



# Análisis de datos: De Excel a ELK

**XIX Foro Tecnológico de Empleo**





# ¿Quiénes somos?



**Vítor Fernández Díaz**  
Ingeniero Informático  
@vfdiaz



**Dani Rodríguez Domínguez**  
Ingeniero Telecomunicaciones

## 01 Motivación

“¿Ha fallado? Sácame una gráfica con los errores cada hora”

## 02 ELK Stack

Un poco de teoría...

## 03 ELK en acción

Porque no nos gusta Excel y somos vagos, ELK

## 04 ¿Y si vamos al cloud?

Experimento conjunto :-)



# Preparación



# Antes de empezar...

**Java 8 - Necesario:** para último ejercicio

**Git + GitBash:** Opcional. En su defecto necesario entorno linux (o cygwin o similar)

Descargar Laboratorio de <https://github.com/OptareSolutions/forotecnoloxico2019>

**Docker** - Opcional

Accedemos a <http://foro-tec.siaws.optaresolutions.com:5601/app/kibana#/>



# Motivación」



## Log de aplicación

*“En informática, se usa el término log, historial de log o registro a la **grabación secuencial en un archivo o en una base de datos de todos los acontecimientos (eventos o acciones)** que afectan a un proceso particular (aplicación, actividad de una red informática, etc.). De esta forma constituye una **evidencia del comportamiento del sistema**”*

<https://github.com/OptareSolutions/forotecnoloxico2019>

```
vfernandez@CHOURENSE MINGW64 /c/code/MMGIT/foroteleco2019/lab01_bash_excel (master)
$ head -n 20 sample.log
0.0.0.0 - - [28/Feb/2019:00:00:05 +0100] "PUT /domicilios/2541c6e4-1a72-36d4-bac8-a31083d0a9ab HTTP/1.1" 200 527 107 106 http-nio-8080-e/2e8-386
0.0.0.0 - - [28/Feb/2019:00:00:08 +0100] "PUT /cliente/xce10c65-6c64-3ac2-8ee8-379d8399983c HTTP/1.1" 200 1072 269 268 http-nio-8080-e/2e8-393
0.0.0.0 - - [28/Feb/2019:00:00:11 +0100] "PUT /domicilios/2541c6e4-1a72-36d4-bac8-a31083d0a9ab HTTP/1.1" 200 527 128 128 http-nio-8080-e/2e8-371
0.0.0.0 - - [28/Feb/2019:00:00:21 +0100] "PUT /cliente/x7f882c2-c999-3439-a9f3-6e0dbaa53822 HTTP/1.1" 201 1073 145 145 http-nio-8080-e/2e8-360
0.0.0.0 - - [28/Feb/2019:00:01:04 +0100] "POST /ordenProcesos HTTP/1.1" 201 809 90 90 http-nio-8080-e/2e8-246
0.0.0.0 - - [28/Feb/2019:00:01:09 +0100] "GET /servicioCliente/25eb2dc8-efc6-3162-b6b1-8686261f92b4/servicioRef HTTP/1.1" 200 5916 29 28 http-nio-80
0.0.0.0 - - [28/Feb/2019:00:01:17 +0100] "GET /servicioCliente/25eb2dc8-efc6-3162-b6b1-8686261f92b4 HTTP/1.1" 200 3383 18 18 http-nio-8080-e/2e8-294
0.0.0.0 - - [28/Feb/2019:00:01:18 +0100] "POST /ordenProcesos HTTP/1.1" 201 809 33 33 http-nio-8080-e/2e8-376
0.0.0.0 - - [28/Feb/2019:00:01:20 +0100] "GET /servicioOrdenes/22e31a25-6780-3398-ac88-404ea21060ff HTTP/1.1" 500 1798 62 61 http-nio-8080-e/2e8-393
0.0.0.0 - - [28/Feb/2019:00:01:20 +0100] "POST /ordenProcesos HTTP/1.1" 201 809 64 63 http-nio-8080-e/2e8-386
0.0.0.0 - - [28/Feb/2019:00:01:23 +0100] "GET /servicioRecurso/ace22c33-e5a8-3895-90f8-fc8f0a6824a3 HTTP/1.1" 200 5919 33 33 http-nio-8080-e/2e8-349
0.0.0.0 - - [28/Feb/2019:00:01:23 +0100] "GET /servicioRecurso/ff2e62b2-e330-3b21-b5d5-20bb361810e1 HTTP/1.1" 200 6282 31 31 http-nio-8080-e/2e8-389
0.0.0.0 - - [28/Feb/2019:00:01:25 +0100] "GET /servicioOrdenes/881f5289-2ed8-3ce8-a008-dfb4fac504f1 HTTP/1.1" 200 1622 41 40 http-nio-8080-e/2e8-195
0.0.0.0 - - [28/Feb/2019:00:01:29 +0100] "PATCH /servicioOrdenes/x21de390-6422-3813-ab31-ba96f753b453 HTTP/1.1" 200 1625 39 39 http-nio-8080-e/2e8-3
0.0.0.0 - - [28/Feb/2019:00:01:31 +0100] "POST /ordenProcesa/asignaServicioOrden?id=10131868-6dd1-3a59-9626-077eb9fc159c HTTP/1.1" 200 22 80 80 http
0.0.0.0 - - [28/Feb/2019:00:01:35 +0100] "GET /servicioOrdenes/a7713838-b0e8-3638-9d78-f89d64bf2590 HTTP/1.1" 200 1625 32 31 http-nio-8080-e/2e8-348
0.0.0.0 - - [28/Feb/2019:00:01:36 +0100] "GET /servicioOrdenes/xaf32f68-13d1-3063-b039-89ccbaa96725/contenidoPor HTTP/1.1" 200 1920 43 42 http-nio-8
0.0.0.0 - - [28/Feb/2019:00:01:36 +0100] "GET /servicioOrdenes/xf6575b5-f405-3a68-83f8-c6667ee42c50 HTTP/1.1" 200 1803 85 85 http-nio-8080-e/2e8-385
0.0.0.0 - - [28/Feb/2019:00:01:37 +0100] "GET /servicioRecurso/d707d2a5-2724-3978-af22-9fc0ce415077 HTTP/1.1" 200 5949 43 42 http-nio-8080-e/2e8-337
0.0.0.0 - - [28/Feb/2019:00:01:38 +0100] "GET /servicioOrdenes/9e996b48-5308-3988-9cd4-ad4d27f507cd HTTP/1.1" 200 1614 30 30 http-nio-8080-e/2e8-375
```



## Bash: Herramienta básica

Cuando estás solo ante un terminal...

**grep** - Filtrar líneas

**awk** - Seleccionar columna

**sort** - Ordenar

**wc** - Contar líneas

**uniq** - Agrupar

Vamos al lab01\_bash\_excel...



## Formato

```
vfernandez@CHOURENSE MINGW64 /c/code/MMGIT/foroteleco2019/lab01_bash_excel (master)
$ grep "HTTP/1.1\" 50\" sample.log
0.0.0.0 - - [28/Feb/2019:00:01:20 +0100] "GET /servicioOrdenes/22e31a25-6780-3398-ac88-404ea21060ff HTTP/1.1" 500 1798 62 61 http-nio-8080-e/2e8-393
0.0.0.0 - - [28/Feb/2019:02:02:18 +0100] "GET /servicioCliente/a7b8a2e0-6512-3e38-b853-380379bf2314/servicioDefRef HTTP/1.1" 500 1213 8 7 http-nio-8080-e/2e8-322
0.0.0.0 - - [28/Feb/2019:02:02:18 +0100] "GET /servicioCliente/adcc1919-41c0-3103-8258-dbdd5588d5f0/servicioRef HTTP/1.1" 500 5614 28 28 http-nio-8080-e/2e8-246
0.0.0.0 - - [28/Feb/2019:02:02:19 +0100] "GET /servicioCliente/25203ce8-42a5-3df0-a6a5-8c427dc298a2/servicioRef HTTP/1.1" 500 5614 29 29 http-nio-8080-e/2e8-375
0.0.0.0 - - [28/Feb/2019:02:02:19 +0100] "GET /servicioCliente/86b92cc8-9362-3666-bb38-d055736febcb/servicioDefRef HTTP/1.1" 500 1213 8 8 http-nio-8080-e/2e8-337
0.0.0.0 - - [28/Feb/2019:09:00:06 +0100] "GET /servicioCliente/a7e78f88-def0-3098-8d78-a42b78d52950/servicioDefRef HTTP/1.1" 500 1498 9 8 http-nio-8080-e/2e8-406
0.0.0.0 - - [28/Feb/2019:14:19:43 +0100] "PATCH /servicioOrdenes/ccbc4609-cc49-31d8-be03-e8a2605ba63e HTTP/1.1" 500 1614 46 46 http-nio-8080-e/2e8-411
0.0.0.0 - - [28/Feb/2019:15:06:49 +0100] "GET /servicioRecurso/x9db09a8-9cb4-3699-bdb0-2f9add309f8a/servicioDefRef HTTP/1.1" 500 1327 11 11 http-nio-8080-e/2e8-375
0.0.0.0 - - [28/Feb/2019:18:05:15 +0100] "GET /servicioOrdenes/x80d6783-caf4-31a8-8b82-6f14c5ec853e/contenidoPor HTTP/1.1" 500 1924 40 39 http-nio-8080-e/2e8-380
```

