

# INFORME DE NEGOCIO Y MULTIMEDIA

GRUPO: MEE6

PROYECTO: "HEARMEE"



Universitat d'Alacant  
Universidad de Alicante

GRADO EN INGENIERÍA MULTIMEDIA.

ABP 2019-2020



# INDICE

1.	INTRODUCCIÓN.....	4
2.	INFOGRAFÍA. REPRESENTACIÓN DE UN NEGOCIO.....	5
2.1	Generación de datos – incorporación de datos en bbdd.....	5
3.	KPI'S.....	6
4.	CMI.....	7
4.1	Mapa estratégico.....	7
5.	GRAFANA.....	8
5.1	Instalación.....	8
5.2	Inclusión en el proyecto.....	8
5.3	Cmi en grafana.....	12
6.	OTROS PROCESOS REALIZADOS RELEVANTES.....	16
6.1	Creación de un script migratorio de datos mongo – mysql.....	16
6.2	Creación y ejecución de un archivo .sh.....	18

## 1. INTRODUCCIÓN.

Negocio y multimedia es mostrar de forma gráfica, clara y concisa el avance de tu proyecto. Gracias a esta parte a desarrollar del proyecto podremos saber en qué aspecto flaquea, o qué parte de él es más llamativa, más interactiva o golosa para el usuario y desarrolladores.

El siguiente informe describirá el trabajo realizado durante los meses en los que esta asignatura se ha introducido en nuestro proyecto. De él se espera haber cumplimentado las exigencias mínimas para superar la asignatura a nivel grupal y haber obtenido los conocimientos esperados para poder aplicarlos en futuros proyectos.

## 2. INFOGRAFÍA. REPRESENTACIÓN DE UN NEGOCIO.

La infografía de HearMee pretende, de la forma más simple, clara y visual, mostrar al usuario (clientes músicos y clientes ofertantes) de qué va nuestro producto, con una breve descripción, qué pretende y se espera de él y hacia donde lo hemos enfocado para conseguir éxito empresarial con él.

La siguiente imagen muestra un ejemplo de infografía. Lamentablemente, debido a que no fue una tarea entregable durante el curso no la conservamos como archivo en nuestros pc's.



### 2.1 Generación de datos – incorporación de datos en bbdd.

Para gestionar la actividad de los usuarios registrados en la app web se requiere de una importación considerable de usuarios interaccionando. Es por ello por lo que se ha optado por crear 10.000 datos sintéticos de ambos roles (artistas y ofertantes).

Los 10.000 datos han sido generados a través del siguiente programa de escritorio:

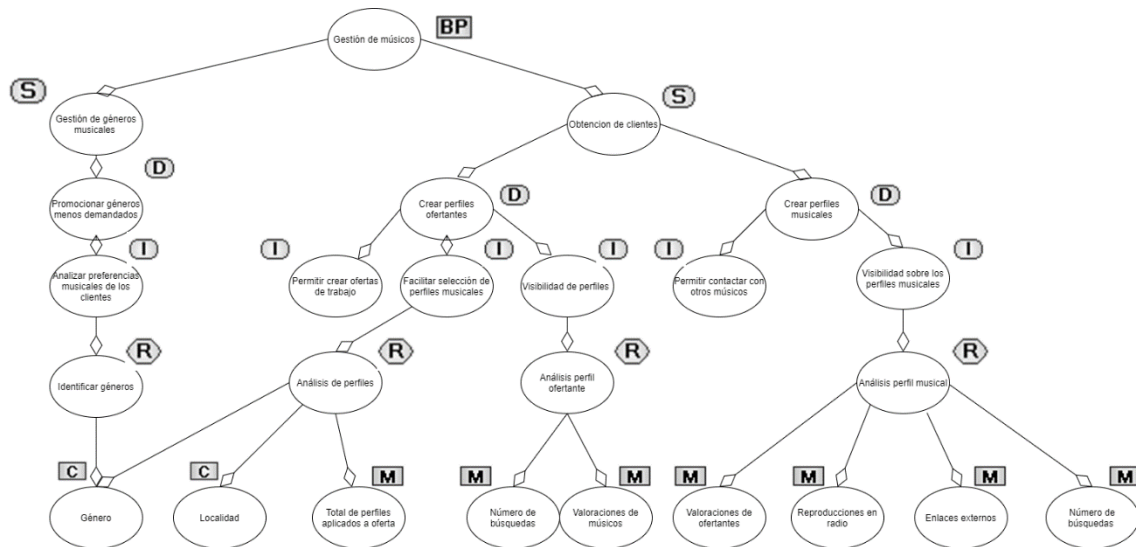
[https://www.proinf.net/permalink/generador\\_aleatorio\\_de\\_base\\_de\\_datos](https://www.proinf.net/permalink/generador_aleatorio_de_base_de_datos)

El proceso ha sido grabado para que no quede ninguna duda en los pasos que se han realizado hasta la importación de esos 10.000 datos. A través del siguiente video se puede visualizar:

<https://drive.google.com/open?id=1qlKR06SKyJhatDIwin1gfDcMXI5bExLf>

### 3. KPI'S.

El objetivo con el que se ha diseñado un árbol de objetivos del proyecto es visualizar, de forma clara, el plan estratégico o de empresa, que describirá el estado que se pretende alcanzar y su plan de mantenimiento, así como el diseño de indicadores, con lo que se espera aumentar en número de clientes.



