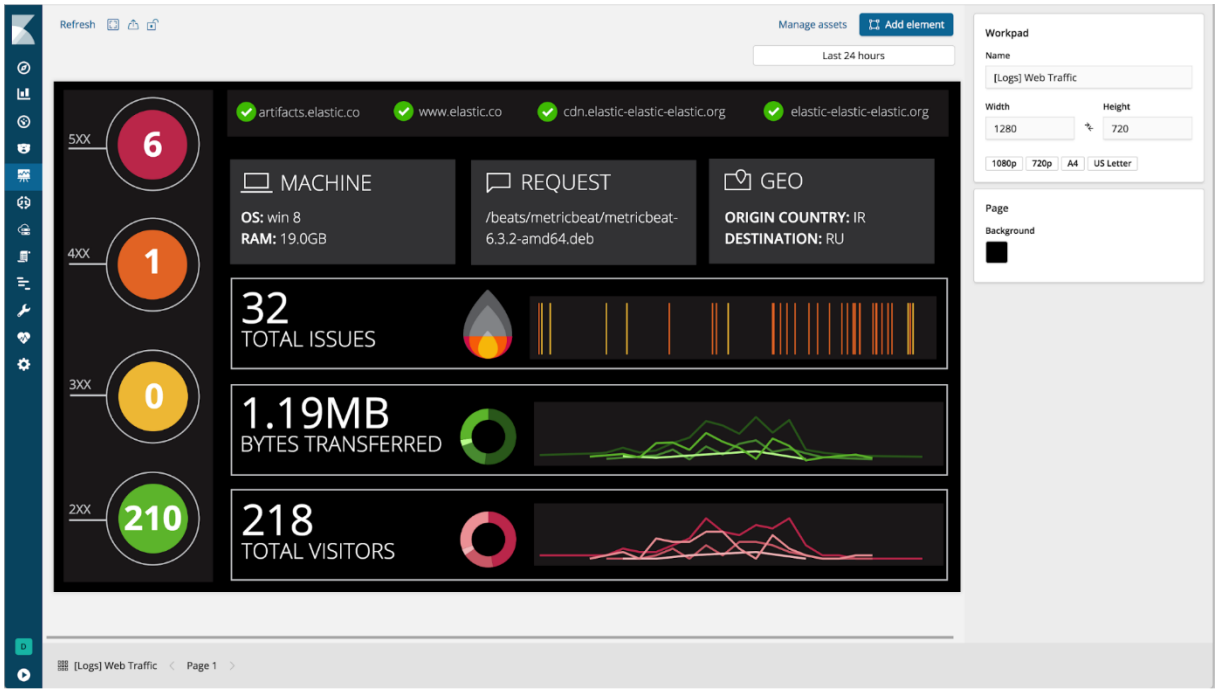


Figura 8 - Exemplo Canvas

- ***Machine Learning***: Nesta subsecção pode se encontrar estatísticas, anomalias da máquina. É composta por inteligência artificial que deteta anomalias e dá a perceber os resultados. Dentro do separador *machine learning* existe um outro que se chama *data visualizer* (Figura 9), aqui podemos ver os dados do nosso *index*;

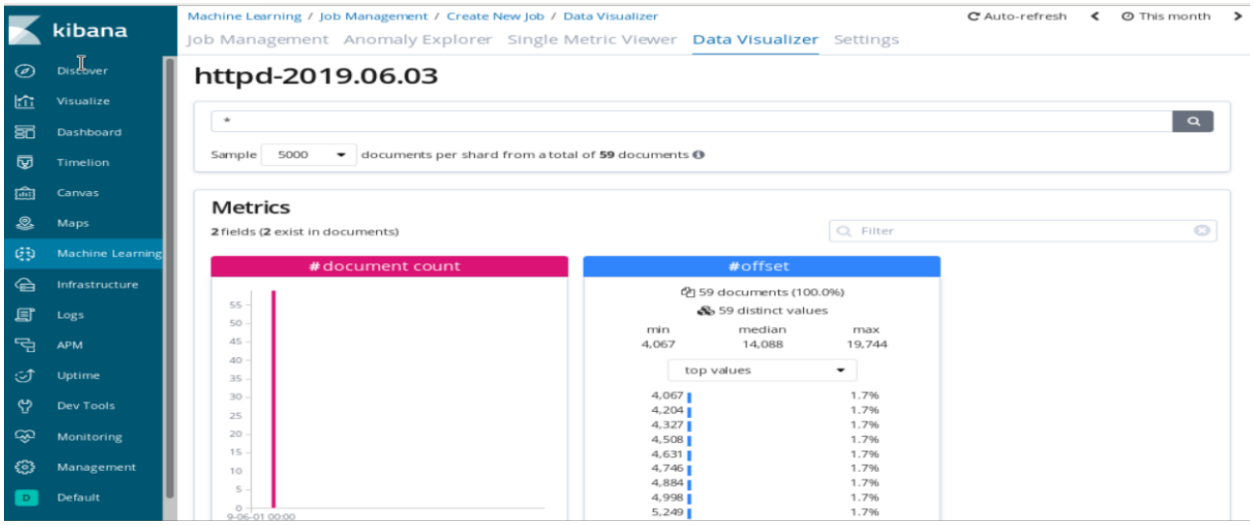


Figura 9 - Data visualizer

- ***Maps***: Permite mostrar as localizações dos *ips* pertencentes ao ficheiro de *log;*

- ***Infrastructure*:** Serve para monitorizar a nossa infraestrutura e identificar os problemas em tempo real. Podemos explorar as métricas e *logs* dos servidores mais comuns, *containers* e serviços;
- ***Logs*:** Permite visualizar os *logs de* servidores mais comuns, *containers* e serviços.;
- ***APM* (*Application Performance Monitoring*):** Monitoriza o desempenho dos dados e localizar rapidamente o desempenho dos *bottlenecks*;
- ***Uptime*:** Permite monitorizar o *status* dos *endpoints* da rede via *HTTPS*, *TCP* e *ICMP*. Permite explorar o status ao longo do tempo, pesquisar em monitores específicos e visualizar um rápido nível do ambiente selecionado;
- ***Dev tools*:** São ferramentas de desenvolvimento;