IP

Fecha y hora

Método HTTP

URL

Protocolo Respuesta

Tiempo respuesta



## Reto: ¿Cuántos errores (HTTP 50x) ocurrieron por hora?

Filtramos los errores 50x

**grep** - Filtrar líneas

**awk** - Seleccionar columna

**sort** - Ordenar

**wc** - Contar líneas

**uniq** - Agrupar

```
$ grep "HTTP/1.1\" 50" sample.log
```

```
vfernandez@CHOURENSE MINGW64 /c/code/MMGIT/foroteleco2019/lab01_bash_excel (master)
$ grep "HTTP/1.1\" 50" sample.log
0.0.0.0 - - [28/Feb/2019:00:01:20 +0100] "GET /servicioOrdenes/22e31a25-6780-3398-ac88-404ea21060ff HTTP/1.1" 500 1798 62 61 http-nio-8080-e/2e8-393
0.0.0.0 - - [28/Feb/2019:02:02:18 +0100] "GET /servicioCliente/a7b8a2e0-6512-3e38-b853-380379bf2314/servicioDefRef HTTP/1.1" 500 1213 8 7 http-nio-8080-e/2e8-322
0.0.0.0 - - [28/Feb/2019:02:02:18 +0100] "GET /servicioCliente/adcc1919-41c0-3103-8258-dbdd5588d5f0/servicioRef HTTP/1.1" 500 5614 28 28 http-nio-8080-e/2e8-246
0.0.0.0 - - [28/Feb/2019:02:02:19 +0100] "GET /servicioCliente/25203ce8-42a5-3df0-a6a5-8c427dc298a2/servicioRef HTTP/1.1" 500 5614 29 29 http-nio-8080-e/2e8-375
0.0.0.0 - - [28/Feb/2019:02:02:19 +0100] "GET /servicioCliente/86b92cc8-9362-3666-bb38-d055736feb7/servicioDefRef HTTP/1.1" 500 1213 8 8 http-nio-8080-e/2e8-337
0.0.0.0 - - [28/Feb/2019:09:00:06 +0100] "GET /servicioCliente/a7e78f88-def0-3098-8d78-a42b78d52950/servicioDefRef HTTP/1.1" 500 1498 9 8 http-nio-8080-e/2e8-406
0.0.0.0 - - [28/Feb/2019:14:19:43 +0100] "PATCH /servicioOrdenes/ccbc4609-cc49-31d8-be03-e8a2605ba63e HTTP/1.1" 500 1614 46 46 http-nio-8080-e/2e8-411
0.0.0.0 - - [28/Feb/2019:15:06:49 +0100] "GET /servicioRecurso/x9db09a8-9cb4-3699-bdb0-2f9add309f8a/servicioDefRef HTTP/1.1" 500 1327 11 11 http-nio-8080-e/2e8-375
0.0.0.0 - - [28/Feb/2019:18:05:15 +0100] "GET /servicioOrdenes/x80d6783-caf4-31a8-8b82-6f14c5ec853e/contenidoPor HTTP/1.1" 500 1924 40 39 http-nio-8080-e/2e8-380
```



## Reto: ¿Cuántos errores (HTTP 50x) ocurrieron por hora?

Extraemos la hora

**grep** - Filtrar líneas

**awk** - Seleccionar columna

**sort** - Ordenar

**wc** - Contar líneas

**uniq** - Agrupar

```
$ grep "HTTP/1.1\" 50" sample.log | awk -F':' '{print $2}'
```

```
vfernandez@CHOURENSE MINGW64 /c/code/MMGIT/foroteleco2019/
$ grep "HTTP/1.1\" 50" sample.log | awk -F':' '{print $2}'
00
02
02
02
02
02
09
14
15
18
```

## Reto: ¿Cuántos errores (HTTP 50x) ocurrieron por hora?

Agrupamos

**grep** - Filtrar líneas

**awk** - Seleccionar columna

**sort** - Ordenar

**wc** - Contar líneas

**uniq** - Agrupar

```
$ grep "HTTP/1.1\" 50" sample.log | awk -F':' '{print $2}' | uniq -c | awk '{print $2,$1}'
```

```
vfernandez@CHOURENSE MINGW64 /c/code/MMGIT/foroteleco2019/lab01_bash_excel (master)
$ grep "HTTP/1.1\" 50" sample.log | awk -F':' '{print $2}' | uniq -c | awk '{print $2,$1}'
00 1
02 4
09 1
14 1
15 1
18 1
```



**Reto:** ¿Cuántos errores (HTTP 50x) ocurrieron por hora?

**Excel:** Ponerlo bonito (en este caso [G Suite](#))

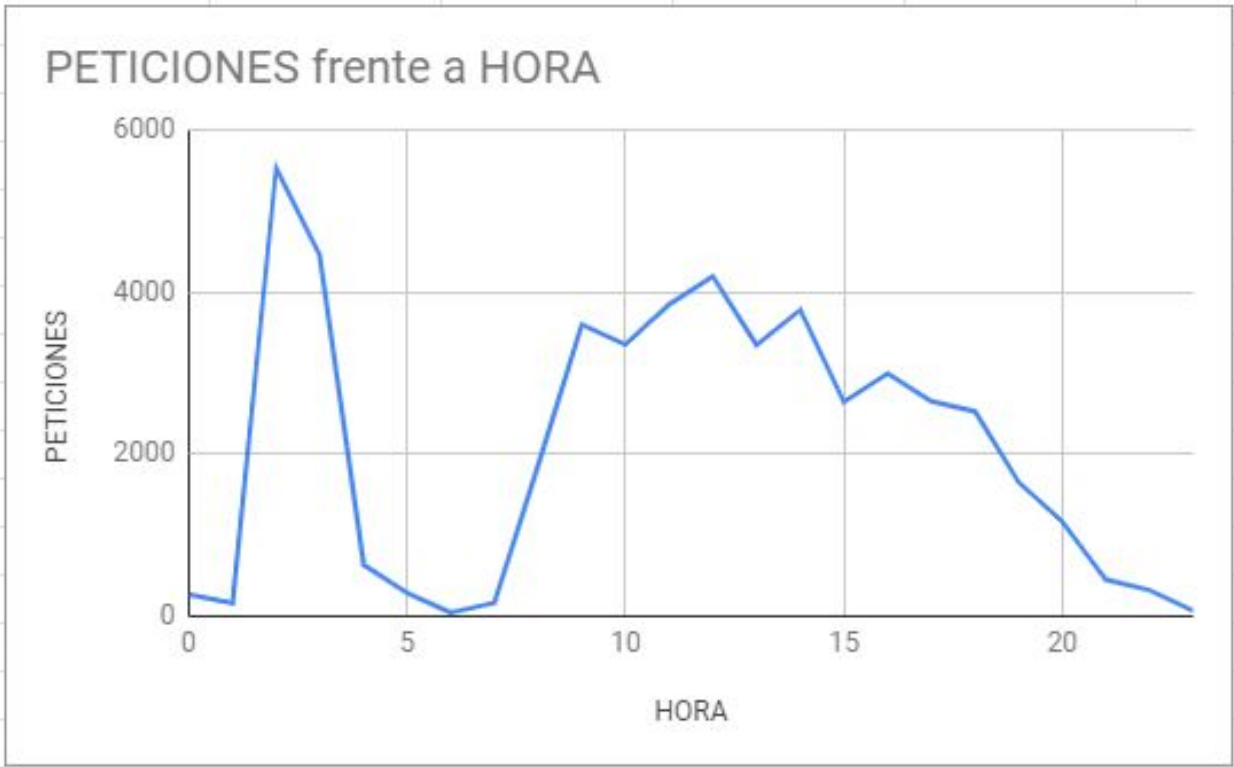


## Más Retos:

- ¿Cuántas peticiones entran por hora?

```
$ cat sample.log | awk -F':' '{print $2}' | uniq -c | awk '{print $2,$1}'
```