#### 4. CMI.

##### 4.1 Mapa estratégico.

Objetivos corporativos	Perspectiva financiera	Perspectiva del cliente	Perspectiva del proceso interno	Perspectiva de aprendizaje y conocimiento
Aumentar crecimiento en el mercado del 5%.	Crecimiento	Adquisición de clientes	Innovación	Productividad del cliente
Aumentar beneficios 10%	Sostenimiento	Precios	Proceso operativo	Sistema de participación para reducción de costes
Aumentar	Cosecha	Rentabilidad del cliente	Postventa	Retención de empleados

##### PERSPECTIVA FINANCIERA.

Nos indica que los resultados financieros se consiguen únicamente si los clientes están satisfechos.

*Crecimiento.* Esto enfatiza en el crecimiento de ventas en nuevos mercados y a nuevos clientes, y nuevos servicios manteniendo un nivel de gasto adecuado para el desarrollo de procesos, sistemas, marketing, distribución. Sufrir un crecimiento del 5% en el mercado.

*Sostenimiento:* Beneficios de explotación y margen bruto: 10%. Tratamos de obtener buenos rendimientos sobre el capital aportado.

*Cosecha:* Presionada hacia el Cash Flow (Cualquier inversión que se haga debe tener una serie de restituciones de dinero seguros e inmediatas) Aumentando el cash Flow en un 15% (Beneficio neto + Amortizaciones + Previsiones) se refinanciaría la deuda para ahorrar un 0,5 de intereses.

##### PERSPECTIVA DEL CLIENTE.

La propuesta de valor para el cliente describe el método para generar ventas y consumidores fieles. Por eso se encuentra íntimamente ligada con la perspectiva de los procesos necesarios para que los clientes queden satisfechos.

*Satisfacción del cliente:* El cliente estará satisfecho si a las ofertas que se ha inscrito dan resultados y conseguirá empleo en una empresa.

*Rentabilidad del cliente:* Fidelidad. Se “premia” a un usuario otorgándole como una veracidad de su perfil, similar al “click” de Instagram.

*Precios:* Aumentar el valor la app modificando su envasado. Aumentar la publicidad en géneros poco utilizados.

## PERSPECTIVA DEL PROCESO INTERNO.

Constituyen el engranaje que lleva a la práctica la propuesta de valor para el cliente, constituyendo por tanto el sistema de organización de la empresa.

*Proceso de innovación.* Analizando el mercado (y a nuestros competidores, ya cercanos relacionados con la música, o más globales estilos jobtoday) y creando más servicios, como la reserva de estudios de grabación para músicos.

*Proceso operativo.* Producción y distribución. Aumentar publicidad y marketing en un 5%.

*Servicio postventa.* Analizar qué géneros son los que más triunfan en el servicio web, cuáles son las ciudades más demandadas.

## PERSPECTIVA DE APRENDIZAJE Y CONOCIMIENTO.

La estrategia debe especificar y tener en cuenta qué tareas (capital humano), qué tecnología (capital de la información) y qué entorno (cultura organizacional) para apoyar los procesos. Sin ellos la creación de valor no existirá. Por lo tanto, en última instancia, tampoco se cumplirán los objetivos financieros.

*Productividad del cliente:* Utilizar este servicio es una herramienta que facilita tanto a la empresa como al músico. No necesita invertir tiempo en poner anuncios en otros portales de empleo.

Sistema de participación en reducción de costes.

*Retención del empleado:* El empleado tiene acceso a la innovación, a través de cursos, para su progreso personal como programador.

## 5. GRAFANA.

### 5.1 Instalación.

En el caso de este proyecto, HearMee, se ha procedido a instalar Grafana a través del terminal mediante los comandos:

```
sudo apt-get update  
sudo apt-get install grafana
```

Además, se ha procedido a la configuración de ciertas directivas recomendadas por el profesor para su correcta integración en el proyecto (se han seguido todos los videotutoriales dirigidos por Juan Antonio en la plataforma *Moodle*).

### 5.2 Inclusión en el proyecto.

Grafana por defecto se establece en el puerto :3000, algo que, al estar automatizado, resulta poco costoso de configurar, ya que viene todo prácticamente hecho.

La ruta establecida ha sido, por tanto, <http://hearmee.ovh:3000/> y el usuario de acceso y contraseña se han decidido establecer las siguiente:



Usuario: admin  
Contraseña: jaimemee6

Tras permitir el acceso al panel principal se ha creado la conexión a la BBDD. En el particular caso de este proyecto, al tener el sistema alojado en mongobd, Grafana interponía ciertos problemas de incompatibilidad: directamente, no funciona con esta BBDD.

Es por ello por lo que se ha planteado una alternativa a ello: sin borrar la base de datos desplegada en Mongobd, se ha realizado una migración a otra BBDD con la que Grafana no tiene problemas de compatibilidad: MySQL.

El método interpuesto para ello ha sido la realización de un Script encargado de migrar dichos datos a la nueva BBDD sin pérdidas de ninguno de ellos.

Previamente, se ha creado la Database que nos interesa manejar: *nm*; y las Tables correspondientes: *artista*, *empresa* y *oferta*.