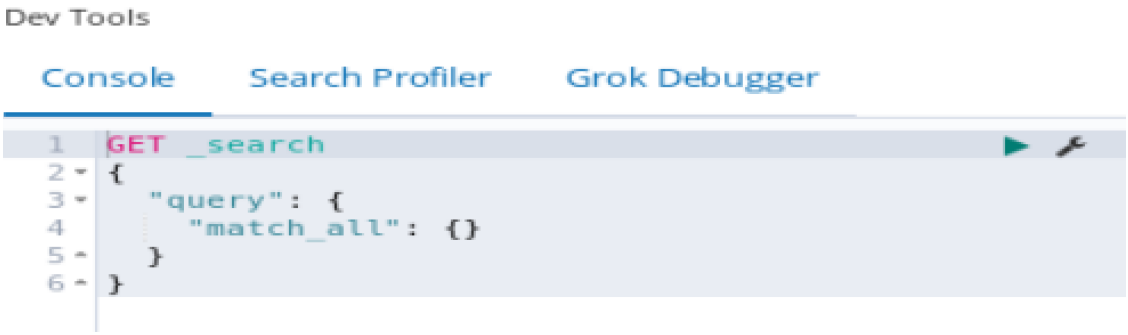


Figura 10 - Dev tools

- ***Monitoring*:** O *monitoring* divide-se em dois propósitos:

  - Visualizar dados do *Elastic Stack*. Pudemos ver a saúde e performance dos dados do *elasticsearch*, *Logstash* e *Beats* em tempo real;

  - Monitoriza o *Kibana* e monitoriza o *cluster*.

- ***Management:*** No *Management* é onde se executa a configuração temporal da execução do *kibana*, incluindo a configuração inicial e contínua de padrões de índice, configurações avançadas que ajustam os comportamentos do próprio *Kibana* e os vários objetos que podemos salvar, como pesquisas, visualizações e *dashboards*;

Nota: *Aggregations(agregações) – A Aggregations framework ajuda a agregar dados com base numa pesquisa de query*. Esta *Framework* é baseada em blocos de construção simples, chamados agregações, que podem ser compostos para criar resumos complexos dos dados. Uma agregação pode ser vista como uma unidade de trabalho que cria informações analíticas sobre um conjunto de documentos. O contexto de execução define o que é esse conjunto de documentos. Há diferentes tipos de agregações, cada um com o seu propósito e *output*, são eles o *bucketing*, *metric*, *matrix* e *pipeline*.

3.1.5. *Filebeat*

O ***Filebeat*** consiste em dois eventos principais: o ***input*** e o ***harvester***. Estes dois componentes trabalham em conjunto para finalizar ficheiros e enviar os respetivos dados para uma saída especificada.

O ***harvester*** é responsável por ler o conteúdo de um ficheiro único. O ficheiro é lido, linha a linha, e o conteúdo é enviado para a saída (*output*). O ***harvester*** é também responsável por abrir e fechar os ficheiros, o que significa que os descritores dos ficheiros continuam abertos enquanto o ***harvester*** está em execução. Se um ficheiro é removido ou renomeado enquanto está em "***harvested***", o ***Filebeat*** continua a ler o ficheiro. O espaço do disco do computador está reservado enquanto o ***harvester*** se desligue ou feche. Por omissão, o ***Filebeat*** mantém o ficheiro até que a opção "***close_inactive***" seja alcançada.

Parar o ***harvester*** tem as seguintes consequências:

- O ***handler*** do ficheiro é fechado, libertando os recursos subjacentes, se o ficheiro for eliminado enquanto o ***harvester*** ainda estava lendo o ficheiro.
- O ***harvesting*** do ficheiro só é iniciado outra vez após a conclusão do ***scan_frequency***.
- Se o ficheiro for movido ou removido enquanto o ***harvester*** estiver fechado, o ***harvester*** do arquivo não continuará.

O ***input*** é responsável por gerir os ***harvesters*** e encontrar todos os caminhos/ficheiros para ler.

Há vários tipos de *beats* (ver Nota) que servem para agregar dados. Dos que estão disponíveis foi usado o ***Filebeat*** uma vez que é uma ferramenta leve e ajuda a manter a simplicidade dos dados, oferecendo um caminho que reencaminha e centraliza *logs* e ficheiros.

O *filebeat* não falha nenhuma linha do ficheiro do *log*, se falhar, é memorizada a localização onde falha e, quando tudo estiver novamente *online*, a linha volta a ser reexaminada.

O *filebeat* vem ainda com modelos internos (*auditd*, *Apache*, *NGINX*, *System*, etc) que simplifica a coleção, filtragem e visualização de formatos de *log* comuns com um único comando.

O *filebeat* não permite que o *pipeline* entre em sobrecarga pois usa o protocolo "***backpressure-sensitive***" quando há dados enviados para o ***Logstash*** ou para o ***Elasticsearch***. Se o ***Logstash*** *crashar* avisa o ***Filebeat*** para abrandar a sua leitura. Após a resolução do *crash*, o ***Filebeat*** volta ao funcionamento normal.

Nota: *Beats* é uma plataforma exclusiva para enviar dados. Há vários tipos de *beats*, tais como *Filebeat*, *Metricbeat*, *Packetbeat*, *Winlogbeat*, *Auditbeat*, *Heartbeat* e *Funciontbeat*. Os *beats* têm de ser instalados nos próprios servidores, com os *containers*, ou implementados como funções (centralizando dados no *elasticsearch*). Os *beats* também podem enviar dados para o *logstash* para transformação e análise,

3.2. CentOS 7

Um dos requisitos do projeto era instalar neste sistema operativo com base em Linux, logo tudo que vai ser apresentado e entregue vai ser neste sistema operativo.

