



# Trabajo de Fin de Grado

Grado en Ingeniería Informática

## Almacenamiento y Visualización de Series Temporales

*Time Series Storage and Visualization*

Waldo Nazco Torres

La Laguna, 4 de septiembre de 2018

Dña. **Isabel Sánchez Berriel**, con N.I.F. 42.885.838-S profesora Contratada Doctora adscrita al Departamento de Ingeniería Informática y de Sistemas de la Universidad de La Laguna, como tutora.

Dña. **Luz Marina Moreno De Antonio**, con N.I.F. 45.457.492-Q profesora Contratada Doctora adscrita al Departamento de Ingeniería Informática y de Sistemas de la Universidad de La Laguna, como cotutora.

## **C E R T I F I C A ( N )**

Que la presente memoria titulada:

*“Almacenamiento y Visualización de Series Temporales”*

ha sido realizada bajo su dirección por D. **Waldo Nazco Torres**,  
con N.I.F. 78.708.857-M.

Y para que así conste, en cumplimiento de la legislación vigente y a los efectos oportunos firman la presente en La Laguna a 4 de septiembre de 2018.

## Agradecimientos

A mi familia y amigos por su apoyo incondicional.  
A Isabel Sánchez Berriel y Luz Marina Moreno De Antonio por su ayuda, disposición y labor docente durante el desarrollo del trabajo de fin de grado.

# Licencia



© Esta obra está bajo una licencia de Creative Commons Reconocimiento-  
NoComercial 4.0 Internacional.

## **Resumen**

*El objetivo de este trabajo ha sido la consulta y recogida de datos procedentes del Instituto Canario de Estadística (ISTAC) relacionados con distintas áreas (turismo en Canarias, empleo registrado, matriculación de vehículos, empresas inscritas en la seguridad social, etc.) mediante peticiones a los indicadores estadísticos a una API. Se utiliza una base de datos temporal (Time Series Database) y se visualizan los datos registrados mediante una aplicación especializada en la visualización de cuadros de mando basados en métricas definidas sobre series de tiempo.*

**Palabras clave:** API, ISTAC, Base de datos temporal (Time Series Database), Visualización de cuadros de mando.

## **Abstract**

*The objective of this work has been the consultation and collection of data from the Canary Institute of Statistics (ISTAC) related to different areas (tourism in the Canary Islands, registered employment, vehicle registration, companies registered in the social security, etc...) through requests to statistical indicators to an API. A temporary database (Time Series Database) is used and the registered data are visualized through a specialized application in the visualization of dashboards based on metrics defined over time series.*

**Keywords:** API, ISTAC, Time Series Database, Visualization of dashboards, Statistical Indicators.

# Índice general

<b>Parte I: Introducción y conceptos básicos.....</b>	<b>1</b>
<b>Capítulo 1 Introducción.....</b>	<b>2</b>
1.1 Objetivos.....	2
1.2 Alcance.....	3
<b>Capítulo 2 Estado del arte .....</b>	<b>4</b>
2.1 Estado de la tecnología.....	4
2.1.1 Introducción a las Bases de datos de series temporales.....	4
2.1.2 Estudio de Bases de datos de series temporales.....	4
2.1.3 Tecnologías a usar en este proyecto.....	10
2.1.4 API ISTAC.....	11
2.1.5 API BBVA MARKET.....	12
2.2 Antecedentes del software.....	12
<b>Parte II: Cuerpo principal. Descripción del trabajo.....</b>	<b>16</b>
<b>Capítulo 3 Indicadores estadísticos.....</b>	<b>17</b>
3.1 Introducción a los indicadores estadísticos .....	17
3.2 Selección de indicadores .....	17
3.3 Tabla de indicadores .....	20
<b>Capítulo 4 Tratamiento y almacenamiento de los datos.....</b>	<b>21</b>
4.1 Introducción al tratamiento y almacenamiento de los datos .....	21
4.2 Introducción a InfluxDB .....	22
4.3 Estructura de datos en InfluxDB .....	23
4.4 Operaciones básicas en InfluxDB .....	25

<b>Capítulo 5 Visualización de los datos.....</b>	<b>30</b>
5.1 Introducción a la visualización de los datos .....	30
5.2 Chronograf .....	30
5.2.1 Introducción a Chronograf.....	30
5.2.2 Configuración de Chronograf.....	32
5.2.3 Caso de ejemplo en Chronograf.....	35
5.3 Grafana.....	36
5.3.1 Introducción a Grafana.....	36
5.3.2 Configuración de Grafana.....	36
5.3.3 Caso de ejemplo en Grafana.....	38
5.4 Chronograf vs Grafana .....	39
<b>Capítulo 6 Desarrollo .....</b>	<b>40</b>
6.1 Plan de trabajo.....	40
6.2 Problemas encontrados.....	40
<b>Parte III: Conclusiones.....</b>	<b>42</b>
<b>Capítulo 7 Conclusiones y líneas futuras .....</b>	<b>43</b>
<b>Capítulo 8 Summary and Conclusions .....</b>	<b>44</b>
<b>Capítulo 9 Presupuesto .....</b>	<b>45</b>
<b>Apéndice A. Almacenamiento de datos de indicadores estadísticos en InfluxDB.</b>	<b>47</b>
A.1 Ejemplo de script en Python.....	47
<b>Bibliografía.....</b>	<b>51</b>

# Índice de figuras

Figura 2.1 Ejemplo de consulta en el swagger de la API del ISTAC.....	11
Figura 2.2 Opciones de licencia que ofrece API BBVA MARKET.....	12
Figura 2.3 Pantalla principal PULSEC.....	13
Figura 2.4 Resultados de consulta en PULSEC.....	14
Figura 2.5 Ejemplo de consulta VISTac.....	15
Figura 4.1 Ejecución de InfluxDB en la terminal de consola.....	25
Figura 4.2 Mostrar bases de datos en InfluxDB.....	26
Figura 4.3 Seleccionar base de datos en InfluxDB.....	26
Figura 4.4 Mostrar medidas en InfluxDB.....	26
Figura 4.5 Mostrar claves de etiquetas en InfluxDB.....	27
Figura 4.6 Mostrar valores de etiquetas en InfluxDB.....	27
Figura 4.7 Mostrar toda la información de un indicador en InfluxDB.....	28
Figura 5.1 Presentación de Chronograf.....	32
Figura 5.2 Configuración de conexión Chronograf.....	33
Figura 5.3 Conexiones creadas en Chronograf.....	33
Figura 5.4 Conexión establecida entre Chronograf e InfluxDB.....	34
Figura 5.5 Prueba de conexión entre Chronograf e InfluxDB.....	34
Figura 5.6 Ejemplo de gráfica en Chronograf.....	35
Figura 5.7 Ejemplo de diagrama de barras en Chronograf.....	36
Figura 5.8 Pantalla principal de Grafana.....	37
Figura 5.9 Configuración de datasources en Grafana.....	37
Figura 5.10 Creación de query con Grafana.....	38
Figura 5.11 Ejemplo de visualización con Grafana.....	38

# Índice de tablas

Tabla 2.1 Comparativa de bases de datos temporales.....	7
Tabla 3.1 Indicadores estadísticos y sus características.....	20
Tabla 6.1 Plan de trabajo del proyecto.....	40
Tabla 10.1 Estimación temporal de desarrollo del proyecto.....	45
Tabla 10.2 Presupuesto total del proyecto.....	46

**Parte I**

**Introducción y fundamentos básicos**

# Capítulo 1

## Introducción

### Objetivos

El objetivo de este proyecto titulado tratamiento y visualización de series temporales consiste en la consulta y recogida de datos procedentes de una API perteneciente al Instituto Canario de Estadística (ISTAC) [2]. Los datos de los diferentes indicadores recopilados se almacenan en una base de datos temporal (Time Series Databases) [15]. Cabe destacar entre los objetivos el estudio y análisis de este tipo de tecnologías NoSQL, orientadas al almacenamiento de grandes volúmenes de datos, generalmente generados por aplicaciones de datos en tiempo real y de los que se almacena el histórico. Del mismo modo, una vez almacenados los datos en la base de datos temporal se procederá a su visualización mediante una herramienta para la elaboración de cuadros de mando basada en métricas sobre series de tiempo.

Los datos que recopilaremos del ISTAC abarcarán diferentes áreas como son el turismo en Canarias, empleo registrado en hostelería, afiliaciones a la seguridad social, empresas inscritas en la seguridad social, matriculación de vehículos y turismos, sociedades mercantiles creadas, tráfico aéreo comercial de aeronaves, mercancías y pasajeros, índice de cifra de negocios y de ocupación (IASS), índice de precios de consumo general y subyacente (IPC), etc.

Para llevar a cabo el trabajo, se realiza un estudio sobre las diferentes bases de datos temporales disponibles, de forma que se pueda contar con criterios objetivos basados en sus requisitos, funcionalidades, ventajas, etc, para seleccionar una de ellas con objeto de almacenar los datos que recogeremos de la API de indicadores del ISTAC.

Para la visualización de los datos se aprovechará una plataforma abierta diseñada para la monitorización de datos en tiempo real.

Otro objetivo secundario fue el estudio de una segunda API perteneciente a la compañía BBVA, el propósito era determinar la viabilidad de obtener datos de turismo en dicha API para el desarrollo de nuestro proyecto, si bien finalmente fue descartada ya que no ofrece un nivel de acceso a la información gratuito (véase capítulo 2.1.5), sino que se requiere un contrato de pago para poder tener acceso sin límites a sus datos.

## **Alcance**

Se estudiarán diferentes soluciones para el almacenamiento de series históricas de datos.

Se diseñará e implementará una Base de Datos de Series Temporales utilizando como datos de ejemplo indicadores del ISTAC.

Se desarrollarán programas para la carga de datos (uno por cada indicador del ISTAC) que realizarán las consultas a la información disponible en la API relacionadas con las diferentes áreas anteriormente mencionadas (turismo en Canarias, empleo registrado, empresas inscritas en la seguridad social, etc...) insertándolos en la base de datos temporal.

Finalmente se configurará y desplegará una aplicación de visualización de datos para la elaboración de cuadros de mandos basados en métricas de series de tiempo que muestren los datos almacenados en la base de datos y su evolución a lo largo del tiempo.

# Capítulo 2

## Estado del Arte

### 2.1 Estado de la tecnología

#### 2.1.1 Introducción a las Bases de datos de series temporales

Una base de datos de series de tiempo podría definirse como un sistema gestor de bases de datos, que como su propio nombre indica, presta especial atención a los aspectos temporales, contando con un modelo de datos temporal y una versión temporal del lenguaje de consulta estructurado.

En la actualidad existen diversas bases de datos de series temporales para el almacenamiento y tratamiento de los datos. No todas las bases de datos de series temporales ofrecen las mismas funcionalidades y características por lo que debemos seleccionar la que mejor se ajuste a nuestras necesidades. Existen bases de datos de series temporales de código abierto y licencia privativa. Para el desarrollo de nuestro proyecto utilizaremos una base de datos temporal de código abierto.