HORA	PETICIONES
0	270
1	160
2	5523
3	4456
4	634
5	289
6	42
7	167
8	1856
9	3597
10	3351
11	3847
12	4193
13	3348
14	3777
15	2643
16	2998
17	2651
18	2528
19	1657
20	1171
21	452
22	322
23	68





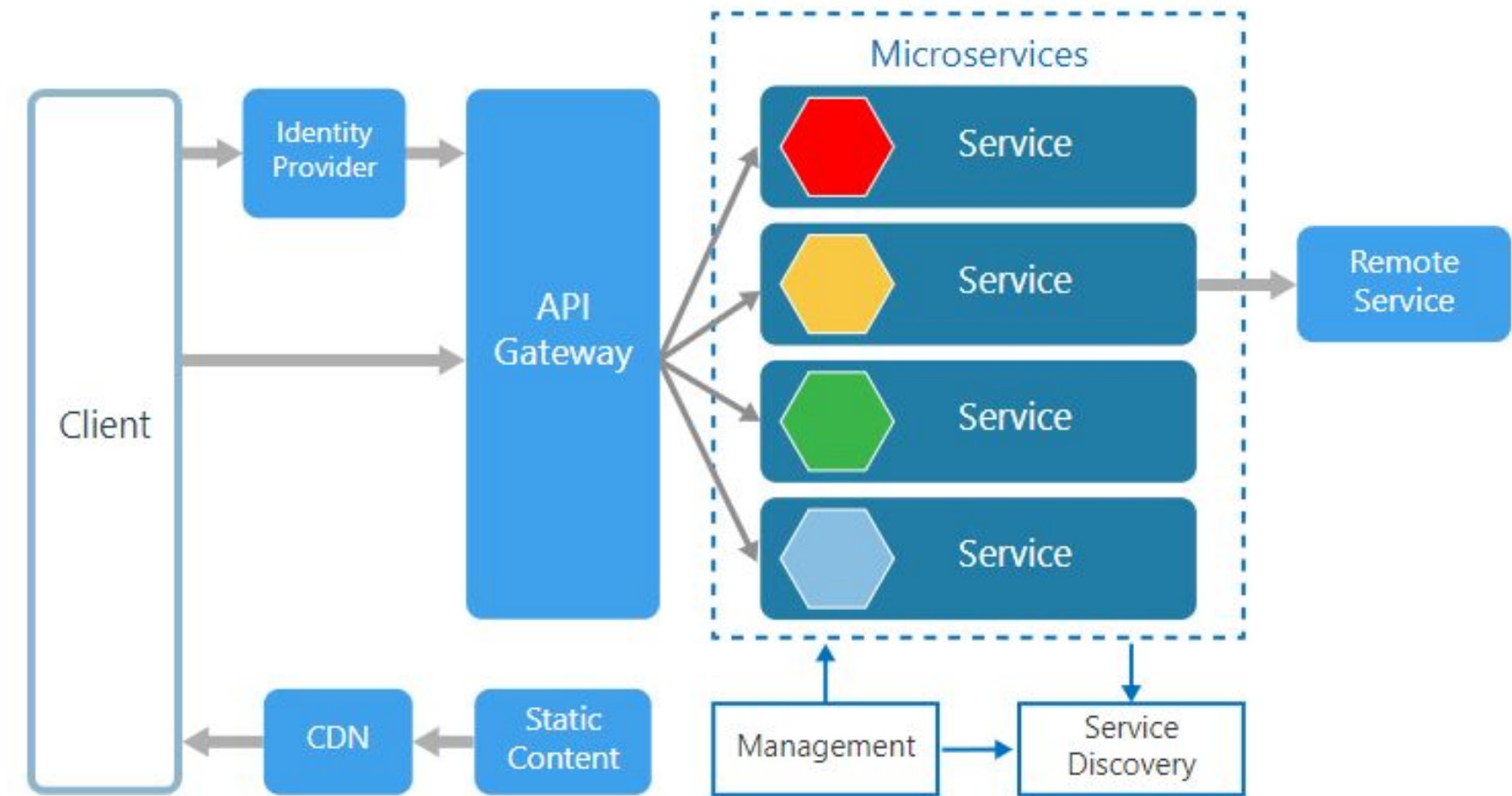
## Más difícil:

- ¿Cuales son los tiempos medios? ¿Por hora?
- ¿Cuales son las operaciones más invocadas? Top 10
- ¿Cual es el porcentaje de OK / KO?
- ¿GETs vs POST?
- ...

# Motivación

## Retos monitorización

- Gran cantidad de información
- Aplicaciones distribuidas
  - Varias aplicaciones
  - Varios nodos
  - Cloud
- Tiempo real



<https://docs.microsoft.com/en-us/azure/architecture/guide/architecture-styles/microservices>

**Resumen: Poder centrarse más en analizar la información, y menos en el *cocinado***



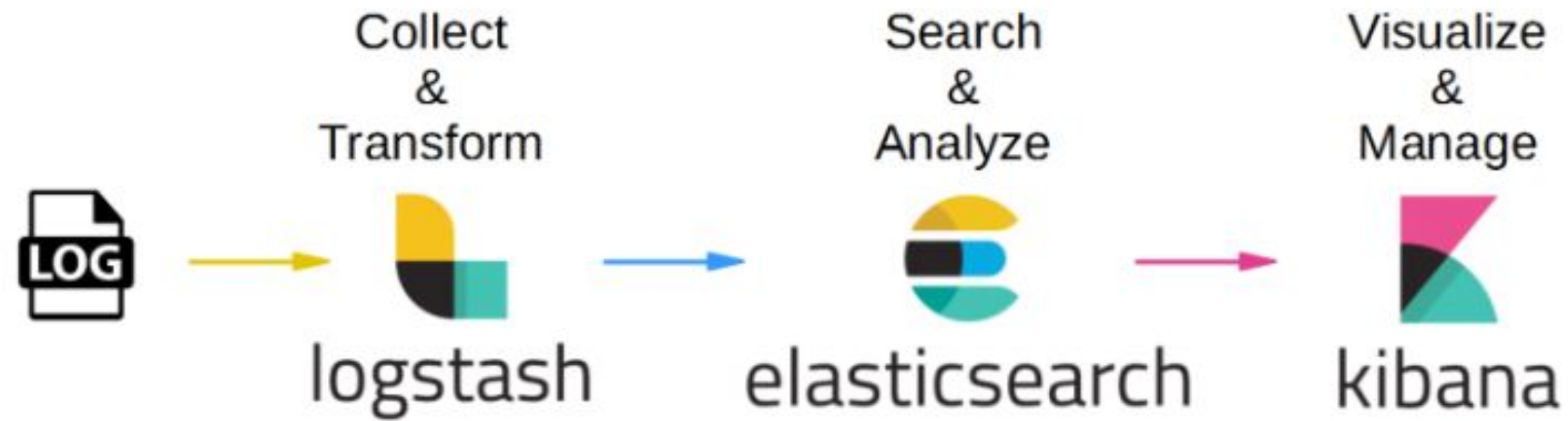


# ELK Stack」



# ELK Stack

¿Qué es ELK?



Log Analysis-Search-Visualize



# ELK Stack

## Logstash

Parseo de log, dejarlo listo para elastic.