3.3. Cerebro 0.8.1

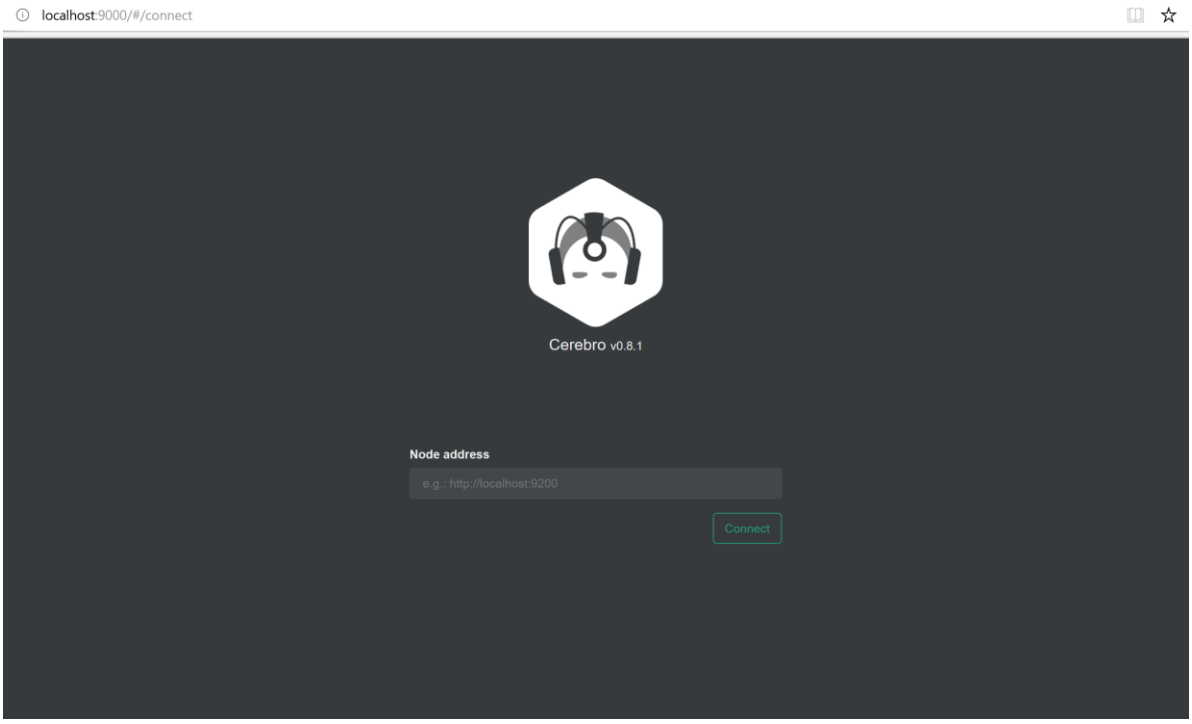


Figura 11 - Cerebro

O *Cerebro* permite fazer uma gestão do *elasticsearch* num ambiente gráfico onde o *elk* não o permite. O *cerebro* foi usado num âmbito de criar um *index* predefinido que permita o uso de geo localização, já que sem este não se conseguia localizar os pontos ips, no Kibana. Neste código *opensource* podemos ainda monitorizar o *elasticsearch.*

4. Desenvolvimento do projeto

4.1. ETAPAS

4.1.1. Apresentação de *logs* do sistema no *kibana*.

Esta etapa está subdividida em 2:

1- Utilização de ***template***: Nesta etapa foi usado o módulo *system* do ***Filebeat***. Para isso o módulo foi ativado ("*filebeat modules enable system*") e foram corridos os dois comandos seguintes:

- sudo filebeat setup --template -E output.logstash.enabled=false -E 'output.elasticsearch.hosts=["x.x.x.x:9200"]'(x ip do computador);

- sudo filebeat setup -e -E output.logstash.enabled=false -E output.elasticsearch.hosts=['x.x.x.x:9200'] -E setup.kibana.host=x.x.x.x:5601;

2- Utilização do ***Logstash***: para obter os campos das linhas do ***system*** foi usado o seguinte *grok*

```
filter {
        if [fileset][name] == "syslog"{
                grok {
                match => {"message" => "\{SYSLOGTIMESTAMP:syslog_timestamp} %{SYSLOGHOST:hostname} %{DATA:syslog_program} (?:\%{POSINT:syslog_pid}\)?: %{GREEDYDATA:syslog_message}"}
                    }
                }} .
```

(campos a rodeados a preto na figura 14)