#### 2.1.2 Estudio de Bases de datos de series temporales

A continuación se ha realizado un estudio de diversas Bases de datos de series temporales disponibles [1]. En base a sus funcionalidades y características seleccionaremos la que se ajuste a nuestras necesidades para el desarrollo del proyecto. No todas las bases de datos temporales son compatibles con la herramienta elegida para la visualización de datos que utilizaremos en nuestro proyecto por lo que nos centraremos solamente en las compatibles con dicha aplicación.

- **Influxdb:**

- **Descripción:** DBMS para almacenar series de tiempo, eventos y métricas.
- **Modelo de base de datos primaria:** Time Series DBMS.
- **Sitio Web:** [www.influxdata.com/time-series-platform/influxdb](http://www.influxdata.com/time-series-platform/influxdb)
- **Documentación Técnica:** [docs.influxdata.com/influxdb](http://docs.influxdata.com/influxdb)

- **Graphite:**
  - **Descripción:** Herramienta de registro de datos y gráficos para datos de series de tiempo.
  - **Modelo de base de datos primaria:** Time Series DBMS.
  - **Sitio Web:** [github.com/graphite-project/graphite-web](https://github.com/graphite-project/graphite-web)
  - **Documentación Técnica:** [graphite.readthedocs.io](http://graphite.readthedocs.io)
  
- **OpenTSDB:**
  - **Descripción:** Serie de tiempo escalable DBMS basado en HBase.
  - **Modelo de base de datos primaria:** Time Series DBMS.
  - **Sitio Web:** [opentsdb.net](http://opentsdb.net)
  - **Documentación Técnica:** [opentsdb.net/docs/build/html/index.html](http://opentsdb.net/docs/build/html/index.html)
  
- **Prometheus:**
  - **Descripción:** Series de tiempo DBMS de código abierto y sistema de monitoreo.
  - **Modelo de base de datos primaria:** Time Series DBMS.
  - **Sitio Web:** [prometheus.io](http://prometheus.io)
  - **Documentación Técnica:** [prometheus.io/docs](http://prometheus.io/docs)
  
- **Blueflood:**
  - **Descripción:** Serie de tiempo escalable DBMS basado en Cassandra.
  - **Modelo de base de datos primaria:** Time Series DBMS.
  - **Sitio Web:** [blueflood.io](http://blueflood.io)
  - **Documentación Técnica:** [github.com/rackerlabs/blueflood/wiki](https://github.com/rackerlabs/blueflood/wiki)
  
- **Druid:**
  - **Descripción:** Almacén de datos analíticos de código abierto diseñado para consultas OLAP en datos de series temporales.
  - **Modelo de base de datos primaria:** Time Series DBMS.
  - **Sitio Web:** [druid.io](http://druid.io)
  - **Documentación Técnica:** [druid.io/docs/latest/design/index.html](http://druid.io/docs/latest/design/index.html)

- **Hawkular:**
  - **Descripción:** La métrica Hawkular es el almacenamiento métrico del sistema de monitoreo Hawkular patrocinado por Red Hat. Está basado en Casandra.
  - **Modelo de base de datos primaria:** Time Series DBMS.
  - **Sitio Web:** [www.hawkular.org](http://www.hawkular.org)
  - **Documentación Técnica:** [www.hawkular.org/hawkular-metrics/docs/-user-guide](http://www.hawkular.org/hawkular-metrics/docs/-user-guide)
  
- **Heroic:**
  - **Descripción:** Series de tiempo DBMS construido en Spotify basado en Casandra o Google Cloud Bigtable, y ElasticSearch.
  - **Modelo de base de datos primaria:** Time Series DBMS.
  - **Sitio Web:** [spotify.github.io/heroic/#!/index](http://spotify.github.io/heroic/#!/index)
  - **Documentación Técnica:** [spotify.github.io/heroic/#!/docs/overview](http://spotify.github.io/heroic/#!/docs/overview)
  
- **KairosDB:**
  - **Descripción:** Series de tiempo distribuida DBMS basada en Casandra o H2.
  - **Modelo de base de datos primaria:** Time Series DBMS.
  - **Sitio Web:** [github.com/kairosdb/kairosdb](http://github.com/kairosdb/kairosdb)
  - **Documentación Técnica:** [kairosdb.github.io](http://kairosdb.github.io)

A continuación se muestra una tabla comparativa entre las bases de datos temporales estudiadas y sus características:

BASES DE DATOS									
Características	InfluxDB	Graphite	OpenTSDB	Prometheus	Blueflood	Druid	Hawkular	Heroic	KairosDB
<b>Versión Inicial</b>	2013	2006	2011	2015	2013	2012	2014	2014	2013
<b>Licencia</b>	Código Abierto	Código Abierto	Código Abierto	Código Abierto	Código Abierto	Código Abierto	Código Abierto	Código Abierto	Código Abierto
<b>Basado en la nube</b>	No	No	No	No	No	No	No	No	No
<b>Lenguaje de implementación</b>	Go	Python	Java	Go	Java	Java	Java	Java	Java
<b>Sistemas operativos de servidor</b>	Linux OS X	Linux Unix	Linux Windows	Linux Windows	Linux OS X	Linux OS X Unix	Linux Windows OS X	Linux	Linux OS X Windows
<b>Esquema de datos</b>	Esquema Libre	Sí	Esquema Libre	Sí	Esquema Predefinido	Sí	Esquema Libre	Esquema Libre	Esquema Libre
<b>Mecanografía</b>	Datos numéricos y cadenas	Datos numéricos	Datos numéricos para métricas, cadenas para etiquetas	Datos numéricos únicamente	Sí	Sí	Sí	Sí	Sí
<b>Soporte XML</b>	No	No	No	No	No	No	No	No	No
<b>Índices secundarios</b>	No	No	No	No	No	Sí	No	Sí	No
<b>SQL</b>	SQL-like query language	No	No	No	No	Soporte SQL a través del adaptador Apache Calcite	No	No	No

<b>APIs y otros métodos de acceso</b>	HTTP API JSON over UDP	HTTP API Sockets	HTTP API Telnet API	RESTful HTTP/JSON API	HTTP REST	JSON over HTTP	HTTP REST	HQL HTTP API	Graphite Protocol HTTP REST Telnet API
<b>Lenguajes de programación soportados</b>	.Net Clojure Erlang Go Haskell Java JavaScript(Node.js) Lisp Perl PHP Python R Ruby Rust Scala	JavaScript(Node.js) Python	Erlang Go Java Python R Ruby	.Net C++ Go Haskell Java JavaScript(Node.js) Python Ruby	Java Python	Clojure JavaScript PHP Python R Ruby Scala	Go Java Python Ruby	Java Python	Java JavaScript PHP Python
<b>Scripts del lado del servidor</b>	No	No	No	No	No	No	No	No	No
<b>Desencadenantes</b>	No	No	No	No	No	No	Sí	No	No
<b>Métodos de particionamiento</b>	Sharding	Ninguno	Sharding	Sharding	Sharding	Sharding	Sharding	Sharding	Sharding
<b>Métodos de replicación</b>	Factor de replicación seleccionable	Ninguno	Factor de replicación seleccionable	Sí	Factor de replicación seleccionable	Sí, a través de HDFS, S3 u otros motores de almacenamiento	Factor de replicación seleccionable	Sí	Factor de replicación seleccionable

<b>Mapa reducido</b>	No	No	No	No	No	No	No	No	No
<b>Llaves exteriores</b>	No	Ninguno	No						
<b>Conceptos de transacción</b>	No	No	No	No	No	No	No	No	No
<b>Concurrencia</b>	Sí	No	Sí						
<b>Durabilidad</b>	Sí	Sí	Sí	Sí	Sí	Sí	Sí	Sí	Sí
<b>Capacidades de memoria</b>	Sí	Sí	No						
<b>Conceptos de usuario</b>	Simple gestión de derechos a través de cuentas de usuario	No	No	No	No	No	No	No	No

Tabla 2.1 Comparativa de bases de datos temporales

### 2.1.13 Tecnologías a usar en este proyecto

En primer lugar cabe destacar que el desarrollo íntegro del proyecto se ha realizado bajo un sistema Linux (Ubuntu 18.04).

La base de datos seleccionada para el desarrollo de nuestro proyecto es InfluxDB [6], es una de las bases de datos de series de tiempo más utilizadas en la actualidad, según los rankings [1] que se han consultado. Soporta un mayor número de lenguajes de programación con respecto al resto y puede obtener datos mediante métodos de acceso HTTP API y JSON. Actualmente es complicado encontrar información técnica y bibliográfica de las bases de datos temporales más allá de las que ofrecen los sitios web oficiales de cada una, siendo InfluxDB de las que más información se puede llegar a encontrar en la red mediante foros y artículos.

Los datos que almacenaremos en InfluxDB los obtendremos en formato JSON [13] mediante peticiones a la API del Instituto Canario de Estadística (ISTAC) [2] relacionadas con las áreas de turismo, empleo, IPC, PIB, matriculación de vehículos, etc...

Para la recogida de los datos desarrollaremos programas en Python [11], un script por cada indicador. En cada programa realizaremos las peticiones a la API del ISTAC almacenando la URL de la petición y obteniendo en formato JSON todos los datos que nos interesa recopilar. Una vez obtenidos, se procesarán para insertar en InfluxDB únicamente los que sean de interés, para lo que se necesita la librería python-influxdb [14], que da soporte a la conexión con la base de datos y permite realizar operaciones sobre ella.

En cuanto a la visualización de los datos almacenados en InfluxDB utilizaremos las aplicaciones para la visualización de métricas basadas en series de tiempo Chronograf [7] y Grafana [10]. Chronograf es una herramienta desarrollada por el equipo de InfluxData que realiza la función de interfaz gráfica administrativa de InfluxDB permitiendo la administración y gestión de las bases de datos además de visualizar los datos, crear dashboards, etc. Mientras que Grafana consultará las bases de datos de InfluxDB para visualizar los datos almacenados mediante la elaboración de dashboards únicamente. Grafana admite el despliegue para ser utilizada como aplicación web, haciendo de capa externa para el acceso a los datos de forma segura y que podrán utilizar los usuarios para generar los cuadros de mando que necesiten.

Fuera del desarrollo de nuestro proyecto se han utilizado para la realización de pruebas dos herramientas secundarias desarrolladas por el equipo de InfluxData [5]. Las herramientas son Telegraf [8] y Kapacitor [9]. Telegraf es un agente para recopilar y reportar métricas y datos, mientras que Kapacitor es un motor de procesamiento de datos de transmisión en tiempo real. La aplicación Chronograf constituye una interfaz gráfica que complementa tanto a InfluxDB, como a Telegraf y Kapacitor.

Como complemento para el desarrollo de este proyecto se ha facilitado una máquina virtual del IaaS perteneciente a la Universidad de La Laguna en la que hemos instalado todo el software necesario para el desarrollo del proyecto. La

finalidad de esta máquina virtual es la de disponer de un servidor donde se encuentre accesible la solución.