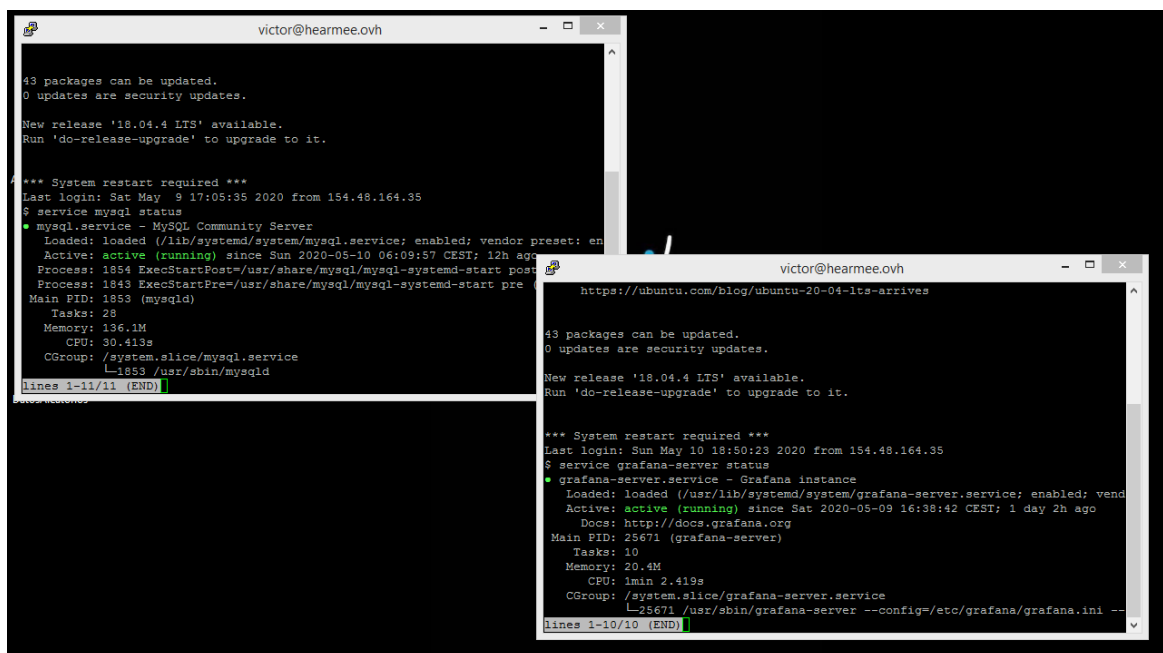
Lo demás no nos resulta interesante y sería una pérdida de tiempo copiarlo y no usarlo.

Una vez creada la nueva base de datos y metida en ella unos datos de prueba con los que se puede manejar, se le pide al servicio que comience, y comprobamos, mediante los comandos siguientes, si se está ejecutando correctamente:

```
sudo service mysql start
sudo service mysql status

-----

sudo service grafana-server start
sudo service grafana-server status
```



The image shows two terminal windows from a user named 'victor@hearmee.ovh'. The left window shows the output of 'service mysql status', indicating that the MySQL service is loaded, active, and running. The right window shows the output of 'service grafana-server status', indicating that the Grafana service is loaded, active, and running. Both windows also show system update notifications and a system restart requirement.

```
victor@hearmee.ovh
43 packages can be updated.
0 updates are security updates.

New release '18.04.4 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

*** System restart required ***
Last login: Sat May 9 17:05:35 2020 from 154.48.164.35
$ service mysql status
mysql.service - MySQL Community Server
Loaded: loaded (/lib/systemd/system/mysql.service; enabled; vendor preset: en
Active: active (running) since Sun 2020-05-10 06:09:57 CEST; 12h ago
Process: 1854 ExecStartPost=/usr/share/mysql/mysql-systemd-start post
Process: 1843 ExecStartPre=/usr/share/mysql/mysql-systemd-start pre
Main PID: 1853 (mysqld)
Tasks: 28
Memory: 136.1M
CPU: 30.413s
CGroup: /system.slice/mysql.service
└─1853 /usr/sbin/mysqld
lines 1-11/11 (END)
```