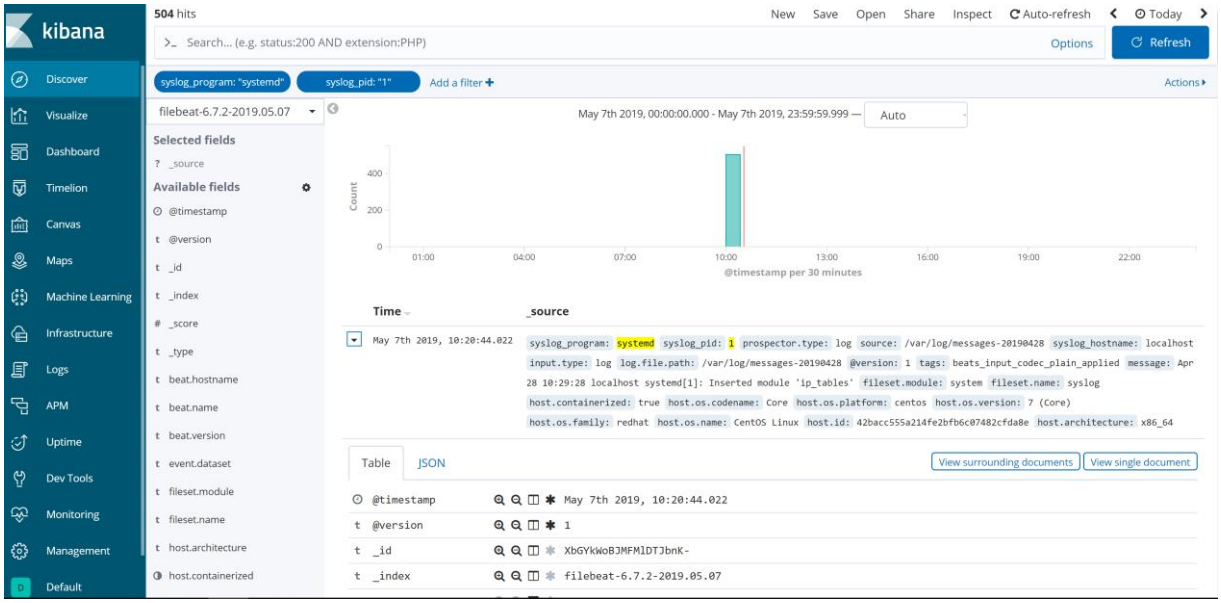


Figura 12 - Discover system logs

Figura 13 - Linha log syslog

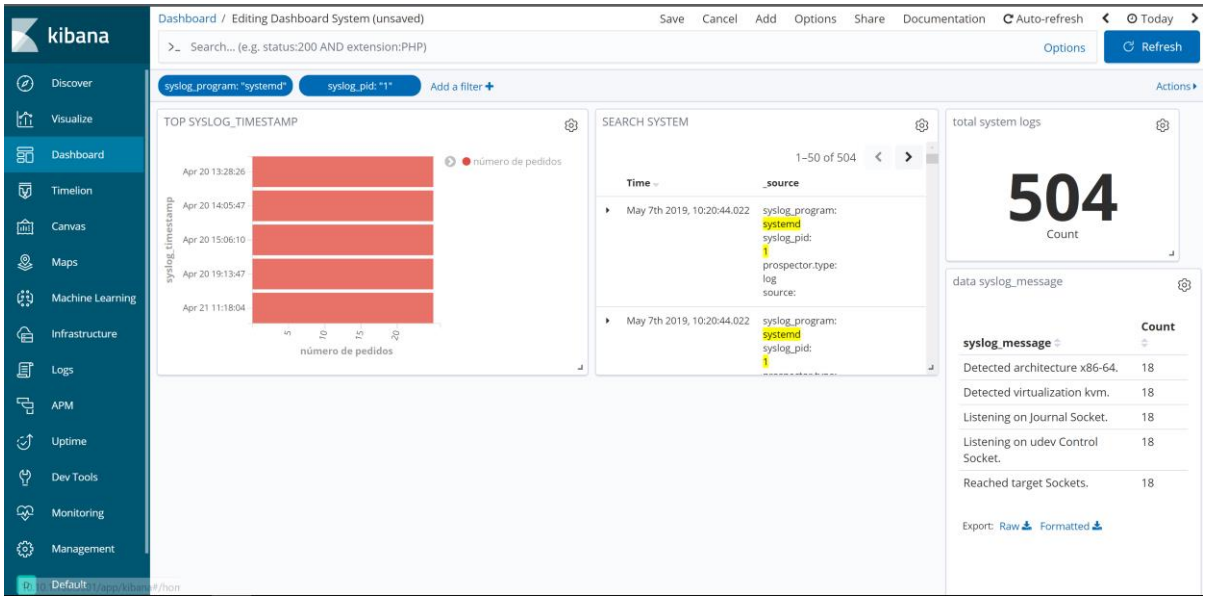


Figura 14 - Dashboard syslog

4.1.2. Apresentação do apache do sistema no *kibana*.

De modo a permitir a apresentação dos *logs* do *apache* foi instalado o serviço **httpd** (*apache* no centOS).

O primeiro passo foi configurar o httpd (ficheiro **/etc/httpd/conf/httpd.conf**) colocando o ip do computador e a respetiva porta 80. Para permitir a consulta do site nos computadores externos à máquina é preciso colocar o comando:

firewall-cmd --permanent –add-port=80/tcp

Figura 15 - Página httpd

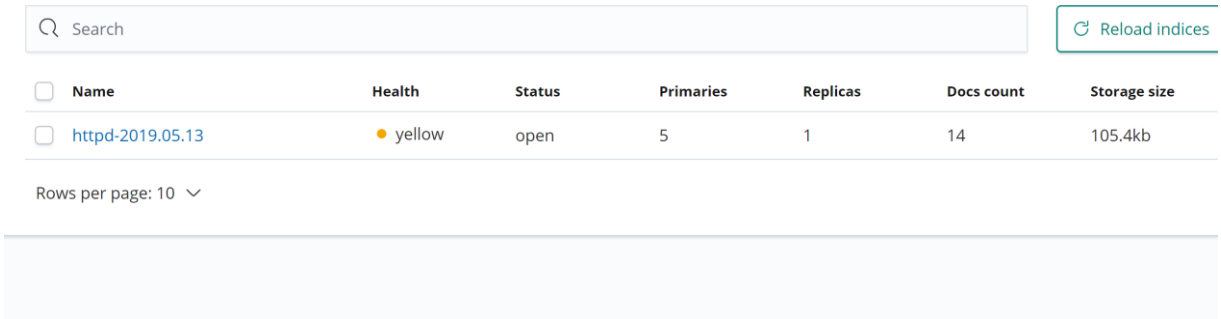


Figura 16 - criação do index do elasticsearch para o httpd

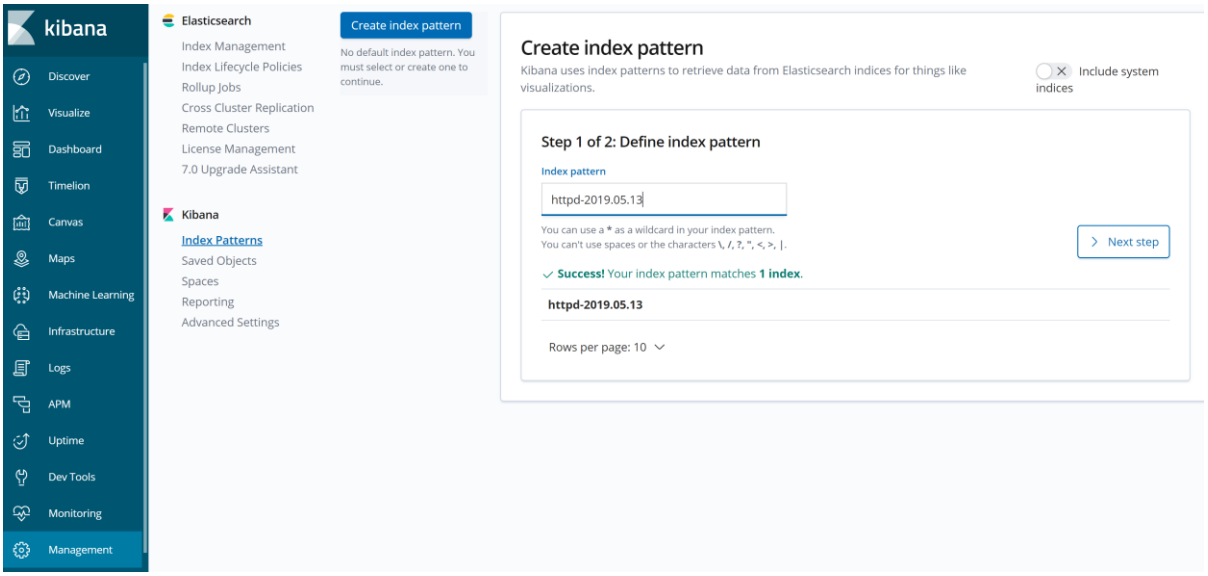


Figura 17 - criação do index do kibana para o httpd

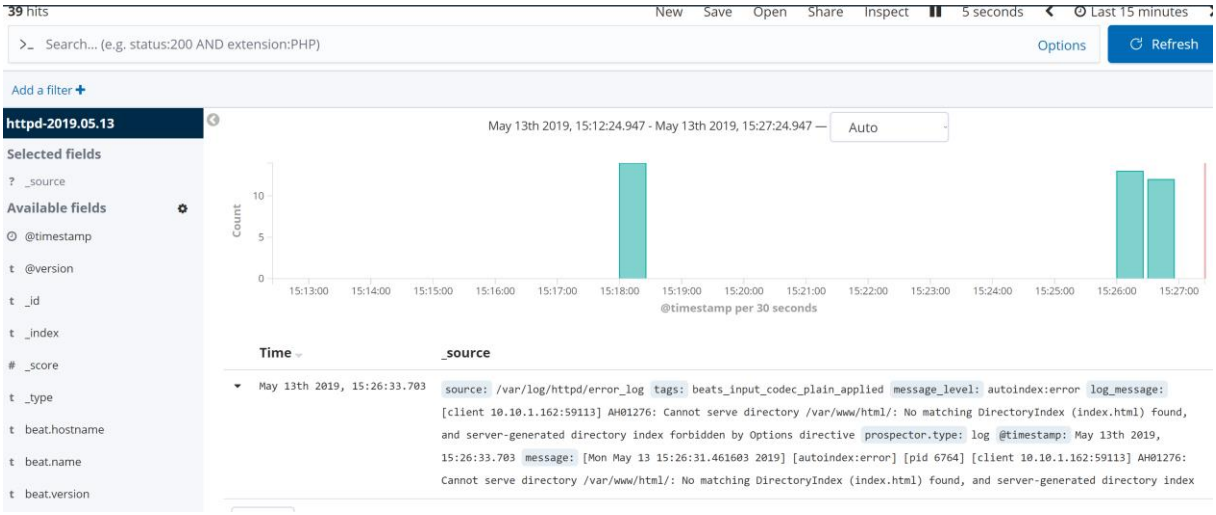


Figura 18 - Discover index httpd.