## 2.1.14 API ISTAC

Para la obtención de datos mediante la API del ISTAC [2], se realizarán peticiones que serán devueltas en formato JSON [13]. La API dispone de información muy útil sobre su funcionamiento de forma que el desarrollador pueda aprender de una manera sencilla a hacer uso de la plataforma y acceder a los datos que ofrece mediante consultas explicadas de una manera fácil. La plataforma dispone de un swagger, herramienta que facilita la construcción de las urls para realizar las peticiones de los datos, haciendo esta tarea muy intuitiva.

Actualmente, la API cuenta con información sobre indicadores estadísticos, entre los que se encuentran los que consultaremos en el desarrollo de nuestro proyecto. Cuenta con variables de granularidad que permiten filtrar datos por diferentes categorías. Las granularidades que se pueden especificar son:

- **Granularidad geográfica**, nos permite clasificar los datos de medidas que vayamos a recopilar según su granularidad geográfica: regional, insular, provincial y municipal (dependiendo del indicador se puede limitar solamente a Canarias o extenderse a nivel nacional).
- **Granularidad temporal**, nos permite clasificar los datos mediante periodicidades anuales, mensuales o trimestrales.
- Las **medidas** a consultar para nuestras series de tiempo podrán ser Absolutas, Tasa de Variación Anual, Tasa de Variación Interperiódica, Variación Anual y Variación Interperiódica. Las medidas serán los datos que analizaremos para ver su evolución a lo largo del tiempo.

Parámetro	Valor	Descripción	Tipo del Parámetro
indicatorCode	ALOJATUR_ABIERTOS	Código del indicador a obtener.	path
representation	MEASURE[ABSOLUTE]	Permite filtrar las observaciones mediante el valor de las mismas. Ejemplo: representation=GEOGRAPHICAL[35003 35005],MEASURE[ABSOLUTE].	query
granularity	TIME[YEARLY],GEOGRAPHICAL[MUNICIPALITIE	Permite filtrar las observaciones mediante las granularidades de las mismas. Ejemplo: granularity=GEOGRAPHICAL[MUNICIPALITIES PROVINCES],TIME[MONTHLY].	query
fields		Permite personalizar la respuesta mediante la exclusión de campos. Los posibles valores son: "observationsMetadata". Ejemplo: fields=observationsMetadata.	query

[Pruébalo!](#) [Ocultar Respuesta](#)

curl -X GET --header 'Accept: application/json' 'https://www3.gobiernodecanarias.org/istac/api/indicators/v1.0/indicators/ALOJATUR\_ABIERTOS/data?representation=MEASURE[ABSOLUTE]&granularity=TIME[YEARLY],GEOGRAPHICAL[MUNICIPALITIES|PROVINCES]&fields=observationsMetadata'

URL de la Solicitud

https://www3.gobiernodecanarias.org/istac/api/indicators/v1.0/indicators/ALOJATUR\_ABIERTOS/data?representation=MEASURE[ABSOLUTE]&granularity=TIME[YEARLY],GEOGRAPHICAL[MUNICIPALITIES|PROVINCES]&fields=observationsMetadata

Cuerpo de la Respuesta

```
{
  "kind": "indicators#indicatorData",
  "selfLink": "https://www3.gobiernodecanarias.org/istac/api/indicators/v1.0/indicators/ALOJATUR_ABIERTOS/data?representation=MEASURE[ABSOLUTE]&granularity=TIME[YEARLY],GEOGRAPHICAL[MUNICIPALITIES|PROVINCES]&fields=observationsMetadata",
  "observations": [
    {
      "date": "2017-01-01",
      "value": 1000000,
      "unit": "habitantes",
      "type": "absoluta"
    },
    {
      "date": "2017-07-01",
      "value": 1000000,
      "unit": "habitantes",
      "type": "absoluta"
    },
    {
      "date": "2018-01-01",
      "value": 1000000,
      "unit": "habitantes",
      "type": "absoluta"
    },
    {
      "date": "2018-07-01",
      "value": 1000000,
      "unit": "habitantes",
      "type": "absoluta"
    },
    {
      "date": "2019-01-01",
      "value": 1000000,
      "unit": "habitantes",
      "type": "absoluta"
    },
    {
      "date": "2019-07-01",
      "value": 1000000,
      "unit": "habitantes",
      "type": "absoluta"
    },
    {
      "date": "2020-01-01",
      "value": 1000000,
      "unit": "habitantes",
      "type": "absoluta"
    },
    {
      "date": "2020-07-01",
      "value": 1000000,
      "unit": "habitantes",
      "type": "absoluta"
    },
    {
      "date": "2021-01-01",
      "value": 1000000,
      "unit": "habitantes",
      "type": "absoluta"
    },
    {
      "date": "2021-07-01",
      "value": 1000000,
      "unit": "habitantes",
      "type": "absoluta"
    }
  ]
}
```

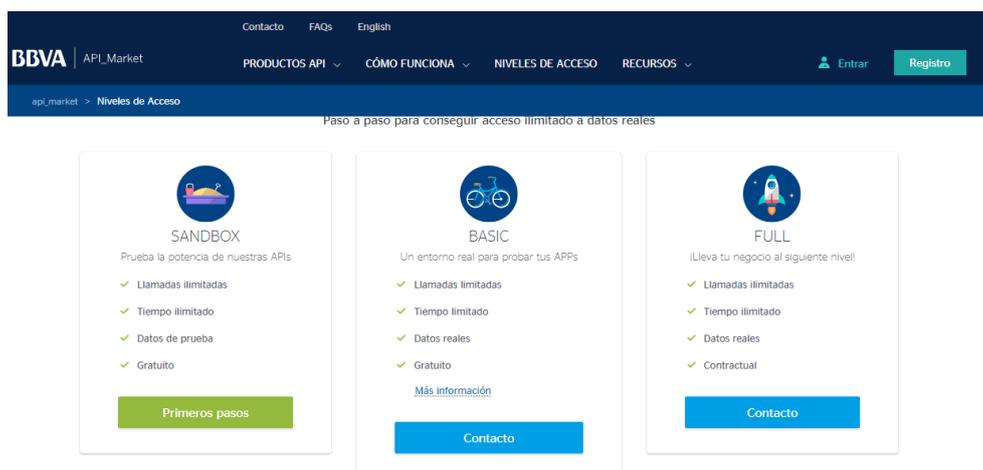
Figura 2.1 Ejemplo de consulta en el swagger de la API del ISTAC.

## 2.1.15 API BBVA MARKET

Además de la API del ISTAC se propuso estudiar la API BBVA MARKET [12] para determinar si era viable su utilización en el desarrollo de nuestro proyecto y recopilar datos relacionados con el área de turismo.

Entre el catálogo de productos APIs que ofrece la API BBVA MARKET la que nos interesa utilizar para el proyecto es BBVA España PayStats. El producto PayStats ofrece al desarrollador estadísticas agregadas y anonimizadas de millones de transacciones realizadas con tarjetas BBVA y de cualquier otro banco en TPVs BBVA para crear un mapa virtual permitiendo analizar los hábitos de consumo de clientes. La información disponible es actualizada semanalmente.

Después de estudiarla con detenimiento se ha llegado a la conclusión de que su uso no sería viable, ya que no ofrece un servicio gratuito con la cantidad suficiente de datos como para llevar a cabo experimentos que permitan la visualización de series históricas. Ofrece una versión de prueba SANDBOX para familiarizarse con su uso, pero los datos que ofrece no son datos reales. Finalmente dispone de 2 versiones BASIC y FULL. La primera opción BASIC ofrece un servicio gratuito y acceso a datos reales pero el tiempo de uso que se le permite al desarrollador es limitado, al igual que el número de consultas que pueda realizar. Por último la opción FULL ofrece un servicio de pago mediante contrato con la compañía BBVA con acceso a datos reales y tiempo de uso sin limitaciones de ningún tipo, lo que la deja fuera del alcance del proyecto.



The screenshot shows the BBVA API Market website with a dark blue header. The header includes navigation links for 'Contacto', 'FAQs', and 'English'. Below the header, there are dropdown menus for 'PRODUCTOS API', 'CÓMO FUNCIONA', 'NIVELES DE ACCESO', and 'RECURSOS'. A 'Registro' button is visible in the top right corner. The main content area is titled 'Niveles de Acceso' and features three cards representing different access levels: 'SANDBOX', 'BASIC', and 'FULL'. Each card lists its features and includes a call-to-action button.

Nivel de Acceso	Descripción	Características	Acción
SANDBOX	Prueba la potencia de nuestras APIs	<ul style="list-style-type: none"><li>✓ Llamadas ilimitadas</li><li>✓ Tiempo ilimitado</li><li>✓ Datos de prueba</li><li>✓ Gratuito</li></ul>	Primeros pasos
BASIC	Un entorno real para probar tus APPs	<ul style="list-style-type: none"><li>✓ Llamadas limitadas</li><li>✓ Tiempo limitado</li><li>✓ Datos reales</li><li>✓ Gratuito</li></ul>	Contacto
FULL	¡Lleva tu negocio al siguiente nivel!	<ul style="list-style-type: none"><li>✓ Llamadas ilimitadas</li><li>✓ Tiempo ilimitado</li><li>✓ Datos reales</li><li>✓ Contractual</li></ul>	Contacto

Figura 2.2 Opciones de licencia que ofrece API BBVA MARKET.

## 2.2 Antecedentes de software

En la actualidad ante la proliferación masiva de datos que se ha producido en los últimos años hace que resulte necesaria la utilización de herramientas que nos ayuden a procesar esa información para ser presentada al usuario final de una manera comprensible.

Podemos encontrar herramientas similares a las que hemos utilizado en el desarrollo de este proyecto. Con ellas podemos visualizar datos que abarcan diferentes áreas por lo que sería de gran interés realizar un análisis de estas

herramientas que puede resultar de gran utilidad para el desarrollo de este proyecto y poder así ofrecer las mismas posibilidades o al menos lo más aproximadas posible.

Un antecedente de software es PULSEC [3], un trabajo de fin de grado de la Ingeniería Informática de la Universidad de La Laguna realizado durante el curso 2013/2014. Se trata de una aplicación móvil para Firefox OS destinada a la comunicación de resúmenes de información de coyuntura económica procedentes de indicadores del ISTAC. Su objetivo es la visualización de los datos de manera gráfica facilitando en la medida de lo posible la comprensión de la información. A diferencia de nuestro proyecto en primer lugar se debe descargar el indicador que el usuario selecciona para posteriormente filtrar los datos en base a los criterios seleccionados por el usuario. Cabe destacar que para el desarrollo del proyecto PULSEC no se encontraba disponible la API del ISTAC con las funcionalidades que ofrece en la actualidad, funcionalidades que nos permiten indicar concretamente los datos que deseamos obtener. En esta aplicación se debe seleccionar el indicador que se desea consultar, una vez se descarguen los datos de dicho indicador podremos filtrar específicamente los datos obtenidos a través de un menú lateral de opciones. Dispone también de una serie de botones que nos permite cambiar el modo de representación gráfica para la visualización de los datos.