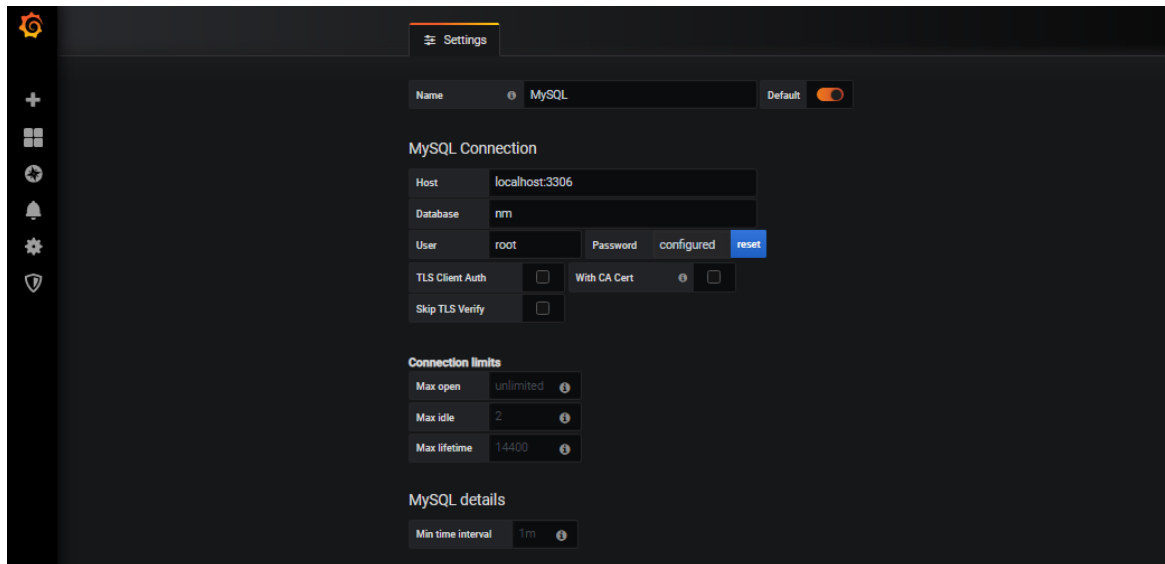
```
victor@hearmee.ovh
https://ubuntu.com/blog/ubuntu-20-04-lts-arrives

43 packages can be updated.
0 updates are security updates.

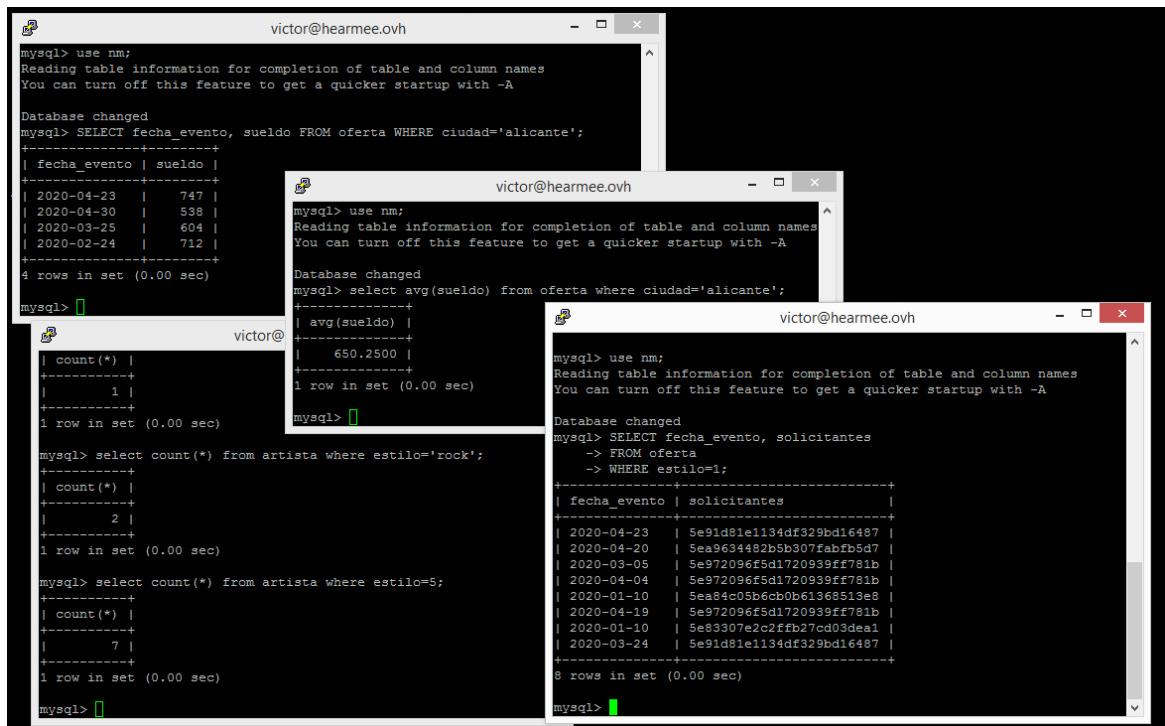
New release '18.04.4 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

*** System restart required ***
Last login: Sun May 10 18:50:23 2020 from 154.48.164.35
$ service grafana-server status
grafana-server.service - Grafana instance
Loaded: loaded (/usr/lib/systemd/system/grafana-server.service; enabled; vend
Active: active (running) since Sat 2020-05-09 16:38:42 CEST; 1 day 2h ago
Docs: http://docs.grafana.org
Main PID: 25671 (grafana-server)
Tasks: 10
Memory: 20.4M
CPU: 1min 2.419s
CGroup: /system.slice/grafana-server.service
└─25671 /usr/sbin/grafana-server --config=/etc/grafana/grafana.ini --
lines 1-10/10 (END)
```

Tras ello, se establece la conexión con Grafana y mysql:

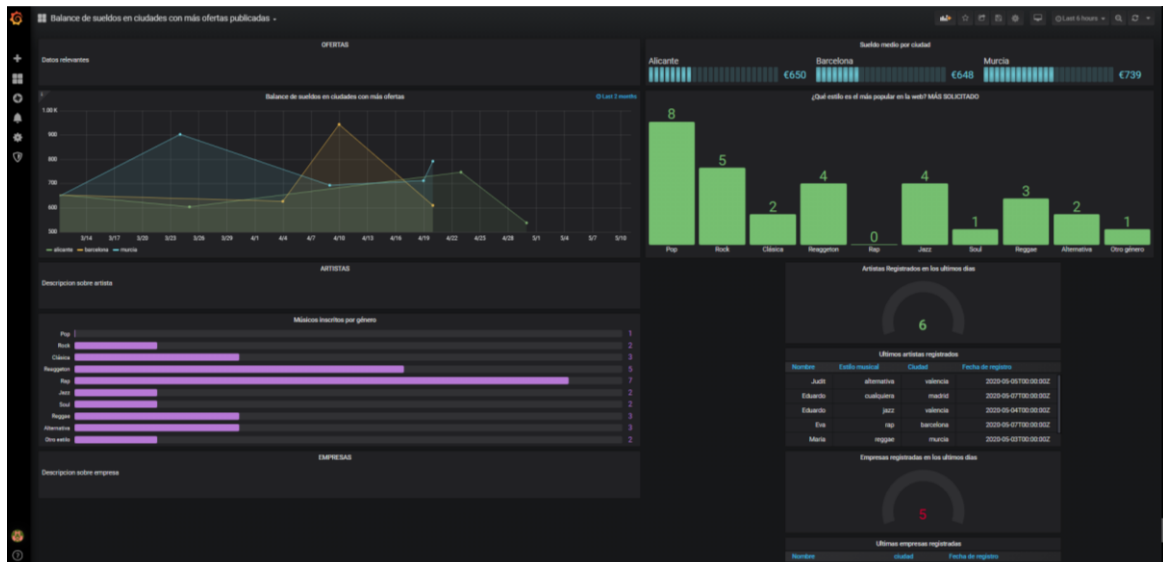


A partir de este punto ya podemos manejar los datos de los que deseamos tener un registro periódicamente. Para ello previamente se han comprobado las queries en mysql, si eran factibles y correctas, para evitar futuros problemas:

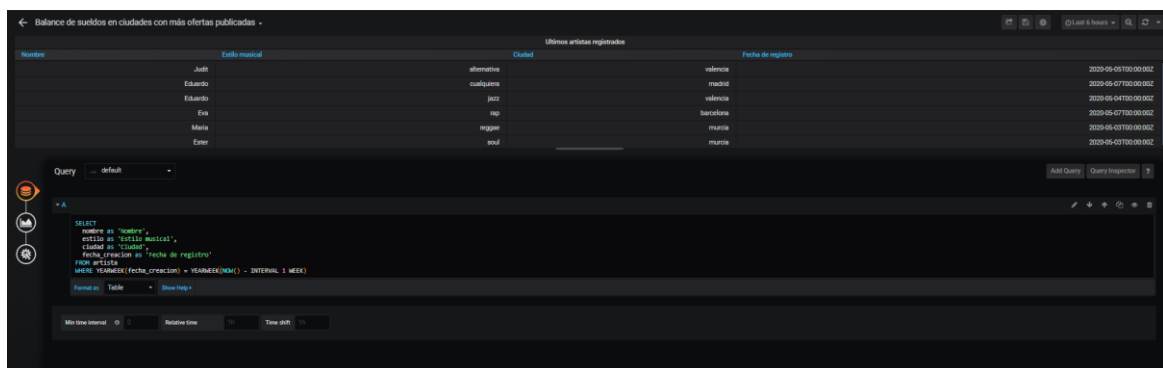


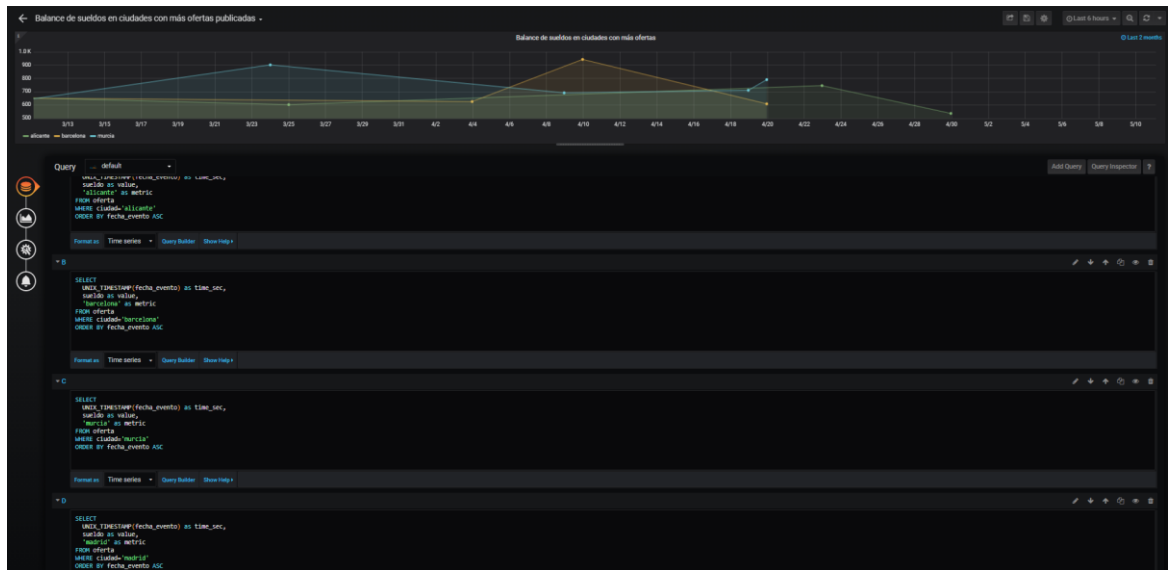
El siguiente enlace redirige a una explicación más detallada del por qué y todas las queries realizadas: <https://drive.google.com/open?id=1x1zEOP-HqzosOX1meZTgmXKtOw4-3yOS>

Una vez comprobadas que son funcionales, se introducen en Grafana.



A continuación, se mostrarán algunas SELECT de los paneles que se han realizado, en las que se tiene en cuenta el tiempo para tener una visión actualizada de la situación en la que se encuentra el proyecto:





Para una visión más profunda de los paneles creados es conveniente acceder a la vista de Grafana que se ha creado con el usuario y la contraseña mencionada en puntos anteriores:

<http://hearmee.ovh:3000/>

### 5.3 Cmi en grafana.

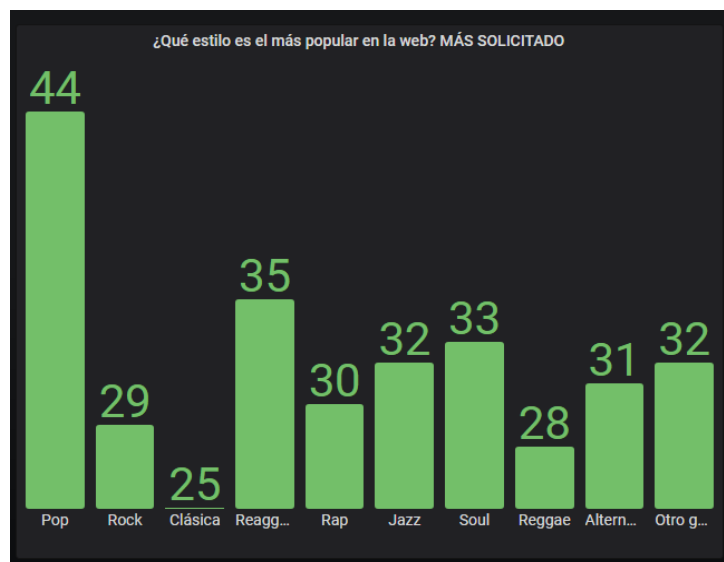
Los cuadros de mando finales creados en Grafana que proporcionan información relevante de la aplicación son los siguientes:



Los tres primeros dashboard están relacionados con la tabla OFERTA:

El primer dashboard representa una grafica con el balance de sueldos pagados en las ciudades más relevantes: de Alicante, Barcelona, Murcia y Madrid. El segundo muestra una media de sueldo pagado hasta la fecha en las ciudades mencionadas anteriormente.