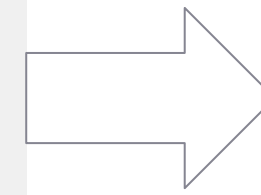
127.0.0.1 - - [11/Dec/2013:00:01:45 -0800] "GET /xampp/status.php HTTP/1.1" 200 3891 "http://cadenza/xampp/navi.php" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10.9; rv:25.0) Gecko/20100101 Firefox/25.0"



```
input { stdin { } }

filter {
  grok {
    match => { "message" => "%{COMBINEDAPACHELOG}" }
  }
  date {
    match => [ "timestamp" , "dd/MMM/yyyy:HH:mm:ss Z" ]
  }
}

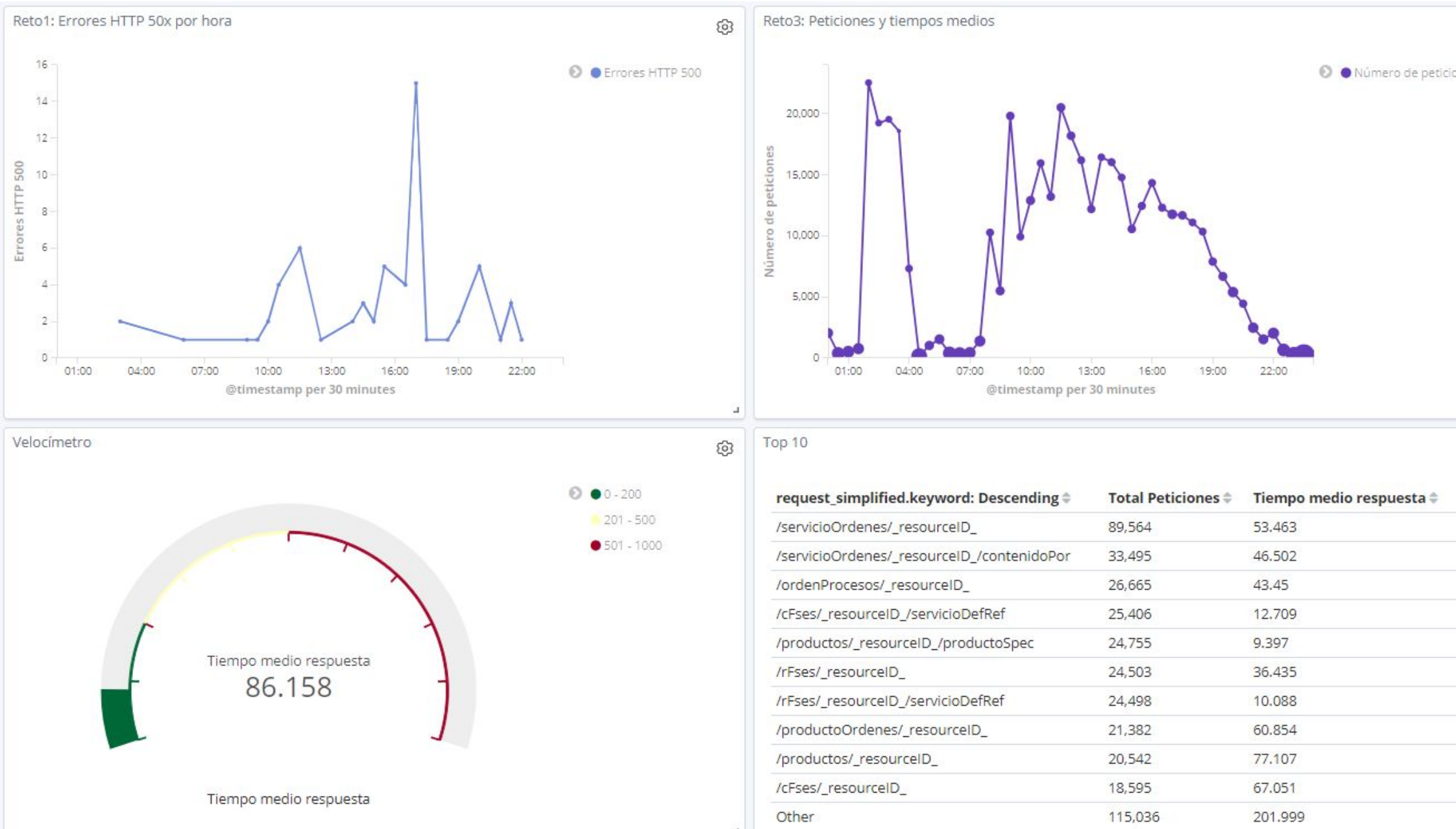
output {
  elasticsearch { hosts => ["localhost:9200"] }
  stdout { codec => rubydebug }
}
```



```
{
  "message" => "127.0.0.1 - - [11/Dec/2013:00:01:45 -0800] \"GET /",
  "@timestamp" => "2013-12-11T08:01:45.000Z",
  "@version" => "1",
  "host" => "cadenza",
  "clientip" => "127.0.0.1",
  "ident" => "-",
  "auth" => "-",
  "timestamp" => "11/Dec/2013:00:01:45 -0800",
  "verb" => "GET",
  "request" => "/xampp/status.php",
  "httpversion" => "1.1",
  "response" => "200",
  "bytes" => "3891",
  "referrer" => "\"http://cadenza/xampp/navi.php\"",
  "agent" => "\"Mozilla/5.0 (Macintosh; Intel Mac OS X 10.9; rv:25.0) Gecko/20100101 Firefox/25.0\"",
}
```

## Elastic & Kibana

### Indexar los datos & Visualización





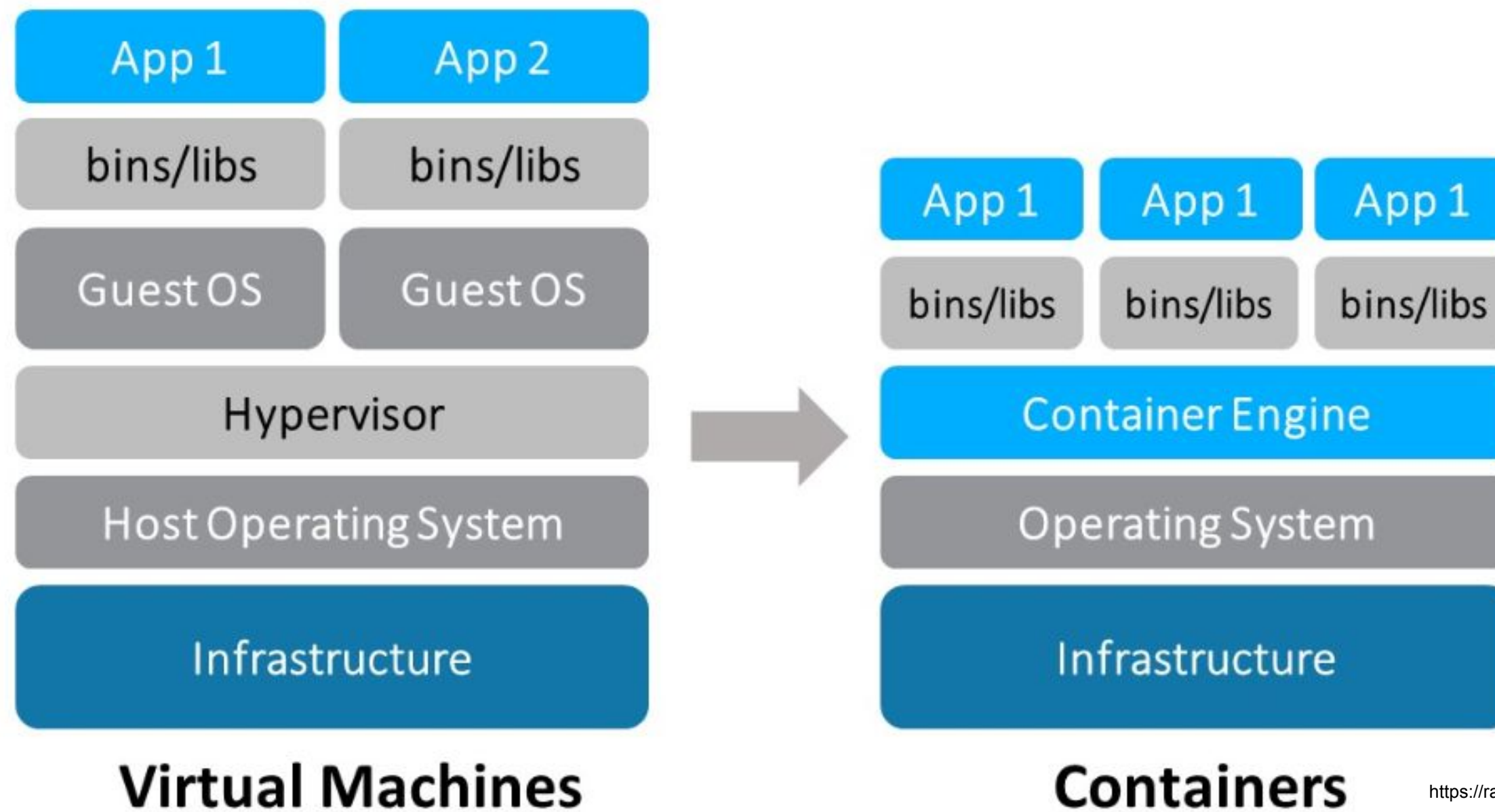
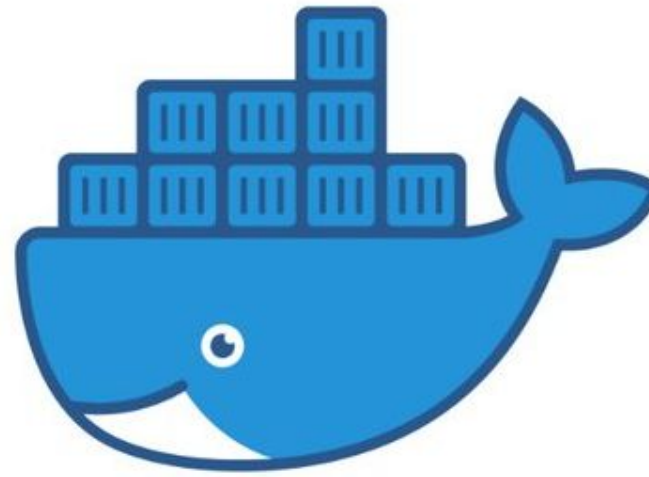