Figura 2.3 Pantalla principal PULSEC.

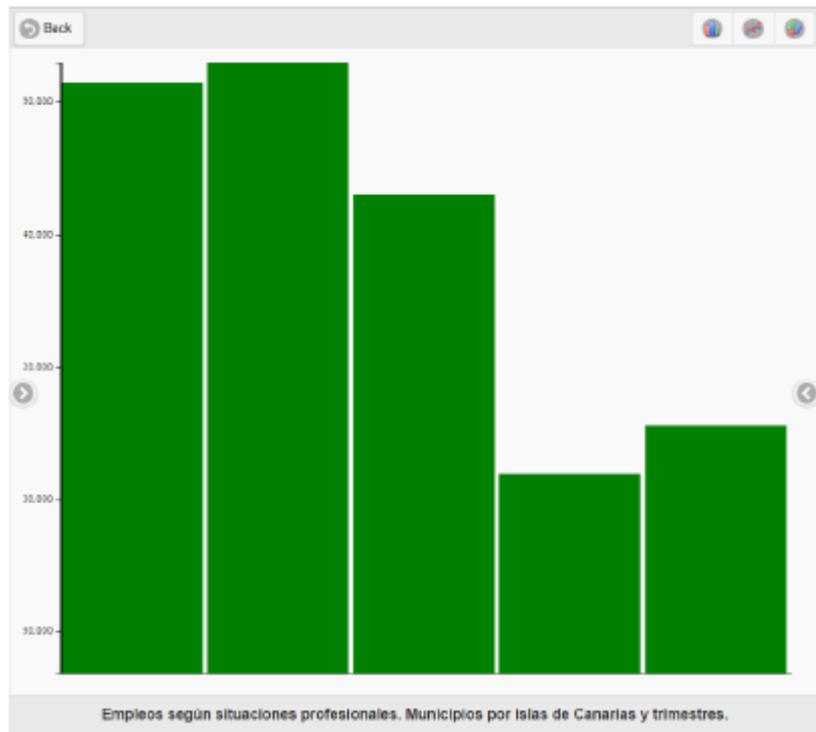


Figura 2.4 Resultados de consulta en PULSEC

En la misma línea se puede citar VISTac-IMAS-Can [4], es también un trabajo de fin de grado de la Ingeniería Informática de la Universidad de La Laguna realizado durante el curso 2014/2015. Este proyecto tiene por objetivo el desarrollo de una aplicación multiplataforma que permita la visualización de datos estadísticos recogidos del ISTAC relacionados con el área de medioambiente y sostenibilidad del Gobierno de Canarias. Los datos al igual que en nuestro proyecto se recogen mediante peticiones y la representación de los datos se muestran tanto de manera numérica como gráfica.

Al iniciar la aplicación se visualizará un listado de categorías de indicadores disponibles en la plataforma, una vez seleccionada la categoría se mostrarán los indicadores correspondientes a la categoría elegida, se deberá seleccionar un indicador para que sean mostradas a continuación las ubicaciones y fechas disponibles permitiendo filtrar los datos para la visualización basándonos en las preferencias establecidas por el usuario.



Figura 2.5 Ejemplo de consulta VIStac.

## **Parte II**

# **Cuerpo principal. Descripción del trabajo**

# Capítulo 3

## Indicadores Estadísticos

### 3.1 Introducción

Se denominan indicadores estadísticos a una serie de datos estadísticos recogidos en relación con algún tema de interés analítico. A partir de estos datos podemos analizar y estudiar su variación a lo largo del tiempo, así como la tendencia con la que pueden variar en el futuro en relación con sucesos actuales o incluso comparando los datos entre distintos indicadores para poder detectar e identificar posibles despuntes que se puedan producir entre indicadores.

Los indicadores estadísticos que incluyen el desarrollo de nuestro proyecto abarcarán las áreas de economía general, sector servicios, turismo, hostelería, transporte y comunicaciones. Los datos que recopilemos pueden limitarse geográficamente al archipiélago canario o extenderse a nivel nacional dependiendo del indicador que se haya seleccionado.

De cada indicador se obtendrán los datos de medidas absolutas, tasas de variación anual, tasas de variación interperiódica, variación anual y variación interperiódica correspondientes a cada ubicación geográfica y cada fecha disponible que podrán ser anuales, mensuales o trimestrales.

### 3.2 Selección de indicadores

A continuación listaremos los indicadores estadísticos recogidos por el Instituto Canario de Estadística ISTAC [2], y que han resultado de interés para conocer las distintas áreas de economía general, sector servicios, hostelería, turismo, transporte y comunicaciones.

- **Economía general:**

- Cuentas económicas (País España y Comunidad Autónoma Canarias):
  - Índice de volumen encadenado del PIB (variaciones reales): Índice de volumen encadenado corregido de estacionalidad y calendario del PIB (Base 2010).
  - PIB a precios corrientes: PIB a precios corrientes corregido de estacionalidad y calendario (Base 2010).
- Precios, consumo e inversión (España):
  - Índice de precios de consumo (IPC). General: El Índice de Precios de Consumo (IPC) tiene como objetivo medir la evolución de los precios de los bienes y servicios de consumo adquiridos por los hogares residentes en España.

- Índice de precios de consumo (IPC). Subyacente: El Índice de Precios de Consumo (IPC) tiene como objetivo medir la evolución de los precios de los bienes y servicios de consumo adquiridos por los hogares residentes en España. El IPC subyacente se define como el IPC general sin alimentos no elaborados ni productos energéticos.
- Empresas y centros de trabajo (España y Canarias):
  - Empresas inscritas en la Seguridad Social (Canarias): Número de empresas de alta en la Seguridad Social en el último día del trimestre.
  - Sociedades mercantiles. Creadas (España): Número de sociedades mercantiles creadas.
- **Sector servicios:**
  - Índice de cifra de negocios (IASS). Servicios (España): Índice de cifra de negocio del sector servicios (Base 2015) por comunidades autónomas.
  - Índice de ocupación (IASS). Servicios (España): Índice de ocupación del sector servicios (Base 2015) por comunidades autónomas.
  - Afiliaciones a la Seguridad Social. Servicios (Canarias): Puestos de trabajo registrados en la Seguridad Social y referidos al último día del último mes del trimestre. El lugar de referencia es el domicilio de cotización de la empresa o del autónomo. Se contabilizan los puestos ocupados por personas en situación de alta, contemplando las afiliaciones múltiples de una misma persona.
  - Empleo registrado. Servicios (Canarias): Puestos de trabajo registrados en el sector de servicios, referidos al último día del último mes de cada trimestre.
  - Empresas inscritas en la Seguridad Social. Servicios (Canarias): Número de empresas de alta en la Seguridad Social en el último día del trimestre.
  - Turismo y hostelería (Canarias):
    - Alojamientos turísticos abiertos: Número de hoteles y apartamentos turísticos abiertos.
    - Plazas ofertadas por alojamientos turísticos: Número de plazas ofertadas por hoteles y apartamentos turísticos.
    - Viajeros entrados en alojamientos turísticos: Número de viajeros entrados en hoteles y apartamentos turísticos.
    - Pernoctaciones en alojamientos turísticos: Número de pernoctaciones en hoteles y apartamentos turísticos.
    - Tasa de ocupación por plazas: Tasa de ocupación por plazas en hoteles y apartamentos turísticos.

- Población turística equivalente en alojamientos turísticos: Población turística equivalente en hoteles y apartamentos turísticos.
- Empleo registrado. Hostelería: Puestos de trabajo registrados en la sección de actividad de la hostelería, referidos al último día del último mes de cada trimestre.
- Transporte y comunicaciones (Canarias):
  - Matriculación de vehículos: Número de vehículos matriculados durante el periodo de referencia.
  - Matriculación de vehículos. Turismos: Número de turismos matriculados durante el periodo de referencia.
  - Tráfico aéreo comercial. Movimiento de aeronaves: Despegue o aterrizaje de una aeronave en tráfico comercial en un aeropuerto.
  - Tráfico aéreo comercial. Movimiento de pasajeros: Personas embarcadas o desembarcadas en tráfico comercial en un aeropuerto.
  - Tráfico aéreo comercial. Movimiento de mercancías: Peso de la carga y el correo embarcados o desembarcados en tráfico comercial en un aeropuerto.

En el siguiente apartado 3.3 mostraremos una tabla (tabla 3.1) donde se muestran las características de los datos que serán recopilados para cada uno de los indicadores, en concreto, la fecha base (año de partida de datos disponibles para el indicador), granularidad geográfica (países, comunidades autónomas, islas, provincias y municipios) que dependiendo del indicador se limitará al archipiélago canario o se extenderá a nivel nacional (España), granularidad temporal (anual, mensual o trimestral) de los datos disponibles y el tipo de medida que podrán ser absolutas (ABS), tasa de variación anual (TVA), tasa de variación interperiódica (TVI), variación anual (VA) y variación interperiódica (VI).

### 3.3 Tabla de indicadores y sus características

TABLA DE INDICADORES Y SUS CARACTERÍSTICAS				
Indicador	Fecha Base	Granularidad geográfica	Granularidad temporal	Tipos de medidas
AFILIACIONES_SERVICIOS	2012	Comunidades autónomas, Islas y Municipios	Trimestral	ABS, TVA, TVI, VA y VI
ALOJATUR_ABIERTOS	2009	Comunidades autónomas, Islas y Municipios	Anual y Mensual	ABS, TVA, TVI, VA y VI
ALOJATUR_PERNOCTACIONES	2009	Comunidades autónomas, Islas y Municipios	Anual y Mensual	ABS, TVA, TVI, VA y VI
ALOJATUR_PLAZAS_OCUPACION	2009	Comunidades autónomas, Islas y Municipios	Anual y Mensual	ABS, VA y VI
ALOJATUR_PLAZAS_OFERTADAS	2009	Comunidades autónomas, Islas y Municipios	Anual y Mensual	ABS, TVA, TVI, VA y VI
ALOJATUR_POBLACION_TUR_ETC	2009	Comunidades autónomas, Islas y Municipios	Anual	ABS, TVA, TVI, VA y VI
ALOJATUR_VIAJEROS_ENTRADOS	2009	Comunidades autónomas, Islas y Municipios	Anual y Mensual	ABS, TVA, TVI, VA y VI
EMPLEO_REGISTRADO_HOSTELERÍA	2009	Comunidades autónomas, Islas y Municipios	Anual y Trimestral	ABS, TVA, TVI, VA y VI
EMPLEO_REGISTRADO_SERVICIOS	2009	Comunidades autónomas, Islas y Municipios	Anual y Trimestral	ABS, TVA, TVI, VA y VI
EMPRESAS_SEGURIDAD_SOCIAL	2012	Comunidades autónomas, Islas y Municipios	Trimestral	ABS, TVA, TVI, VA y VI
EMPRESAS_SEGURIDAD_SOCIAL_SERVICIOS	2012	Comunidades autónomas, Islas y Municipios	Trimestral	ABS, TVA, TVI, VA y VI
IASS_CIFRA_NEGOCIO	2005	Países y Comunidades autónomas	Mensual	ABS, TVA y TVI
IASS_OCUPACION	2005	Países y Comunidades autónomas	Mensual	ABS, TVA y TVI
IPC	2002	Países, Provincias y Comunidades autónomas	Mensual	ABS, TVA y TVI
IPC_SUBYACENTE	2002	Países y Comunidades autónomas	Mensual	ABS, TVA y TVI
MATRICULA_VEHICULOS	2003	Comunidades autónomas, Islas y Municipios	Anual y Mensual	ABS, TVA, TVI, VA y VI
MATRICULA_VEHICULOS_TURISMOS	2003	Comunidades autónomas, Islas y Municipios	Anual y Mensual	ABS, TVA, TVI, VA y VI
PIB_PM_CORRIENTE	2000	Países y Comunidades autónomas	Trimestral	ABS, TVA, TVI, VA y VI
PIB_PM_VOLUMEN	2000	Países y Comunidades autónomas	Trimestral	ABS, TVA y TVI
SOCIEDADES_MERCANTILES_CREADAS	1995	Países, Provincias y Comunidades autónomas	Anual y Mensual	ABS, TVA, TVI, VA y VI
TRAFICO_AEREO_AVIONES	1999	Comunidades autónomas e Islas	Anual y Mensual	ABS, TVA, TVI, VA y VI
TRAFICO_AEREO_MERCANCIAS	1999	Comunidades autónomas e Islas	Anual y Mensual	ABS, TVA, TVI, VA y VI
TRAFICO_AEREO_PASAJEROS	1999	Comunidades autónomas e Islas	Anual y Mensual	ABS, TVA, TVI, VA y VI