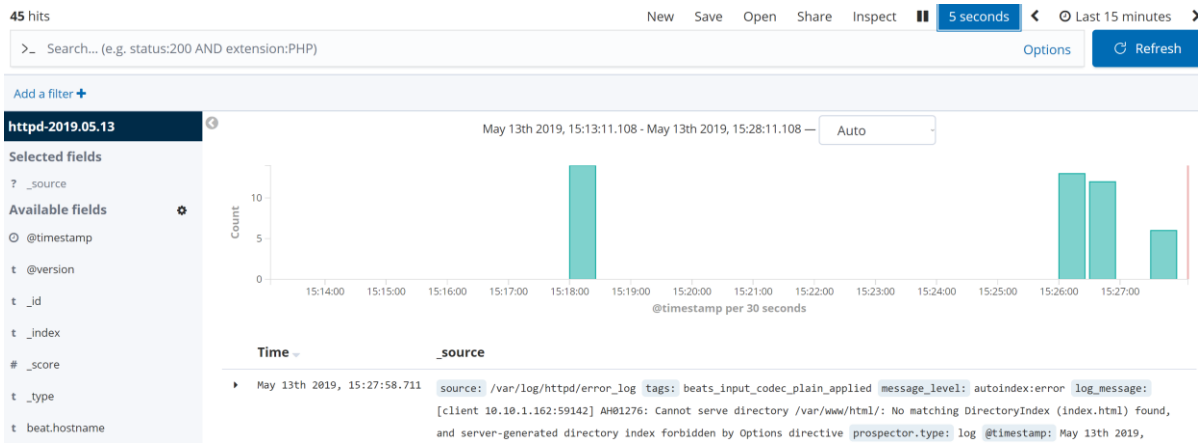


Figura 19 - Discover após consulta da página httpd

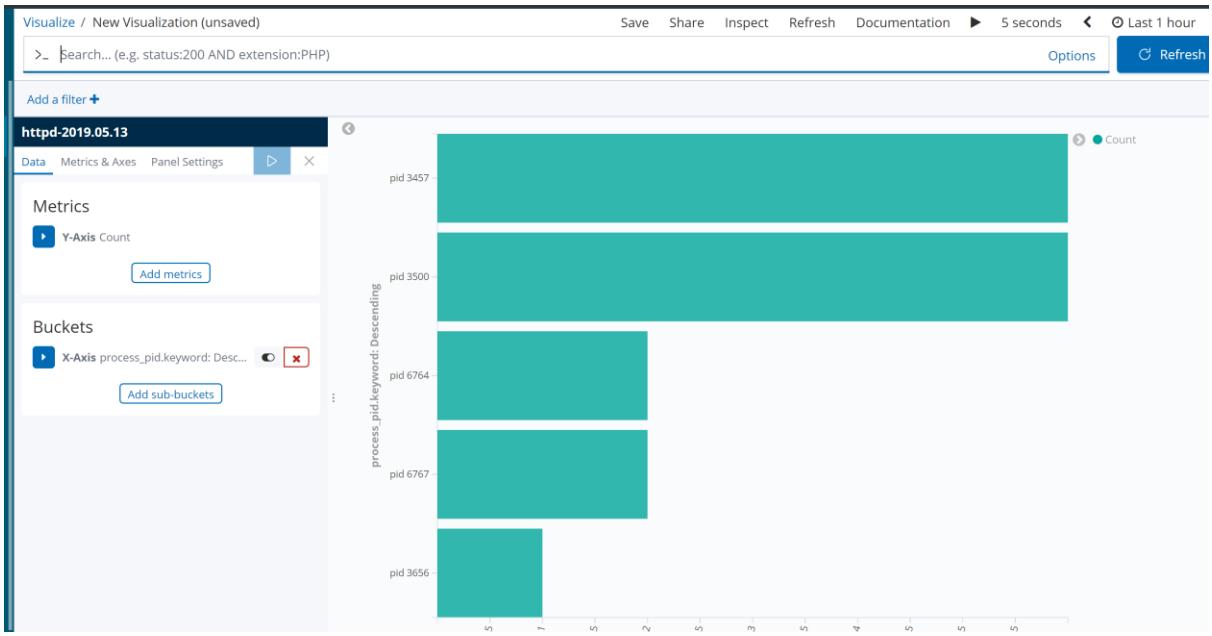


Figura 20 – Dashboard httpd.

4.1.3. Apresentação de *haproxy* do sistema no *kibana*.

De modo a proceder à análise dos *logs* e funcionamento dos *sites* do Grupo Visabeira foi necessário examinar o ficheiro de *logs* produzido pelo *haproxy*. Por questões de confidencialidade dos dados, não serão apresentados neste relatório os *dashboards* com as estatísticas. Contudo, foi-nos disponibilizado um ficheiro destes *logs* e testadas todas as possibilidades para que o *logstash* conseguisse ler a totalidade das linhas do ficheiro sem erros e identificar corretamente os respetivos campos.

Nesta fase de desenvolvimento do projeto foi necessário instalar o programa ***cerebro*** 0.8.1, para gerir os indexes do ***elasticsearch*** e gerar o ***template*** do ***geoip*** deste, para se poder representar as localizações (cidade) de onde foram originados. Nas versões abaixo do ***elasticsearch*** 7.0 é preciso instalar o *plugin ingest-geoip* e atualizar a base de dados com a informação dos ips correspondentes à sua cidade e localização. A explicação dos parâmetros presentes nos logs do *haproxy* pode ser consultada em [3].

4.1.4. Apresentação de *webmail* do sistema no *kibana*.

Relativamente à gestão dos *logs* do servidor de *e-mail* da Visabeira, foi também analisado o respetivo ficheiro (igualmente gerado pelo *haproxy* que controla o arranque e o funcionamento do referido servidor). Tal como na secção anterior, não serão apresentados *dashboards* devido à confidencialidade dos dados. De qualquer forma, foram testadas todas as possibilidades que ocorrem nos ficheiros disponibilizados, garantindo assim que *logstash* consiga ler as linhas de *logs* e identificar os campos presentes sem apresentar qualquer erro.

4.1.5. Apresentação de *Microsoft Exchange* do sistema no *kibana*.

Quanto ao *logs* relativos ao envio de *e-mails*, tarefa a cargo do *Microsoft Exchange*, foram também disponibilizados os respetivos ficheiros. Foi feita a análise destes ficheiros e adaptada a nossa aplicação para que o *logstash* conseguisse ler e interpretar corretamente a totalidade dos *logs*. A explicação dos parâmetros presentes nos *logs* do *Microsoft Exchange* pode ser consultada em *https://docs.microsoft.com/en-us/exchange/mail-flow/transport-logs/message-tracking?view=exchserver-2019#structure-of-the-message-tracking-log-files.*

5. Conclusão

A realização deste trabalho permitiu o contacto com novos horizontes e o mundo real. De realçar também o contacto e a familiarização com poderosas ferramentas *opensource*, bem como a perceção das razões que conduzem à necessidade das empresas fazerem a análise dos *logs*.

O projeto permite adaptar-se facilmente a qualquer tipo de serviços, é relativamente rápido, podemos usar vários serviços ao mesmo tempo para monitorizar. A única parte difícil deste projeto foram os *groks*, e se por ventura alguém quiser implementar esta solução vai conseguir uma boa gestão dos seus serviços da empresa. Há sempre atualizações do código *opensource* e o site principal do *elk* ensina como instalar e tem palestras gratuitas sobre como este código funciona. A experiência que o meu orientador do Grupo da Visabeira exclama é de que este *software* ajudou e muito para o funcionamento da empresa.

Apesar de algumas dificuldades iniciais, nomeadamente em perceber o funcionamento do *elk*, rapidamente conseguimos descortinar os objetivos deste projeto e avançar na sua concretização.