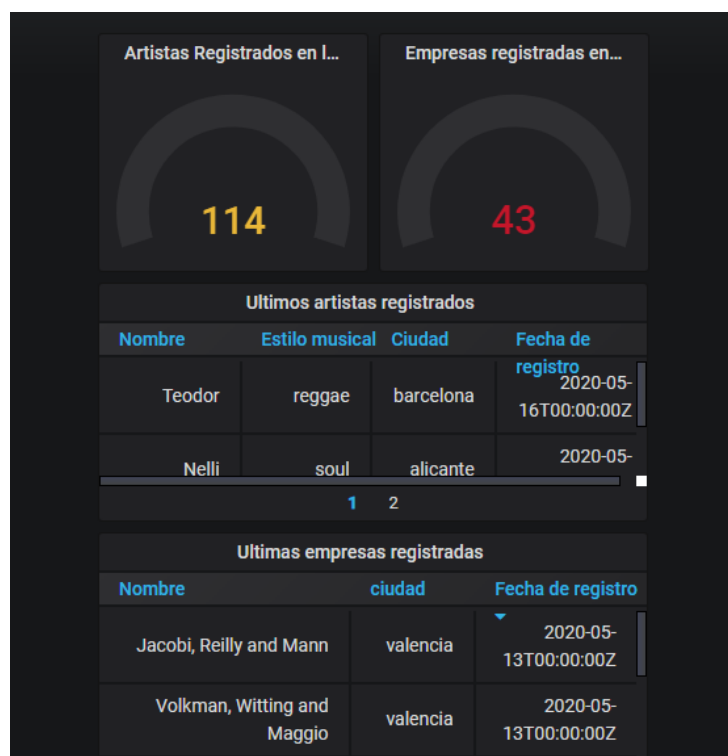
El tercer Dashboard de la aplicación analiza el género más solicitado hasta la fecha de hoy, entre todos los disponibles.



El siguiente dashboard (cuarto) muestra en una tabla el número de músicos registrados según el género que dominan.

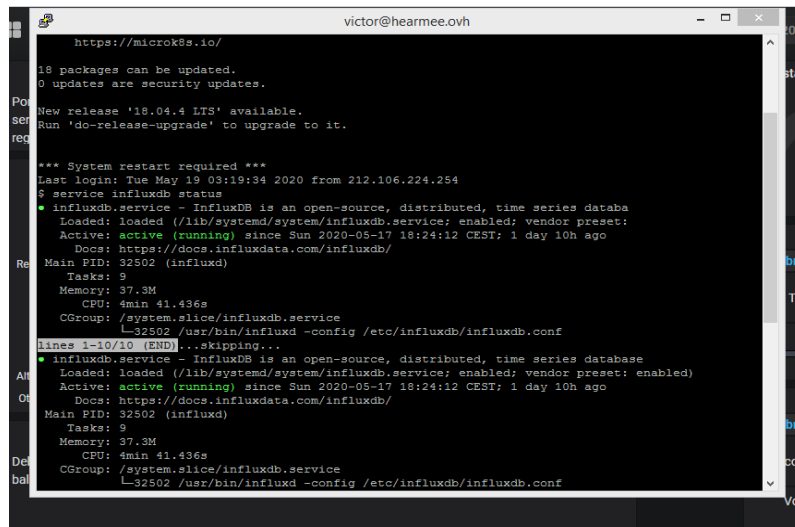


Y los últimos cuatro, los músicos y empresas registradas en la última semana, así como el género, ciudad y fecha de creación.



Con esta última imagen finaliza el primer dashboard creado referido a los datos recopilados de mysql. Por otra parte, se han recopilado datos relevantes referidos a la propia CPU y rendimiento del servidor, en este caso, independiente de mysql.

Este análisis ha sido realizado con InfluxDB, que, tras una simple instalación y configuración, corre ahora mismo en nuestro servidor:

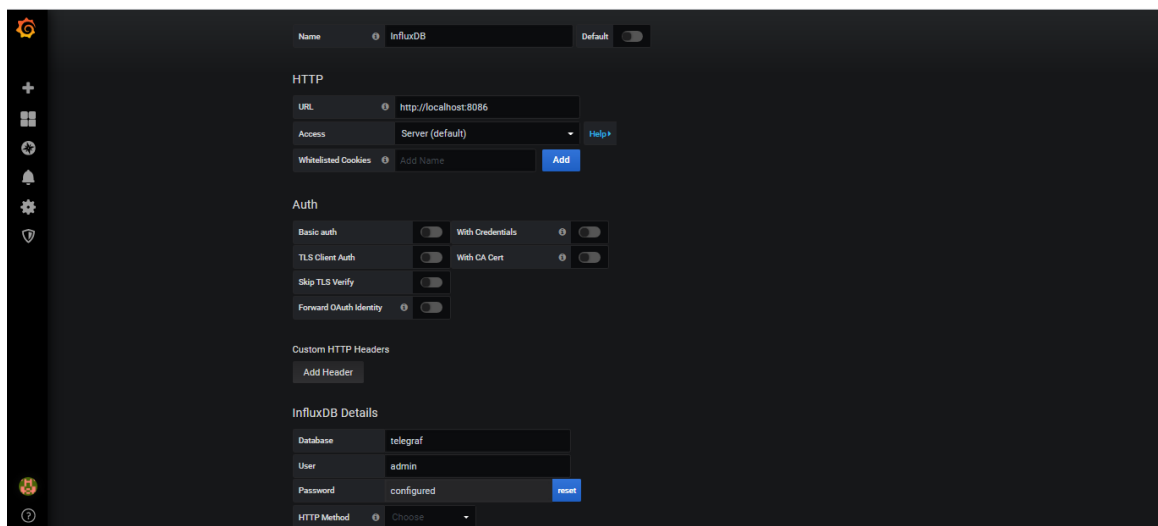


```
https://microk8s.io/
18 packages can be updated.
0 updates are security updates.
New release '18.04.4 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

*** System restart required ***
Last login: Tue May 19 03:19:34 2020 from 212.106.224.254
$ service influxdb status
● influxdb.service - InfluxDB is an open-source, distributed, time series database
   Loaded: loaded (/lib/systemd/system/influxdb.service; enabled; vendor preset:
   Active: active (running) since Sun 2020-05-17 18:24:12 CEST; 1 day 10h ago
   Docs: https://docs.influxdata.com/influxdb/
   Main PID: 32502 (influxd)
   Tasks: 9
   Memory: 37.3M
   CPU: 4min 41.436s
   CGroup: /system.slice/influxdb.service
           └─32502 /usr/bin/influxd -config /etc/influxdb/influxdb.conf

lines 1-10/10 (END) ...skipping...
● influxdb.service - InfluxDB is an open-source, distributed, time series database
   Loaded: loaded (/lib/systemd/system/influxdb.service; enabled; vendor preset: enabled)
   Active: active (running) since Sun 2020-05-17 18:24:12 CEST; 1 day 10h ago
   Docs: https://docs.influxdata.com/influxdb/
   Main PID: 32502 (influxd)
   Tasks: 9
   Memory: 37.3M
   CPU: 4min 41.436s
   CGroup: /system.slice/influxdb.service
           └─32502 /usr/bin/influxd -config /etc/influxdb/influxdb.conf
```

El siguiente paso muestra como se ha conectado influxDB a grafana, y el puerto por el que corre:



The image shows the Grafana configuration page for the InfluxDB data source. The 'Name' is 'InfluxDB' and the 'Default' toggle is off. Under the 'HTTP' section, the 'URL' is 'http://localhost:8086', 'Access' is 'Server (default)', and there are fields for 'Whitelisted Cookies'. The 'Auth' section has 'Basic auth' and 'TLS Client Auth' toggled off, with 'With Credentials' and 'With CA Cert' also off. 'Skip TLS Verify' is on, and 'Forward OAuth Identity' is off. The 'Custom HTTP Headers' section has an 'Add Header' button. The 'InfluxDB Details' section shows 'Database' as 'telegraf', 'User' as 'admin', and 'Password' as 'configured' with a 'reset' button. The 'HTTP Method' is set to 'Choose'.

Por último, mediante sentencias , se han creado los cuadros de mando que monitorizan el rendimiento de la CPU de nuestro servidor:



## 6. OTROS PROCESOS REALIZADOS RELEVANTES.

Para conseguir el éxito de todos los cuadros de mando ha sido necesario realizar determinados procesos/archivos:

### 6.1 Creación de un script migratorio de datos mongo – mysql.

Dado que no es posible conectar MongoDB con Grafana, se ha dedicado a crear un archivo .js que lea los .json exportados desde mongo con las colecciones requeridas y así, realizar las queries necesarias para rellenar los campos que nos interesan:

Nota: El servicio se ha creado en “/var/www/html/api/app/routes/readjson.js”.

```

C:\Users\ASUS\AppData\Local\Temp\scp09669 > var > www > html > api > app > routes > # readjson.js ...
28 Router.get('/', (req, res) => {
29
30   let rawdata = fs.readFileSync('/home/victor/empresasreadjs.json');
31   let empresa = JSON.parse(rawdata);
32   connection.connect(function(err) {
33     if (err) {
34       console.log(err);
35       return;
36     } else {
37
38     }
39   })
40
41   var elements2 = []
42   var query2 = "INSERT INTO empresa (ID, email, nombre, password, ciudad, date"
43   empresa.forEach((obj, i) => {
44     if (obj.date && obj.ciudad != "") {
45
46       var splitDate = obj.date.$date.split("T")
47       var element = {
48         id: obj._id.$oid,
49         email: obj.email,
50         name: obj.nombre,
51         password: obj.password,
52         city: obj.ciudad,
53         date: splitDate[0]
54       }
55       elements2.push(element)
56     }
57   });
58
59   connection.query(query2, elements2.map(item => [item.id, item.email, item.
60     console.log("enter")
61     if (err) {
62       console.log(err);
63       return;
64     } else {
65
66
67     let rawdata = fs.readFileSync('/home/victor/artistareadjs.json');
68     let artist = JSON.parse(rawdata);
69
70     var query = "INSERT INTO artista (ID, email, instrumentos, estilo,
71     artist.forEach((obj, i) => {
72       if (obj.fechaCreacion && obj.ciudad != "null" && obj.instrument
73       var splitDate = obj.fechaCreacion.$date.split("T")
74       var element = {
75         id: obj._id.$oid,
76         email: obj.email,
77         instruments: obj.instrumentos,
78         style: obj.estilo,
79         name: obj.nombre,
80         password: obj.password,
81         city: obj.ciudad,
82         date: splitDate[0]
83       }
84       elements.push(element)
85     })
86
87     connection.query(query, [elements.map(item => [item.id, item.email,
88       console.log("enter")
89       if (err) {
90         console.log(err);
91         return;
92       } else {
93
94
95
96
97     let rawdata = fs.readFileSync('/home/victor/ofertasreadjs.j
98     let ofertas = JSON.parse(rawdata);
99     var elements = []
100     var query = "INSERT INTO oferta (ID, estilo, ciudad, sueldo
101     ofertas.forEach((obj, i) => {
102       if (obj.fechaEvento && obj.ciudad != "null" && obj.suelo
103       var splitDate = obj.fechaEvento.$date.split("T")
104       var element = {
105         id: obj._id.$oid,
106         style: obj.estilo,
107         city: obj.ciudad,
108         pay: obj.sueldo,
109         date: splitDate[0]
110       }
111       elements.push(element)