Tabla 3.1 Indicadores estadísticos y sus características.

# Capítulo 4

## Tratamiento y almacenamiento de los datos

### 4.1 Introducción al tratamiento y almacenamiento de los datos

En el presente capítulo explicaremos cómo se obtuvieron los datos de la API del ISTAC, y los que se almacenan en nuestra base de datos temporal InfluxDB, también el diseño que se ha elegido para la base de datos.

Se ha utilizado el portal web de indicadores del ISTAC para generar las URLs con las peticiones de consultas especificando el indicador de información a consultar (alojamientos turísticos abiertos, sociedades mercantiles creadas, empleo registrado en hostelería, etc.), la granularidad temporal (anual, mensual o trimestral) y el tipo de medida para las observaciones (absoluta (ABS), tasa de variación anual (TVA), tasa de variación interperiódica (TVI), variación anual (VA) y variación interperiódica (VI)).

Se generan dos tipos de url:

- **Metadata:** recogerán los valores de etiquetas que almacenaremos en InfluxDB. Estas nos servirán para filtrar las opciones de visualización de los datos en Chronograf y Grafana, las que se han tenido en cuenta son:
  - Código Geográfico (almacena el código que identifica a cada ubicación geográfica).
  - Nombre Geográfico (almacena el nombre correspondiente a cada ubicación).
  - Código Tiempo (almacena el código que identifica a cada fecha).
  - Nombre Tiempo (almacena el nombre correspondiente a cada fecha).
  - Tipo Medida (almacena el nombre correspondiente a cada tipo de medida).

**Nota:** Se almacena el código geográfico y nombre geográfico, aunque son equivalentes. Esto se hace porque en las aplicaciones Chronograf y Grafana al usuario le resultará más fácil identificar una ubicación por su nombre en lugar de su código. Por ejemplo, es más fácil identificar “Canarias” por su nombre que por su código “ES70”. El mismo caso sucede para los datos Código Tiempo y Nombre Tiempo. Es más fácil identificar por ejemplo una fecha mensual con el nombre “2018 Enero” que por su código “2018M01”. Resumiendo, se almacenan los dos tipos de datos correspondientes a ubicaciones geográficas y datos de tiempo y el usuario podrá elegir la opción de visualización que más cómoda le resulte a la hora de seleccionar las

opciones de búsqueda.

- **Data:** Recogerán todos los datos correspondientes a las observaciones para cada tipo de medida (ABS, TVA, TVI, VA y VI) que cada indicador tenga disponibles, cada tipo de granularidad temporal (anual, mensual y trimestral) para todas las ubicaciones geográficas disponibles.
  - **Dato Observación:** Almacena todos los valores correspondientes para cada tipo de medida, cada ubicación geográfica y cada fecha.

Se ha hecho un estudio de los lenguajes de programación a los que da soporte InfluxDB decantándonos por Python (Versión 3) por su simplicidad, menor complejidad y menor número de líneas de código necesarias para realizar operaciones con InfluxDB respecto a otros. Para poder interactuar con InfluxDB desde un script Python se debe instalar la librería específica `python-influxdb`.

Una vez tengamos generadas las URLs con las peticiones de consulta para cada indicador, se procederá a la ejecución de programas desarrollados en Python, un script para cada indicador que obtendrán los datos en formato JSON de las URLs generadas, se procesarán y parsearán los datos que nos interesan recopilar para finalmente ser almacenados en la base de datos (véase código de ejemplo de script en Python en Apéndice A).

## 4.2 Introducción a InfluxDB

InfluxDB es una base de datos de series temporales diseñada para manejar altas cargas de escritura y consultas. Aunque InfluxDB está diseñado para la gestión de grandes volúmenes de datos con marcas de tiempo, es una tecnología diseñada para bases de datos históricas como las que nos ocupan. Los principales usos de InfluxDB están en la monitorización de servidores, datos de sensores IoT, etc.

A continuación listaremos algunas de las características que InfluxDB admite actualmente que lo convierten en una excelente opción para trabajar con datos de series de tiempo:

- Almacén de datos de alto rendimiento personalizado escrito específicamente para datos de series temporales. El motor TSM permite alta velocidad de ingesta y comprensión de los datos.
- Escrito completamente en Go [16]. Se compila en un solo binario sin dependencias externas.
- API de HTTP de escritura y consulta simple de alto rendimiento.
- Los complementos admiten otros protocolos de ingestión de datos como Graphite, collectd y OpenTSDB.

- Lenguaje de consulta expresivo SQL-like hecho a la medida para consultar fácilmente datos agregados.
- Las etiquetas permiten indexar series para consultas rápidas y eficientes.
- Las políticas de retención expiran automáticamente los datos obsoletos.
- Las consultas continuas calculan automáticamente los datos agregados para que las consultas frecuentes sean más eficientes.

InfluxDB ofrece también una versión “**Enterprise**” no gratuita. Esta versión ofrece clústeres InfluxDB Enterprise altamente escalables en su infraestructura con una interfaz de usuario de administración.

La versión de InfluxDB Enterprise se utiliza para:

- Control de clúster.
- Administrar consultas.
- Administrar usuarios.
- Explorar y visualizar datos.

Para el desarrollo de nuestro proyecto utilizaremos la versión de InfluxDB gratuita de código abierto.

#### 4.2.1 Estructura de datos en InfluxDB

A continuación detallaremos los niveles de jerarquía en los que se divide InfluxDB.

- **Bases de datos.**
  - Las bases de datos constituyen el primer nivel de jerarquía en la distribución de datos en InfluxDB, son estructuras virtuales que disponen de configuraciones totalmente independientes de las demás. Corresponden a los directorios donde se almacenan los datos que contienen las series temporales asociadas.
- **Políticas de retención (RP).**
  - InfluxDB permite establecer políticas de retención en sus bases de datos. Este nivel de jerarquía se sitúa entre la distribución de las series y la base de datos. Con ello se podrá configurar entradas de datos diferentes en una misma base de datos con diferentes consolidaciones temporales.
  - Las políticas de retención ofrecen información acerca de cómo son almacenados los datos en relación a intervalos de tiempo dentro de la base de datos:
    - Determinan la duración máxima de los datos en la base de datos.

- Determinan el periodo de compactación y escritura en disco de los datos en memoria.

- **Measurements.**

Los Measurements actúan como un contenedor para las etiquetas, los campos y la columna de tiempo. El nombre del measurement es la descripción de los datos que están almacenados en los campos asociados. Su objetivo no es otro que recopilar distintas métricas dentro de un mismo grupo relacionado. Estas series temporales están ideadas de tal forma que en cada medida se puedan obtener distintas métricas.

Para nuestro proyecto en los Measurements es donde definiremos el nombre del indicador a consultar (alojamientos turísticos abiertos, empleo registrado en hostelería, sociedades mercantiles creadas, etc.).

- **Series de datos temporales.**

- **Time (tiempo o timestamp).**

- Probablemente **time** sea el factor más importante en este tipo de bases de datos ya que indica el momento en que la métrica ha sido recogida en el servidor. InfluxDB permite una resolución de hasta nanosegundos.
- InfluxDB espera que se le indique el momento de tiempo asociado al dato con el formato AAAA-MM-DDTHH:MM:SS.nnnnnnnnZ.
- Para nuestro proyecto recopilaremos datos de años, meses y trimestres completos:
  - ❖ **Casos anuales:** introducimos todos los años de los datos obtenidos del ISTAC y fijamos para cada año el mes primero Enero, el día 1 y la hora 00:00:00.
  - ❖ **Casos mensuales:** introducimos los años y meses obtenidos del ISTAC y fijamos para cada año y cada mes el día 1 y la hora 00:00:00.
  - ❖ **Casos trimestrales:** introducimos los años y el primer mes de cada trimestre obtenidos del ISTAC y fijamos para cada año y cada primer mes de cada trimestre el día 1 y la hora 00:00:00.

- **Tags (etiquetas).**

- Los **tags** son etiquetas que añaden información sobre la serie de datos temporales. La información que almacenan los tags corresponde a valores indexados dentro de las peticiones de lectura en la base de datos, por tanto, es importante su correcta utilización ya que influyen en el rendimiento. Admiten datos de tipo entero, decimal, booleano o cadenas.

- En los tags almacenaremos los metadatos obtenidos del ISTAC correspondientes al “código geográfico”, “nombre geográfico”, “código de tiempo”, “nombre de tiempo” y el “tipo de medida” para las observaciones.
- **Fields.**
  - Los **fields** representan las diferentes métricas o valores clave dentro del measurement. A diferencia de los tags, los fields no se encuentran indexados dentro de la base de datos. Admiten almacenar datos de tipo entero, decimal, booleano o cadenas.
  - En los fields almacenaremos los datos obtenidos del ISTAC correspondientes a las observaciones de las medidas absoluta, tasa de variación anual, tasa de variación interperiódica, variación anual y variación interperiódica. El nombre del field que almacenará los valores de las observaciones lo llamaremos “Dato Observación”.

#### 4.2.2 Operaciones básicas con InfluxDB

En el presente capítulo mostraremos operaciones básicas con InfluxDB a través de la terminal de consola (Ubuntu 18.04).