Sem a ajuda do meu colega do curso a realizar este projeto não estaria tão bem estruturado e desenvolvido.

Durante o decorrer do estágio, o meu orientador mostrou-se excecional na ajuda e no incentivo que me dava, e também se mostrou disponível para ajudar na realização do projeto.

Em suma, gostei imenso de realizar este projeto na entidade acolhedora GrupoVisabeira.

6. Referências

[1] Grupo Visabeira - PERFIL DO GRUPO. [Em linha]. Grupo Visabeira. [Consult. 13 Jun. 2019] Disponível em: <https://grupovisabeira.com/pt/o-grupo>

[2] Elasticsearch B.V. - Input plugins. [Em linha]. Elasticsearch B.V. [Consult. 13 Jun. 2019] Disponível em: <https://www.elastic.co/guide/en/logstash/current/input-plugins.html>

[3] Exceliance - ALOHA Load-Balancer. [Em linha]. Documento público. [Consult. 13 Jun. 2019] Disponível em: <https://cdn.haproxy.com/wp-content/uploads/2017/07/aloha_load_balancer_memo_log.pdf>.

.

7. Bibliografia

https://www.elastic.co/elk-stack (elastic info)

https://www.elastic.co/blog/get-system-logs-and-metrics-into-elasticsearch-with-beats-system-modules (system modules info)

https://www.digitalocean.com/community/tutorials/how-to-install-elasticsearch-logstash-and-kibana-elastic-stack-on-centos-7 (install elk)

https://docs.microsoft.com/en-us/exchange/mail-flow/transport-logs/message-tracking?view=exchserver-2019#structure-of-the-message-tracking-log-files

(exchange helper)

https://www.guru99.com/elk-stack-tutorial.html (info elk)

https://github.com/lmenezes/cerebro (cerebro)

https://cdn.haproxy.com/wp-content/uploads/2017/07/aloha_load_balancer_memo_log.pdf
(haproxy helper)

https://streamsets.com/documentation/datacollector/latest/help/datacollector/UserGuide/Apx-GrokPatterns/GrokPatterns_title.html (grok debugger)

https://www.linode.com/docs/databases/elasticsearch/a-guide-to-elasticsearch-plugins/#user-agent-processor-plugin (plugins elasticsearch)

https://www.elastic.co/webinars/getting-started-elasticsearch?elektra=startpage (info elastic)

https://www.elastic.co/webinars/getting-started-kibana?elektra=startpage (info kibana)

https://www.digitalocean.com/community/tutorials/how-to-use-logstash-and-kibana-to-centralize-logs-on-centos-7 (install elk)

https://www.elastic.co/blog/geoip-in-the-elastic-stack (geoip info)

https://www.linuxtechi.com/install-elk-stack-elasticsearch-logstash-kibana-centos7-rhel7/
(install elk)

https://www.elastic.co/guide/en/logstash/current/config-examples.html (exemplo conf logstash)

https://www.digitalocean.com/community/tutorials/how-to-use-logstash-and-kibana-to-centralize-logs-on-centos-7 (info elk)

https://www.tecmint.com/install-elasticsearch-logstash-and-kibana-elk-stack-on-centos-rhel-7/?fbclid=IwAR0Aw7yAO87SRmzAfFRqnoH9pyotNNA6AXfvOczZtUEozDavoSt1xMdU5qQ (install elk)

https://pplware.sapo.pt/software/elasticsearch-pesquise-e-analise-os-seus-dados-em-tempo-real/ (info elasticsearch)

Anexo A - Casos de Uso

A.1 Caso de uso Adicionar Ficheiro

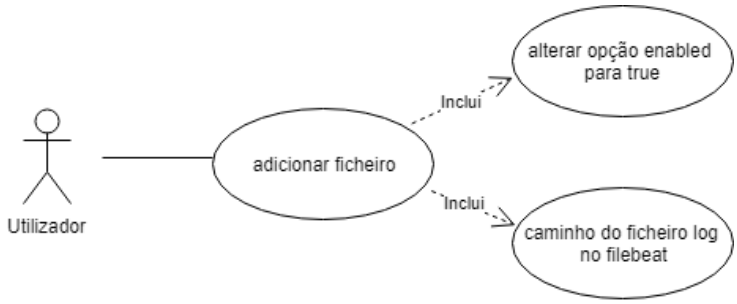


Figura 21 - Casos de uso adicionar ficheiro

Caso de uso: Adicionar ficheiro.

Sumário: O utilizador adiciona um ficheiro.

Clientes: Utilizador

Pré-condição: O utilizador já tem de estar nas configurações do ***filebeat***.

Fluxo principal:

1- O caso de uso começa quando o utilizador tiver nas configurações do ***filebeat***
2- O utilizador altera a opção ***enabled*** para ***true*** se pretender usar um ficheiro específico para fazer grok. Caso não mude esta opção, passa para o fluxo alternativo 1.
3- O utilizador na opção ***Path*** insere o respetivo caminho para o ficheiro log que quer. Caso não insira esta opção, passa para o fluxo alternativo 1.

Fluxo alternativo:

1- A informação do ficheiro não aparece no ***kibana*** e nem a criação do índex do ***elasticsearch*** não é realizada.

Pós-condição:

O ficheiro de configuração tem de ser guardado e o serviço ***filebeat*** reiniciado para que as respetivas alterações tenham efeito.