# ELK en Acción

# ELK en Acción

## Docker

Porque somos vagos...





## Levantar contenedor ELK

### 1. Clonar proyecto de GitHub

```
$ git clone https://github.com/OptareSolutions/forotecnologico2019.git
```

```
$ cd forotecnologico2019/lab02_elk
```

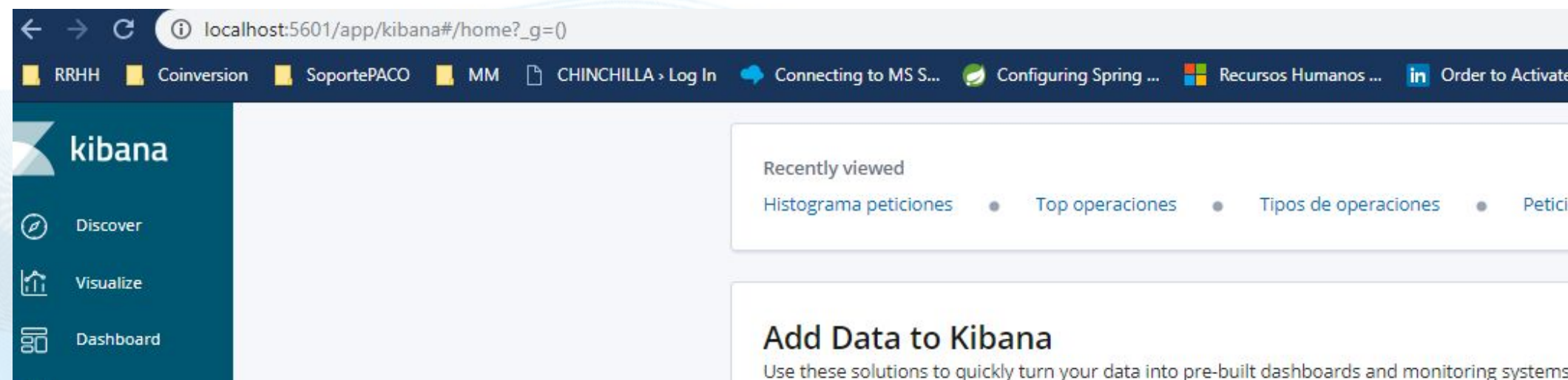
### 2. Construir Imagen docker

```
$ docker build . -t optarelk:1.0
```

### 3. Arrancar contenedor

```
$ docker run -p 5601:5601 -p 9200:9200 -p 5044:5044 -it --name myelk optarelk:1.0
```

### 4. <http://localhost:5601/app/kibana#/>



## Enviar Logs

1. Instalar filebeat ([Descargar](#) y descomprimir)
2. Configurar filebeat para que envíe logs a logstash (filebeat.yml)

```
filebeat.prospectors:  
- input_type: log  
  paths:  
    -  
    C:\Users\vfernandez\Documents\LOGS\input\  
  output.logstash:  
    hosts: ["localhost:5044"]
```