- **Ejecutar InfluxDB:**

- **Comando:** `influx -precision rfc3339`
- **Nota:** El argumento “- precisión rfc3339” especifica el formato/precisión de las marcas de tiempo devueltas. El argumento `rfc3339` indicará las marcas de tiempo en el formato RFC3339 (AAAA-MM-DDTHH:MM:SS.nnnnnnnnnZ).
- Si la ejecución del comando se ha realizado correctamente debería mostrar el resultado que podemos observar en la captura siguiente:

```
waldo@waldo-Lenovo-ideapad-320-15IKB:~$ influx -precision rfc3339
Connected to http://localhost:8086 version 1.6.0
InfluxDB shell version: 1.6.0
> █
```

Figura 4.1 Ejecución de InfluxDB en la terminal de consola.

- **Mostrar bases de datos:**
  - **Comando:** show databases

```
> show databases
name: databases
name
----
telegraf
_internal
INDICADORES_(ISTAC)
>
```

Figura 4.2 Mostrar bases de datos en InfluxDB.

- **Seleccionar base de datos.**
  - **Comando:** use “INDICADORES\_(ISTAC)”

```
> use INDICADORES_(ISTAC)
Using database INDICADORES_(ISTAC)
>
```

Figura 4.3 Seleccionar base de datos en InfluxDB.

- **Mostrar medidas.**
  - **Comando:** show measurements

```
> show measurements
name: measurements
name
----
Afiliaciones_A_La_Seguridad_Social_Servicios_Trimestrales
Alojamientos_Turísticos_Abiertos_En_Canarias_Anuales
Alojamientos_Turísticos_Abiertos_En_Canarias_Mensuales
Empleo_Registrado_En_Hostelería_En_Canarias_Anuales
Empleo_Registrado_En_Hostelería_En_Canarias_Trimestrales
Empleo_Registrado_Servicios_Anuales
Empleo_Registrado_Servicios_Trimestrales
Empresas_Inscritas_En_La_Seguridad_Social_Servicios_Trimestrales
Empresas_Inscritas_En_La_Seguridad_Social_Trimestrales
Matriculación_De_Vehículos_Anuales
Matriculación_De_Vehículos_Mensuales
Matriculación_De_Vehículos_Turismos_Anuales
Matriculación_De_Vehículos_Turismos_Mensuales
PIB_A_Precios_Corrientes_Trimestrales
Pernoctaciones_En_Alojamientos_Turísticos_En_Canarias_Anuales
Pernoctaciones_En_Alojamientos_Turísticos_En_Canarias_Mensuales
Plazas_Ofertadas_Por_Alojamientos_Turísticos_En_Canarias_Anuales
Plazas_Ofertadas_Por_Alojamientos_Turísticos_En_Canarias_Mensuales
Población_Turística_Equivalente_En_Alojamientos_Turísticos_En_Canarias_Anuales
Sociedades_Mercantiles_Creadas_Anuales
Sociedades_Mercantiles_Creadas_Mensuales
Tasa_De_Ocupación_Por_Plazas_En_Canarias_Anuales
Tasa_De_Ocupación_Por_Plazas_En_Canarias_Mensuales
Tráfico_Aéreo_Comercial_Movimiento_De_Aeronaves_Anuales
Tráfico_Aéreo_Comercial_Movimiento_De_Aeronaves_Mensuales
Tráfico_Aéreo_Comercial_Movimiento_De_Mercancías_Anuales
Tráfico_Aéreo_Comercial_Movimiento_De_Mercancías_Mensuales
Tráfico_Aéreo_Comercial_Movimiento_De_Pasajeros_Anuales
Tráfico_Aéreo_Comercial_Movimiento_De_Pasajeros_Mensuales
Viajeros_Entrados_En_Alojamientos_Turísticos_En_Canarias_Anuales
Viajeros_Entrados_En_Alojamientos_Turísticos_En_Canarias_Mensuales
Índice_De_Cifra_De_Negocios_(IASS)_Servicios_Mensuales
Índice_De_Ocupación_(IASS)_Servicios_Mensuales
Índice_De_Precios_De_Consumo_(IPC)_General_Mensuales
```

Figura 4.4 Mostrar medidas en InfluxDB.

- **Mostrar claves de etiquetas.**

- **Comando:** show tag keys on “INDICADORES\_(ISTAC)” from “Alojamientos\_Turísticos\_Abiertos\_En\_Canarias\_Anuales”

```
> SHOW TAG KEYS ON "INDICADORES_(ISTAC)" FROM "Alojamientos_Turísticos_Abiertos_En_Canarias_Anuales"
name: Alojamientos_Turísticos_Abiertos_En_Canarias_Anuales
tagKey
-----
Código_Año
Código_Geográfico
Nombre_Año
Nombre_Geográfico
Tipo_Medida
>
```

Figura 4.5 Mostrar claves de etiquetas en InfluxDB.

- **Mostrar valores de etiquetas.**

- **Comando:** show tag values on “INDICADORES\_(ISTAC)” from “Alojamientos\_Turísticos\_Abiertos\_En\_Canarias\_Anuales” with key = “Nombre\_Geográfico”

```
> SHOW TAG VALUES ON "INDICADORES_(ISTAC)" FROM "Alojamientos_Turísticos_Abiertos_En_Canarias_Anuales" WITH KEY = "Nombre_Geográfico"
name: Alojamientos_Turísticos_Abiertos_En_Canarias_Anuales
key      value
---      -
Nombre_Geográfico Adeje
Nombre_Geográfico Antigua
Nombre_Geográfico Arona
Nombre_Geográfico Arrecife
Nombre_Geográfico Breña Baja
Nombre_Geográfico Canarias
Nombre_Geográfico EL Hierro
Nombre_Geográfico Frontera
Nombre_Geográfico Fuencaliente de La Palma
Nombre_Geográfico Fuerteventura
Nombre_Geográfico Gran Canaria
Nombre_Geográfico Granadilla de Abona
Nombre_Geográfico La Gomera
Nombre_Geográfico La Palma
Nombre_Geográfico Lanzarote
Nombre_Geográfico Llanos de Aridane (Los)
Nombre_Geográfico Mogán
Nombre_Geográfico Oliva (La)
Nombre_Geográfico Palmas de Gran Canaria (Las)
Nombre_Geográfico Paso (EL)
Nombre_Geográfico Pinar de EL Hierro (EL)
Nombre_Geográfico Puerto de La Cruz
Nombre_Geográfico Puerto del Rosario
Nombre_Geográfico Pájara
Nombre_Geográfico San Bartolomé de Tirajana
Nombre_Geográfico San Sebastián de La Gomera
Nombre_Geográfico Santa Cruz de La Palma
Nombre_Geográfico Santa Cruz de Tenerife
Nombre_Geográfico Santiago del Teide
Nombre_Geográfico Tegüise
Nombre_Geográfico Tenerife
Nombre_Geográfico Tias
Nombre_Geográfico Valle Gran Rey
```

Figura 4.6 Mostrar valores de etiquetas en InfluxDB.

- **Mostrar toda la información de un indicador específico.**

- **Comando:** select \* from “Alojamientos\_Turísticos\_Abiertos\_En\_Canarias\_Anuales”

```
> select * from "Alojamientos_Turísticos_Abiertos_En_Canarias_Anuales"
name: Alojamientos_Turísticos_Abiertos_En_Canarias_Anuales
time
```

time	Código_Año	Código_Geográfico	Dato_Observación	Nombre_Año	Nombre_Geográfico	Tipo_Medida
2009-01-01T00:00:00Z	2009	35003	29	2009	Antigua	Absoluta
2009-01-01T00:00:00Z	2009	35004	10	2009	Arrecife	Absoluta
2009-01-01T00:00:00Z	2009	35012	143	2009	Mogán	Absoluta
2009-01-01T00:00:00Z	2009	35014	51	2009	Oliva (La)	Absoluta
2009-01-01T00:00:00Z	2009	35015	72	2009	Pájara	Absoluta
2009-01-01T00:00:00Z	2009	35016	65	2009	Palmas de Gran Canaria (Las)	Absoluta
2009-01-01T00:00:00Z	2009	35017	6	2009	Puerto del Rosario	Absoluta
2009-01-01T00:00:00Z	2009	35019	392	2009	San Bartolomé de Tirajana	Absoluta
2009-01-01T00:00:00Z	2009	35024	49	2009	Teguise	Absoluta
2009-01-01T00:00:00Z	2009	35028	157	2009	Tías	Absoluta
2009-01-01T00:00:00Z	2009	35034	48	2009	Yaiza	Absoluta
2009-01-01T00:00:00Z	2009	38001	111	2009	Adeje	Absoluta
2009-01-01T00:00:00Z	2009	38006	117	2009	Arona	Absoluta
2009-01-01T00:00:00Z	2009	38009	26	2009	Breña Baja	Absoluta
2009-01-01T00:00:00Z	2009	38013	18	2009	Frontera	Absoluta
2009-01-01T00:00:00Z	2009	38014	9	2009	Fuencalliente de La Palma	Absoluta
2009-01-01T00:00:00Z	2009	38017	19	2009	Granadilla de Abona	Absoluta
2009-01-01T00:00:00Z	2009	38024	49	2009	Llanos de Aridane (Los)	Absoluta
2009-01-01T00:00:00Z	2009	38027	20	2009	Paso (El)	Absoluta
2009-01-01T00:00:00Z	2009	38028	90	2009	Puerto de La Cruz	Absoluta
2009-01-01T00:00:00Z	2009	38036	33	2009	San Sebastián de La Gomera	Absoluta
2009-01-01T00:00:00Z	2009	38037	9	2009	Santa Cruz de La Palma	Absoluta
2009-01-01T00:00:00Z	2009	38038	27	2009	Santa Cruz de Tenerife	Absoluta
2009-01-01T00:00:00Z	2009	38040	21	2009	Santiago del Teide	Absoluta
2009-01-01T00:00:00Z	2009	38048	8	2009	Valverde	Absoluta
2009-01-01T00:00:00Z	2009	38049	127	2009	Valle Gran Rey	Absoluta
2009-01-01T00:00:00Z	2009	38901	9	2009	Pinar de El Hierro (El)	Absoluta
2009-01-01T00:00:00Z	2009	ES70	1839	2009	Canarias	Absoluta
2009-01-01T00:00:00Z	2009	ES703	34	2009	El Hierro	Absoluta
2009-01-01T00:00:00Z	2009	ES704	164	2009	Fuerteventura	Absoluta
2009-01-01T00:00:00Z	2009	ES705	610	2009	Gran Canaria	Absoluta
2009-01-01T00:00:00Z	2009	ES706	190	2009	La Gomera	Absoluta
2009-01-01T00:00:00Z	2009	ES707	130	2009	La Palma	Absoluta
2009-01-01T00:00:00Z	2009	ES708	271	2009	Lanzarote	Absoluta

Figura 4.7 Mostrar toda la información de un indicador en InfluxDB.