```

El resultado es que, tras la llamada get a [www.hearmee.com/api/readjson](http://www.hearmee.com/api/readjson) los datos exportados con el comando mongoexport en la ruta “home/victor” (“empresasreadjs.json”,



“artistareads.json” y “ofertasreads.json”) se copiarán en un formato que mysql admite y administra en su base de datos. (Para ello hay que realizar la conexión previa a la base de datos de mysql, que se encuentra entre las primeras líneas del readjson.js.

Solo con este script ya tendríamos los datos copiados de una bbdd no relacional a otra relacional.

Pero este archivo no se ejecuta automaticamete, por ello es necesario crear un fichero adicional .sh que sí permita ejecutarlo tras leerlo.

## 6.2 Creación y ejecución de un archivo .sh

Este es el fichero que necesitamos para automatizar el script. Con él iremos por pasos:

1. Primero vaciaremos las tablas de mysql, para rellenarlas nuevamente.
2. Seguidamente, realizará los mongoexports de las colecciones necesarias.
3. Y para terminar, ejecutará nuestro archivo “readjson.js” tras un delay de 10s.

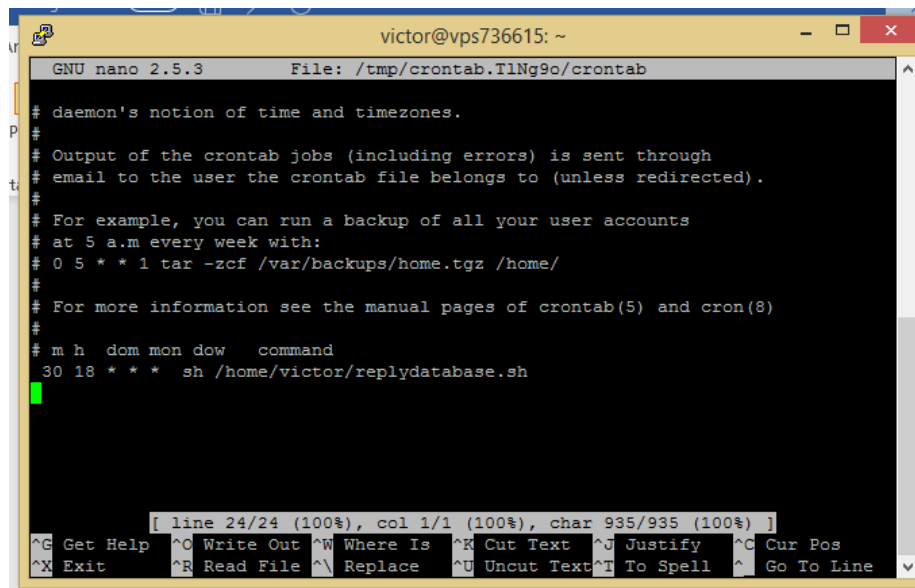
Nota: La ruta del fichero es la principal, igual que la que exporta las colecciones de mongo.

```
replydatabase.sh x
C: > Users > ASUS > AppData > Local > Temp > scp21083 > home > victor > replydatabase.sh
1  #!/bin/sh
2
3  echo "truncate table oferta" | mysql -u'root' -p'123qwe' -D'nm'
4  echo "truncate table artista" | mysql -u'root' -p'123qwe' -D'nm'
5  echo "truncate table empresa" | mysql -u'root' -p'123qwe' -D'nm'
6
7  mongoexport --db api --collection ofertas --out ofertasreadjs.json --jsonArray
8  mongoexport --db api --collection artistas --out artistareadjs.json --jsonArray
9  mongoexport --db api --collection empresas --out empresasreadjs.json --jsonArray
10
11 sleep 10
12
13 curl 'https://hearmee.ovh/api/readjson/'
14
15 echo "Create file" > myfile.txt
16
17 cat myfile.txt
```

Con este fichero casi finalizaríamos la migración con éxito. Ahora tenemos que pensar, ¿Cuándo ejecutar este fichero? Si echamos la vista atrás y nos fijamos en los cuadros de mando que monitorizan el rendimiento de la CPU podemos intuir a qué hora y cada cuanto nos resulta útil lanzar el .sh.

Con crontab eso es posible realizarse. Tras su instalación solo con seguir los siguientes pasos se tiene automatizada la hora de migración de datos, y cada cuánto.

**Crontab -e**



```
victor@vps736615: ~
GNU nano 2.5.3 File: /tmp/crontab.TlNg9o/crontab
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h dom mon dow   command
30 18 * * * sh /home/victor/replydatabase.sh
[ line 24/24 (100%), col 1/1 (100%), char 935/935 (100%) ]
^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos
^X Exit ^R Read File ^\ Replace ^U Uncut Text ^T To Spell ^_ Go To Line
```

Con este último paso ya tendríamos la aplicación completada y lista para usar. Y con ello finaliza el progreso que se ha llevado durante estos meses con la asignatura de Negocios Multimedia.