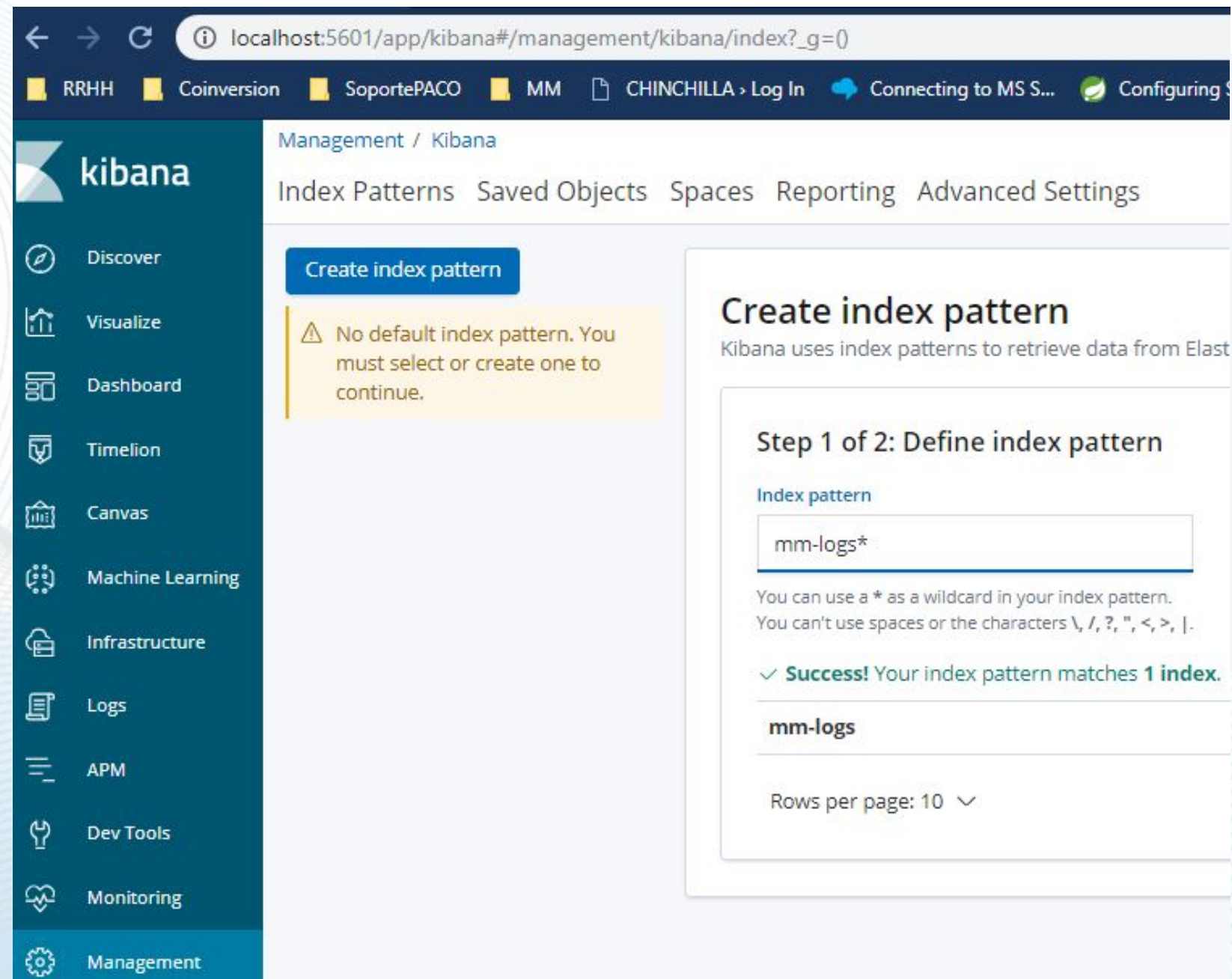
3. Lanzar filebeat

> filebeat -e



## Jugar con Kibana

### 1. Configurar índice



The screenshot shows the Kibana Management interface. The left sidebar contains navigation links: Discover, Visualize, Dashboard, Timelion, Canvas, Machine Learning, Infrastructure, Logs, APM, Dev Tools, Monitoring, and Management. The main content area is titled 'Management / Kibana' and includes tabs for Index Patterns, Saved Objects, Spaces, Reporting, and Advanced Settings. A 'Create index pattern' button is visible. A yellow warning box states: 'No default index pattern. You must select or create one to continue.' The 'Create index pattern' dialog is open, showing 'Step 1 of 2: Define index pattern'. The 'Index pattern' field contains 'mm-logs\*'. A success message says: 'Success! Your index pattern matches 1 index.' Below this, the index 'mm-logs' is listed. The 'Rows per page' is set to 10.

#### Create index pattern

Kibana uses index patterns to retrieve data from Elasticsearch indices for things like visualizations.

☐ Include system indices

#### Step 2 of 2: Configure settings

You've defined **mm-logs\*** as your index pattern. Now you can specify some settings before we create it.

Time Filter field name Refresh

@timestamp

The Time Filter will use this field to filter your data by time.  
You can choose not to have a time field, but you will not be able to narrow down your data by a time range.

[Show advanced options](#)

[< Back](#)

[Create index pattern](#)

## Jugar con Kibana

### 2. Discover

New Save Open Share Inspect Auto-refresh Last 15 minutes

Time Range

Quick Relative Absolute Recent

From

Set To Now

2019-02-28 00:00:00.000

YYYY-MM-DD HH:mm:ss.SSS

To

Set To Now

2019-02-28 23:59:59.999

YYYY-MM-DD HH:mm:ss.SSS

<

February 2019

>

Sun	Mon	Tue	Wed	Thu	Fri	Sat
					01	02
03	04	05	06	07	08	09
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28		

<

February 2019

>

Sun	Mon	Tue	Wed	Thu	Fri	Sat
					01	02
03	04	05	06	07	08	09
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28		

Go

Options

Refresh



# ELK en Acción

## Jugar con Kibana

### 2. Discover

New Save Open Share Inspect Auto-refresh Last 15 minutes

Time Range

Quick Relative Absolute Recent

From 2019-02-28 00:00:00.000 Set To Now To 2019-02-28 23:59:59.999 Set To Now

YYYY-MM-DD HH:mm:ss.SSS YYYY-MM-DD HH:mm:ss.SSS

< February 2019 > < February 2019 >

Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat
					01	02						01	02
03	04	05	06	07	08	09	03	04	05	06	07	08	09
10	11	12	13	14	15	16	10	11	12	13	14	15	16
17	18	19	20	21	22	23	17	18	19	20	21	22	23
24	25	26	27	28			24	25	26	27	28		

Go

Options Refresh

localhost:5601/app/kibana#/discover?\_g=(refreshInterval:(pause:!t,value:0),time:(from:'2019-02-27T23:00:00.000...))

RRHH Coinversion SoportePACO MM CHINCHILLA Log In Connecting to MS S... Configuring Spring ... Recursos Humanos ...

kibana 424,441 hits New Save Open Share Inspect Auto-refresh February 28th 2019, 00:00:00.000 to February 28th 2019, 23:59:59.999

> Search... (e.g. status:200 AND extension:PHP) Options Refresh

Add a filter +

mm-logs\*

Selected fields

? \_source

Available fields

@timestamp

@version

t \_id

t \_index

# \_score

t \_type

t beat.hostname

t beat.name

t beat.version

# bytes

t clientip

t host.name

t httpversion

t message

February 28th 2019, 00:00:00.000 - February 28th 2019, 23:59:59.999 — Auto

Count

@timestamp per 30 minutes

Time \_source

February 28th 2019, 23:59:54.000 verb: GET clientip: 0.0.0.0 query\_params: state=SAVED @version: 1 source: C:\Users\vfernandez\Documents\LOGS\input\sample.log beat.version: 6.4.0 beat.name: CHOURENSE beat.hostname: CHOURENSE request: /ordenActividades/search/findNextProcessableActividades?state=SAVED timestamp: 28/Feb/2019:23:59:54 +0100 time\_process: 5,519 time\_commit: 5,519 host.name: CHOURENSE processor: http-nio-8108-exe8-402

February 28th 2019, 23:59:24.000 verb: GET clientip: 0.0.0.0 @version: 1 source: C:\Users\vfernandez\Documents\LOGS\input\sample.log beat.version: 6.4.0 beat.name: CHOURENSE beat.hostname: CHOURENSE request: /rFses/a8a02593-8218-3896-b778-2502a9c444bc timestamp: 28/Feb/2019:23:59:24 +0100 time\_process: 23 time\_commit: 23 host.name: CHOURENSE processor: http-nio-8108-exe8-395 request\_simplified: /rFses/\_resourceID\_ pre\_url: /rFses/ tags: b

February 28th 2019, 23:59:24.000 verb: PATCH clientip: 0.0.0.0 @version: 1 source: C:\Users\vfernandez\Documents\LOGS\input\sample.log beat.version: 6.4.0 beat.name: CHOURENSE beat.hostname: CHOUR