- **Otros comandos:**

- **Crear bases de datos.**
  - **Comando:** create database “db\_name”
- **Borrar bases de datos.**
  - **Comando:** drop database “db\_name”
- **Mostrar políticas de retención.**
  - **Comando:** show retention policies on “db\_name”
- **Crear políticas de retención.**
  - **Comando:** create retention policy “rp\_name” on “db\_name” duration 30d replication 1 default
- **Borrar políticas de retención.**
  - **Comando:** drop retention policy “rp\_name” on “db\_name”
- **Mostrar consultas continuas.**
  - **Comando:** show continuous queries
- **Crear consultas continuas.**
  - **Comando:** create continuous query “cq\_name” on “db\_name”

```
begin select min("field") into "target_measurement" from  
"current_measurement" group by time(30m) end
```

- **Borrar consultas continuas.**
  - **Comando:** drop continuous query "cq\_name" on "db\_name"
- **Mostrar usuarios.**
  - **Comando:** show users
- **Crear usuarios.**
  - **Comando:** create user "username" with password 'password'
- **Crear usuarios administradores.**
  - **Comando:** create user "username" with password 'password' with all privileges
- **Borrar usuario.**
  - **Comando:** drop user "username"
- **Mostrar estadísticas.**
  - **Comando:** show stats
- **Mostrar diagnósticos.**
  - **Comando:** show diagnostics

# Capítulo 5

## Visualización de los datos

### 5.1 Introducción a la Visualización de datos

La visualización de los datos consiste en la representación gráfica de los datos almacenados en nuestra base de datos, se pueden ofrecer diferentes opciones (diagramas de barras, nubes de puntos, etc.), dependiendo de la herramienta especializada que utilicemos.

En el presente capítulo abordaremos la visualización de los datos de los diferentes indicadores estadísticos y la creación de cuadros de mando mediante las aplicaciones especializadas Chronograf y Grafana.

### 5.2 Chronograf

#### 5.2.1 Introducción a Chronograf

Chronograf es una aplicación web de código abierto desarrollada por el equipo de InfluxData cuya función principal es la de visualizar datos de monitoreo, crear reglas de alerta y automatización.

Chronograf además ofrece otras características y funciones que listamos a continuación:

- **Monitoreo de infraestructura:**
  - Ver todos los hosts y sus estados en la infraestructura.
  - Ver las aplicaciones configuradas en cada host.
  - Controlar aplicaciones mediante los paneles pre-generados de Chronograf.
  
- **Gestión de alertas:**
  - Chronograf ofrece una interfaz de usuario para Kapacitor, el marco de procesamiento de datos creado por InfluxData que permite la creación de alertas, ejecución de trabajos ETL y detección de anomalías en los datos.
  - Permite generación de alertas de diferentes tipos en los datos.
  - Permite habilitación y deshabilitación de reglas de alertas existentes.
  - Permite visualizar todas las alertas activas en el panel de alertas.

- Permite el envío de alertas a controladores de eventos admitidos.
- **Visualización de datos:**
  - Permite controlar los datos de las aplicaciones mediante los paneles pre-generados de Chronograf.
  - Creación de paneles propios personalizados con varias opciones de gráficos y variables de plantilla.
  - Investigación de los datos almacenados usando el explorador de datos de Chronograf y las plantillas de consulta.
- **Gestión de bases de datos:**
  - Permite la creación y eliminación de bases de datos y políticas de retención.
  - Visualización de consultas actualmente en ejecución y capacidad de evitar que consultas ineficaces sobrecarguen el sistema.
  - Creación, eliminación y asignación de permisos a usuarios.
- **Soporte multi-organización y multi-usuario:**
  - Creación de organizaciones y asignación de usuarios a esas organizaciones.
  - Restringir acceso a funciones administrativas.
  - Configuración y mantenimiento por parte de los usuarios de tableros únicos para sus organizaciones.

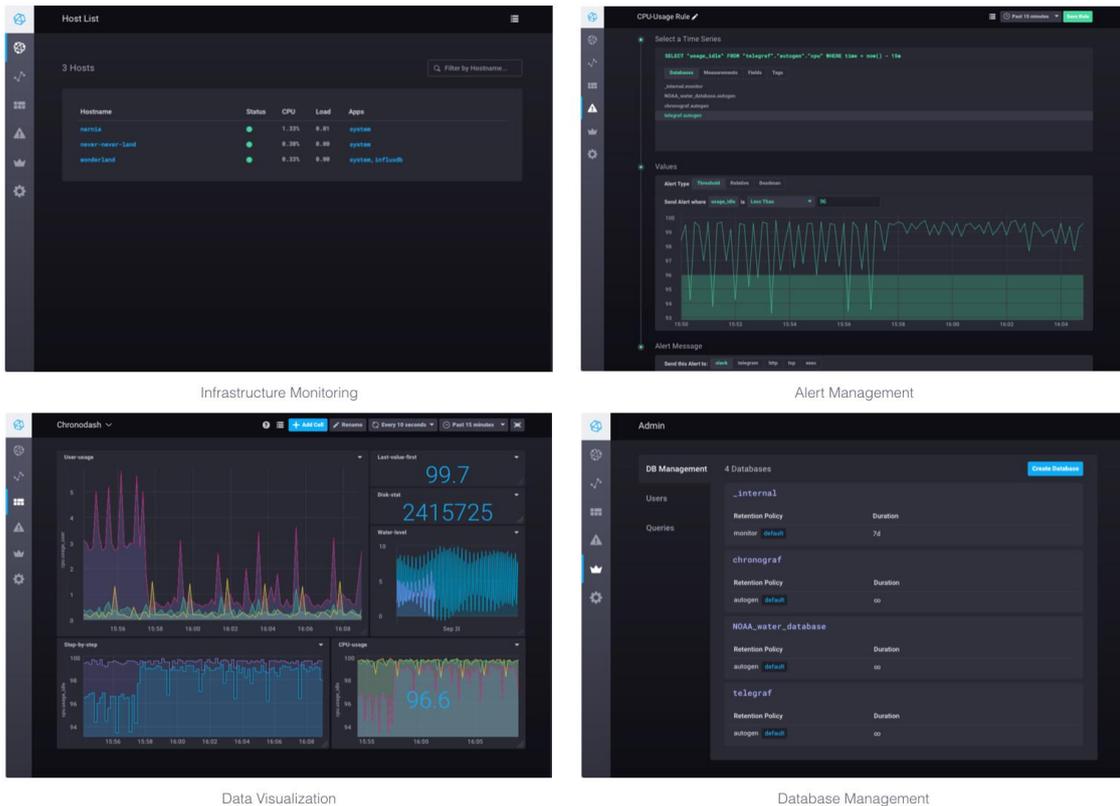


Figura 5.1 Presentación de Chronograf.

## 5.2.2 Configuración de Chronograf

Si la instalación se ha realizado correctamente Chronograf se habrá instalado por defecto en el equipo local (localhost). Chronograf viene configurado para ejecutarse mediante un explorador web a través de la dirección ip o nombre de dominio del equipo o servidor donde se encuentra instalado InfluxDB (localhost en nuestro caso) añadiendo el puerto 8888 que es el puerto que Chronograf trae configurado para acceder a la aplicación.

En el primer acceso se solicita que establezcamos una conexión con la base de datos InfluxDB que tenemos instalada, si bien se permite la creación de una o más conexiones. En el proyecto tenemos habilitadas dos conexiones, una en la máquina local (<http://localhost:8086>) del alumno encargado del desarrollo del proyecto y una segunda conexión en una máquina virtual (<http://IPmaquinavirtual:8086>) facilitada por el IaaS de la Universidad de La Laguna.

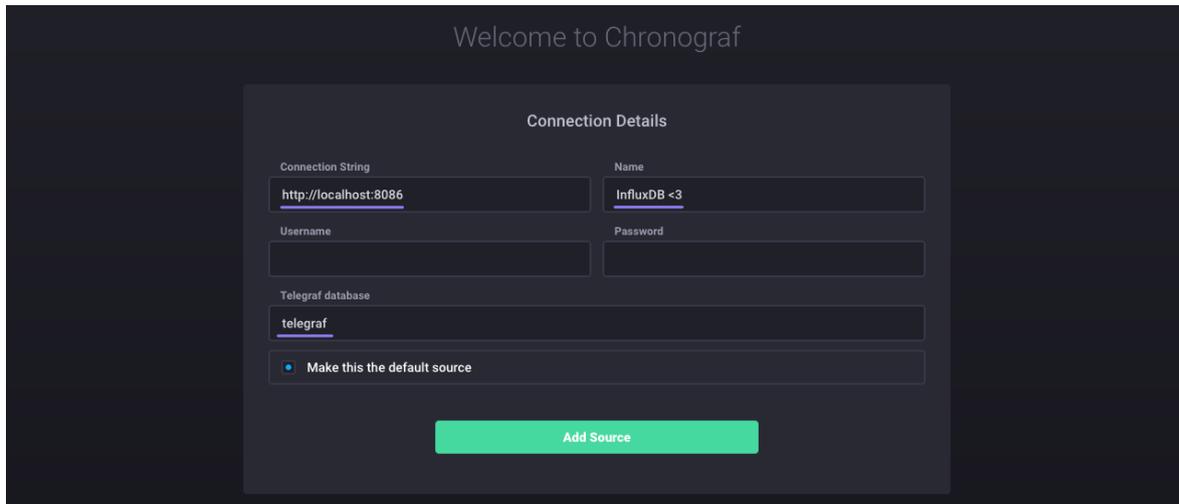


Figura 5.2 Configuración de conexión Chronograf.

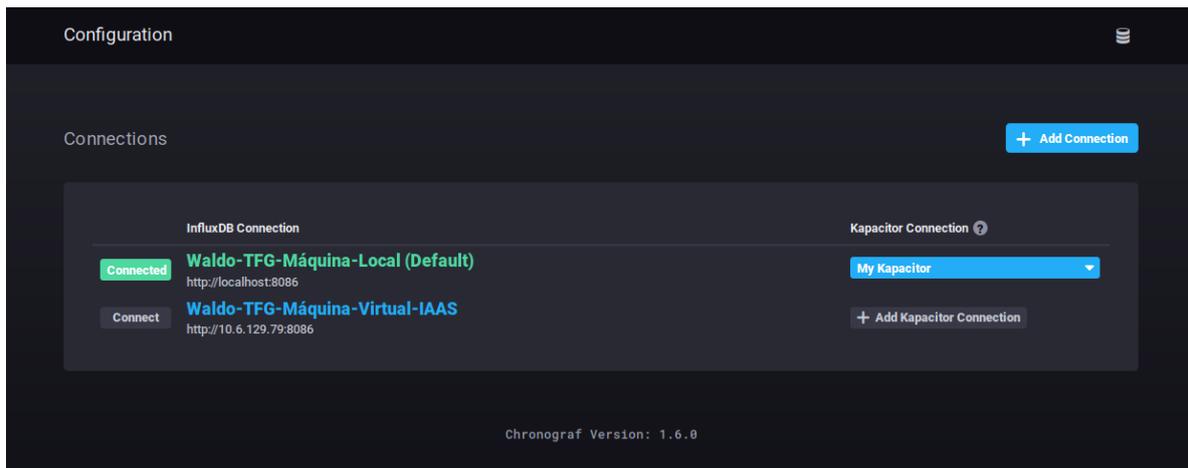


Figura 5.3 Conexiones creadas en Chronograf.