A.2 Caso de uso configuração do logstash

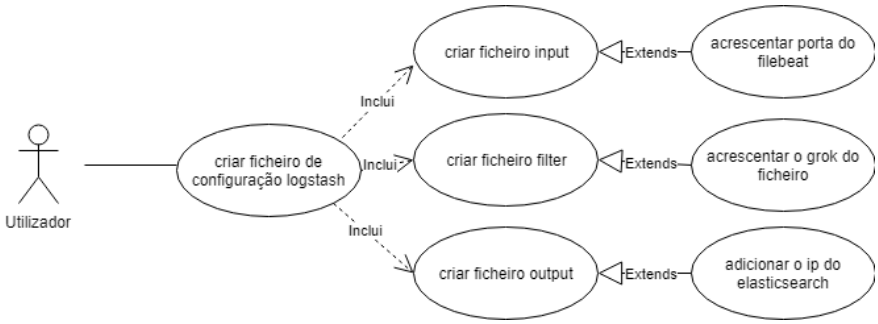


Figura 22 - Caso de uso configuração do logstash

Caso de uso: Configurar o ***logstash***

Sumário: O utilizador faz configurações do ***logstash***.

Clientes: Utilizador

Pré-condição: O utilizador já tem de estar na diretoria "/etc/logstash/conf.d" .

Fluxo principal:

1- O caso de uso começa quando o utilizador cria o ficheiro input.conf no caminho do ***logstah***.
2- O utilizador cria um ficheiro de input.conf onde especifica a entrada para o filebeat. (E1)
3- O utilizador faz o grok do ficheiro e insere o mesmo na configuração input. Se grok não estiver correto ou falhar, tag ***_grokparsefailure*** no ***kibana***. (E2)
4- O utilizador insere as configurações do output e o nome do index. (E3)

Fluxo alternativo:

1- O ***filebeat*** não lê o conteúdo do ficheiro, fazendo com que, não haja representação gráfica dos dados no ***kibana***.
2- O index não é criado por causa do grok. Consultar logs (/var/log/*) do *filebeat e do *logstash para ver a razão.
3- O(s) ip(s) inseridos são incorretos, ou seja, não pertencem à rede da máquina.

Pós-condição:

O ficheiro de configuração tem de ser guardado e o serviço ***filebeat*** reiniciado para que as respetivas alterações tenham efeito.

A.3 Caso de uso adicionar ip

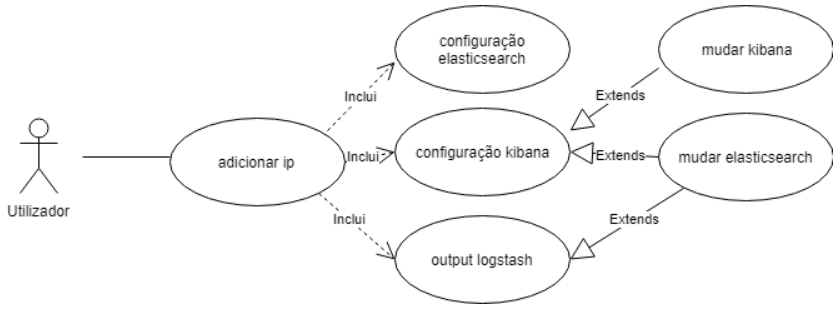


Figura 23 - Casos de uso adicionar ip

Caso de uso: Adicionar o ip

Sumário: O utilizador adiciona o ip no ***kibana***, ***elasticsearch*** e ***logsash***.

Clientes: Utilizador

Pré-condição: O utilizador tem de saber o seu ip.

Fluxo principal:

1. O utilizador altera o ip do ***kibana*** e do ***elasticsearch*** na configuração do ***kibana*** (/etc/kibana/kibana.yml).

2. O utilizador altera o ip do ***elasticsearch*** na sua respetiva configuração (/etc/elasticsearch/elasticsearch.yml).
3. O utilizador altera o ip que está presente no ficheiro output do ***logstash***.

Pós-condição:

Os ficheiros de configurações têm de ser guardados e os serviços reiniciados para que as respetivas alterações tenham efeito.

A.4 Caso de uso gestão do index do elasticsearch

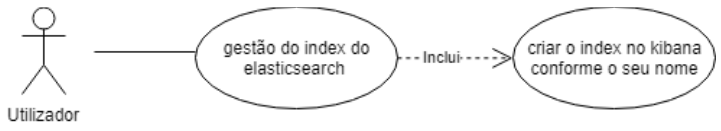


Figura 24 - Caso de uso gestão do index do elasticsearch

Caso de uso: Gerir os indexes do ***elasticsearch***

Sumário: O utilizador gere o index no ***kibana***.

Clientes: Utilizador

Pré-condição: O utilizador tem de estar na opção settings do ***kibana***.

Fluxo principal:

1. O utilizador espera que o index seja criado pelo o output do ***elasticsearch***. (E1) (S1)

Fluxo alternativo:

1- O index não é criado por causa do grok.

Subfluxo alternativo:

1- Consultar logs (/var/log/*) do *filebeat e do *logstash para ver a razão.

Pós-condição:

Selecionar ***Time Filter field name***.

A.5 Caso de uso adicionar dashboard no kibana

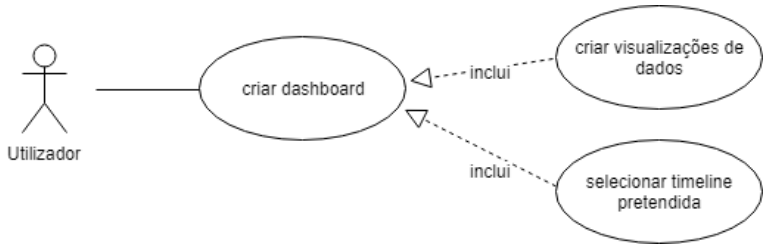


Figura 25 - Caso de uso adicionar do dashboard

Caso de uso: Adicionar do dashboard

Sumário: O utilizador cria o dashboard no ***kibana***.

Clientes: Utilizador

Pré-condição: O utilizador tem de estar na opção dashboard do ***kibana***.

Fluxo principal:

1. O utilizador cria as visualizações de dados que pretender para o seu dashboard. (E1, E2, S1)

Fluxo alternativo:

1. Os campos pretendidos não existem.
2. O formato do campo não é o pretendido.

Subfluxo:

1. Selecionar timeline pretendida.

Pós-condição:

Selecionar guardar.