# ELK en Acción

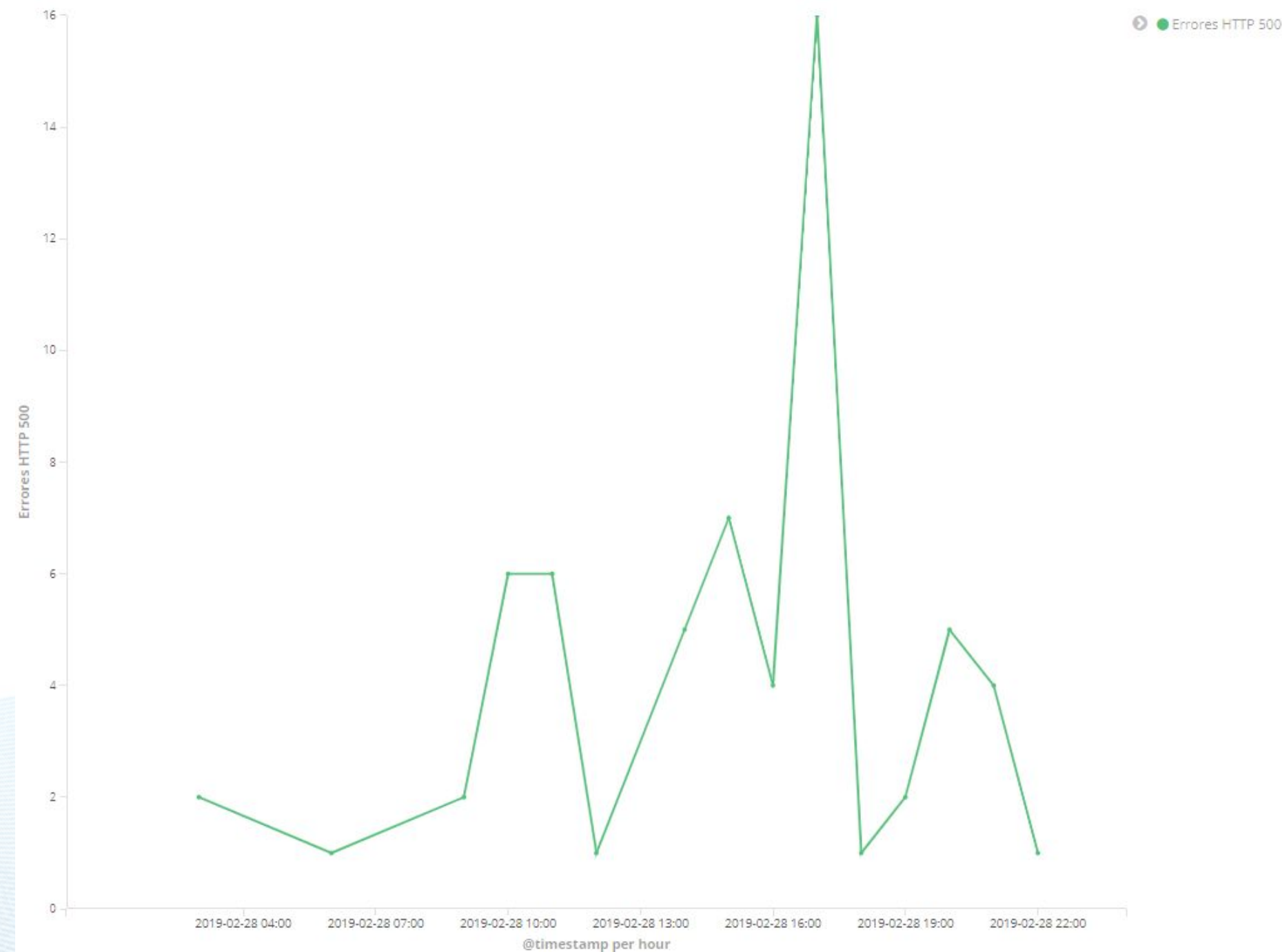


Manos a la obra:

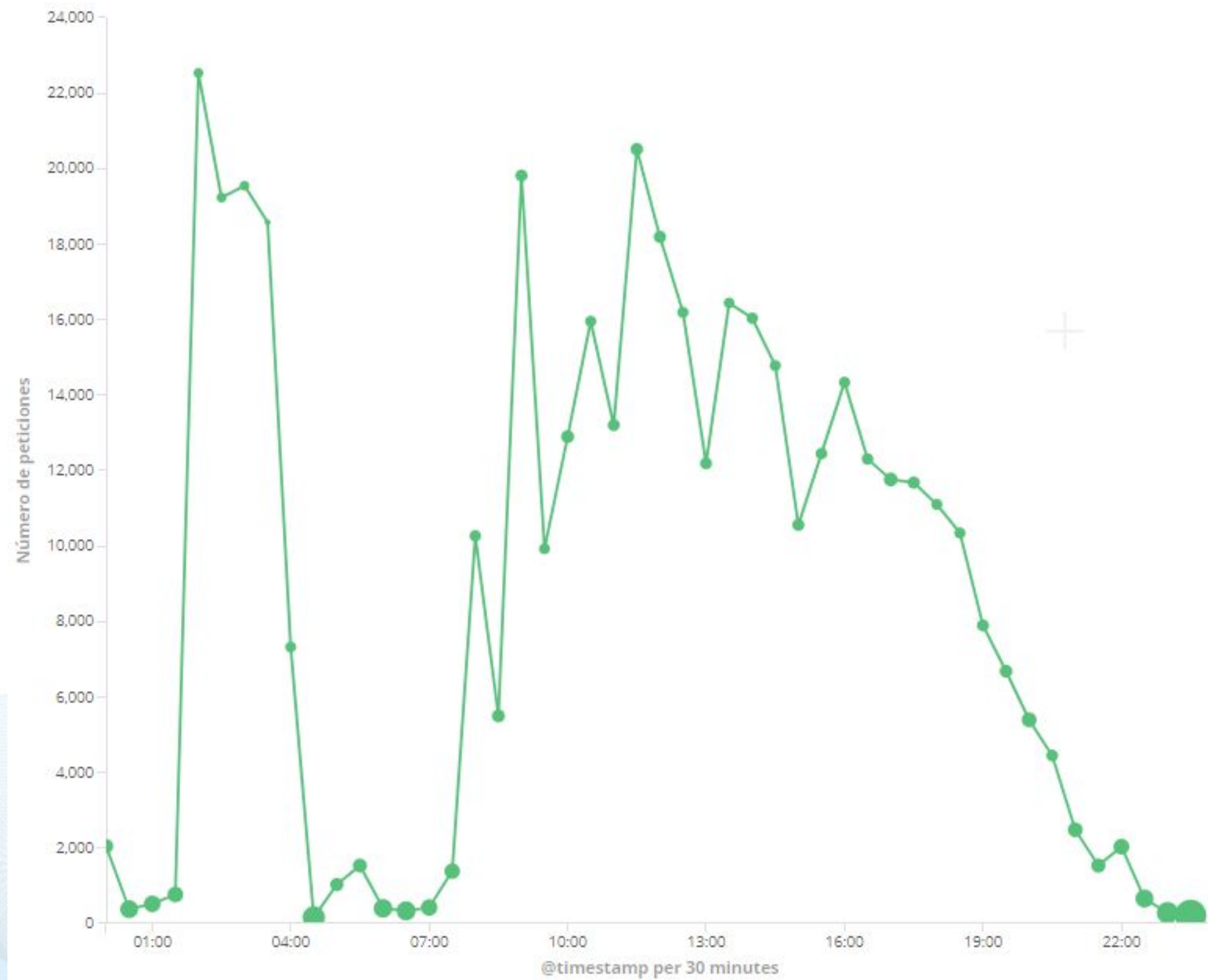
<http://foro-tec.siaaws.optaresolutions.com:5601/app/kibana#/>



## Reto 1: Errores HTTP 50x por hora

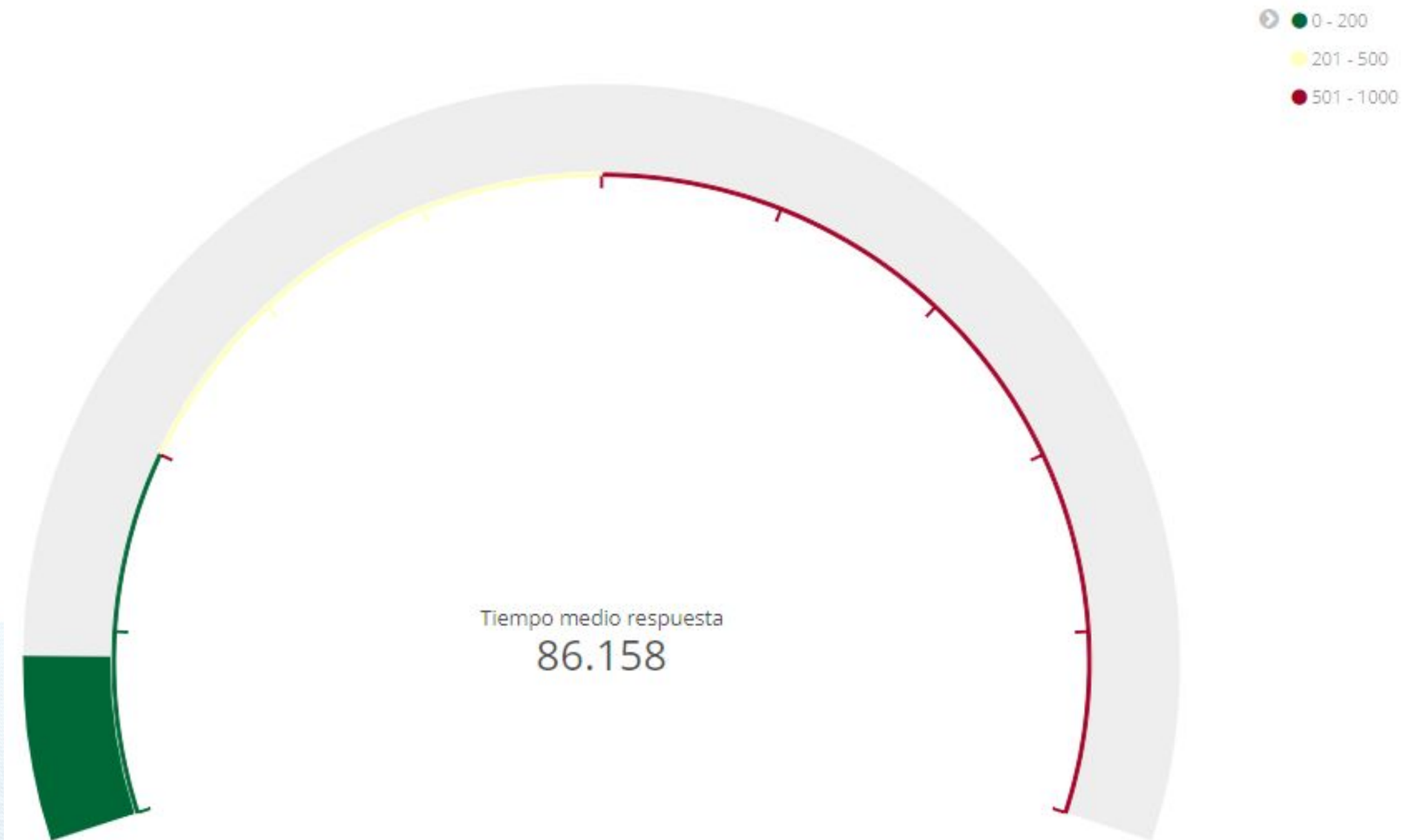


## Reto 2: Peticiones y tiempos de respuesta





## Velocímetro: Tiempo de respuesta



## Top 10 Operaciones

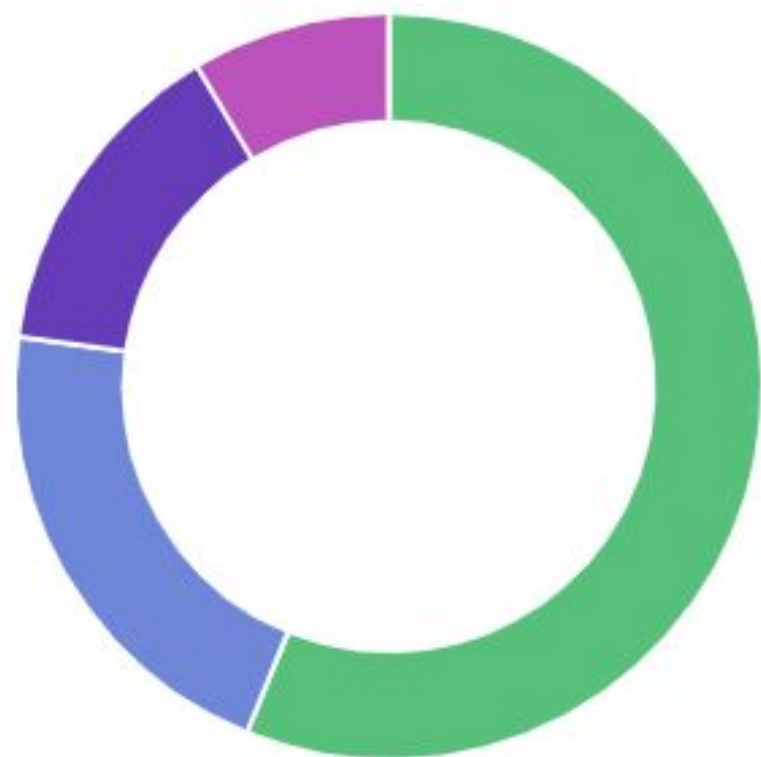
request_simplified.keyword: Descending ↕	Count ↕
/servicioOrdenes/_resourceID_	28,506
/servicioOrdenes/_resourceID_/contenidoPor	10,948
/ordenProcesos/_resourceID_	8,680
/servicioRecurso/_resourceID_/servicioDefRef	8,202
/servicioCliente/_resourceID_/servicioDefRef	7,890
/servicioCliente/_resourceID_	7,868
/servicioRecurso/_resourceID_	5,984
/ordenActividades/_resourceID_	5,804
/ordenProcesos/_resourceID_/ordenProcesSpecRef	4,310
/servicioCliente/_resourceID_/servicioRef	2,928
	<b>91,120</b>





**Experimento Cloud**

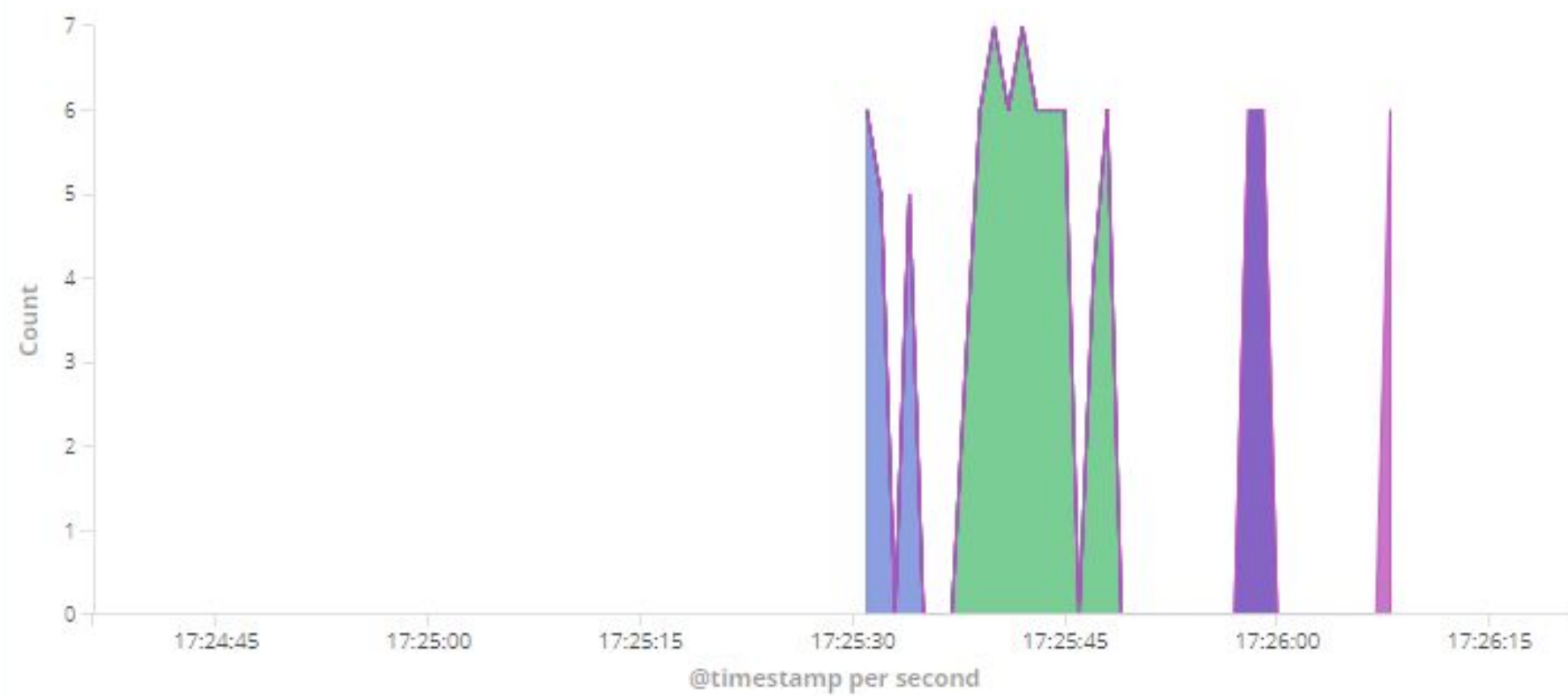
Total



- luis
- juan
- otro
- ultimo



Evolución



- juan
- luis
- otro
- ultimo





# Gracias!

Vítor Fernández | Architecture & Development  
vfernandez@optaresolutions.com

Daniel Rodriguez | Analyst  
drodriguez@optaresolutions.com