Finalmente para comprobar que InfluxDB está correctamente configurado y conectado seleccionamos Host List y deberá mostrarse algo similar a la siguiente captura:

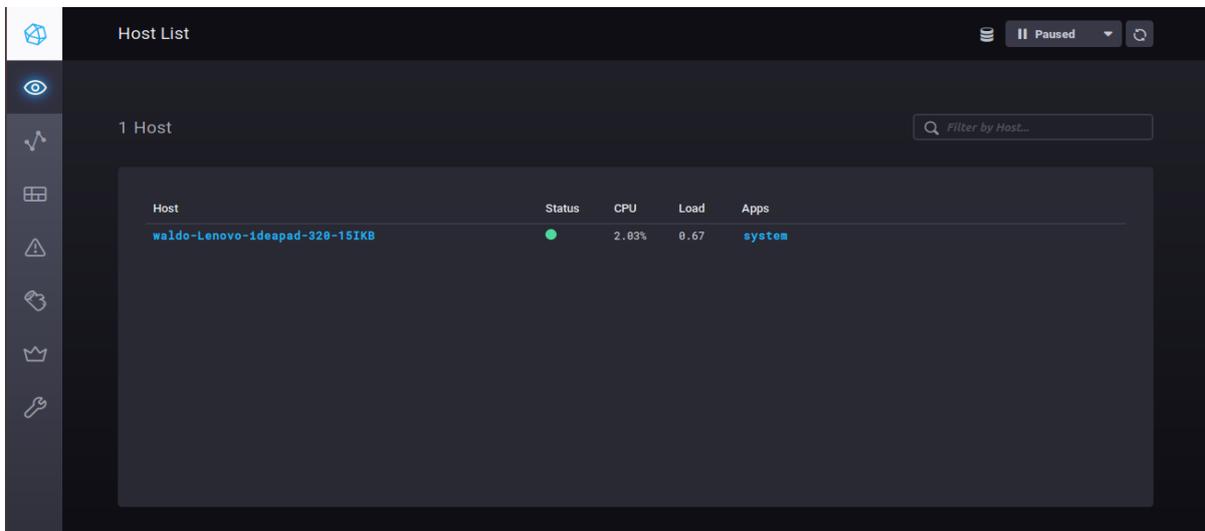


Figura 5.4 Conexión establecida entre Chronograf e InfluxDB.

Si seleccionamos el host que figura en la lista debe mostrar gráficas visualizando datos sobre el sistema que se está ejecutando.

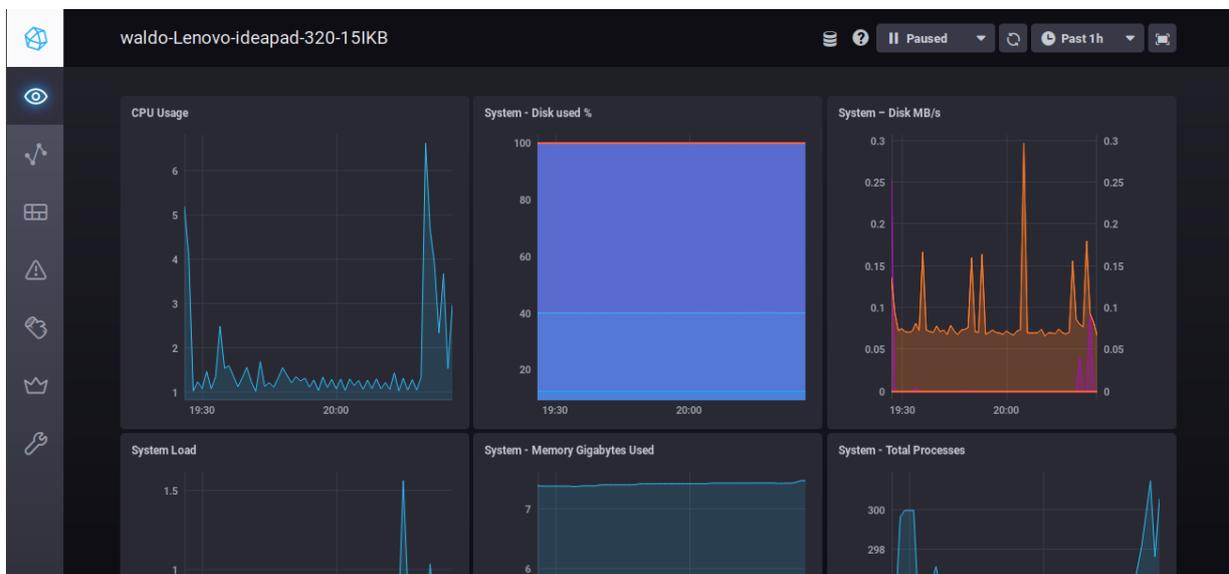


Figura 5.5 Prueba de conexión entre Chronograf e InfluxDB.

### 5.2.3 Caso de ejemplo en Chronograf

A continuación se mostrará un ejemplo para un indicador específico.

Antes de la selección de los datos a mostrar hay que especificar el intervalo de tiempo en el que se encuentren comprendidos. En la parte superior derecha de la aplicación Chronograf seleccionamos en el Date Picker el año 2009 (año base del indicador alojamientos turísticos abiertos), si no se especifica fecha límite para la búsqueda de datos Chronograf establecerá automáticamente como fecha límite la fecha actual del sistema.

Los pasos para crear un dashboard son los siguientes:

- Seleccionamos crear **“dashboard”** en el menú lateral izquierdo.
- Seleccionamos la base de datos **“INDICADORES\_(ISTAC)”**.
- Seleccionamos el indicador de ejemplo **“Alojamientos\_Turísticos\_Abiertos\_En\_Canarias\_Mensuales”**.
- Seleccionamos el nombre geográfico **“Canarias”**.
- Seleccionamos el tipo de medida **“absoluta”**.
- Marcamos la opción **“Dato\_Observación”**.

Si la consulta se ha realizado correctamente debería mostrar el contenido de las siguientes capturas:

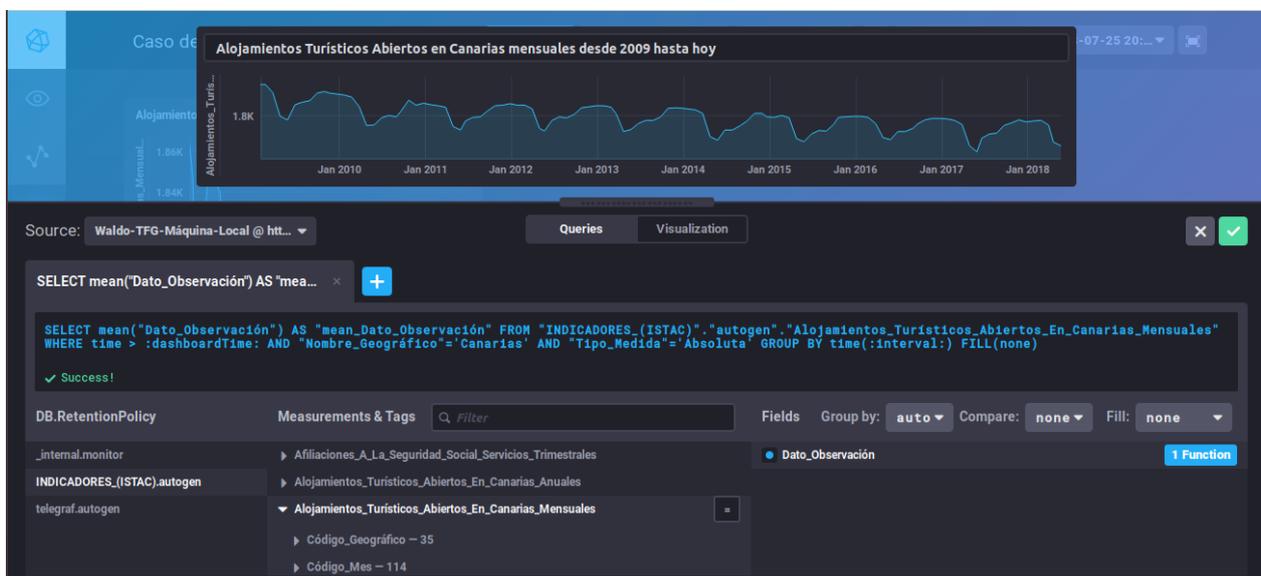


Figura 5.6 Ejemplo de gráfica en Chronograf.

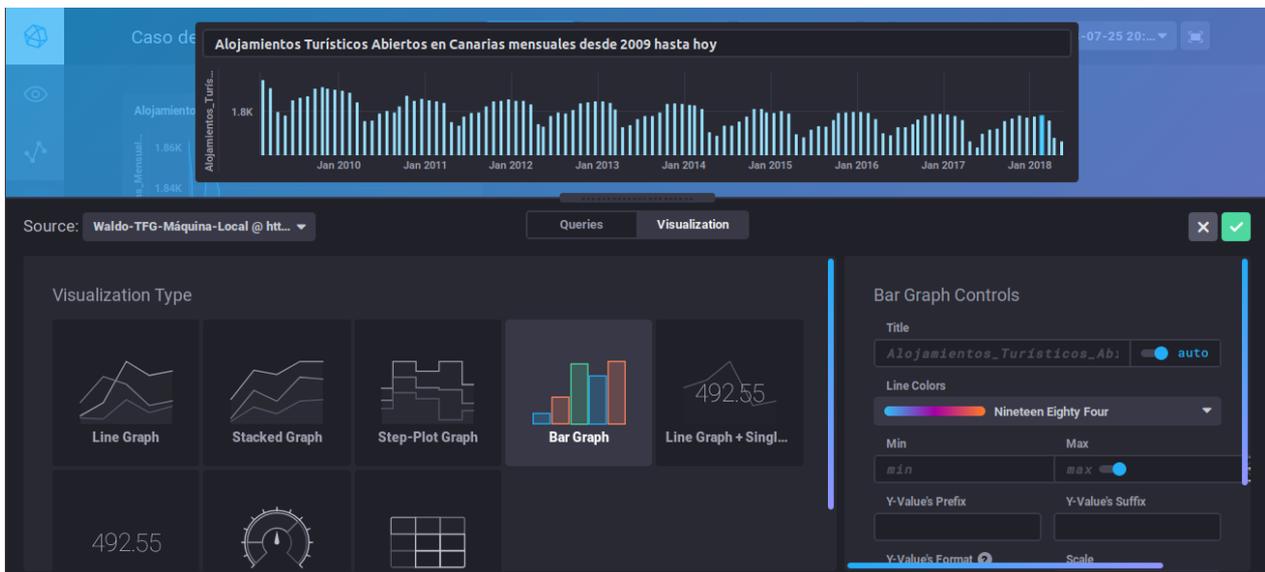


Figura 5.7 Ejemplo de diagrama de barras en Chronograf.

En las capturas anteriores podemos observar un ejemplo de visualización de los datos en Chronograf mediante una gráfica y un diagrama de barras relacionados con los alojamientos turísticos abiertos en Canarias mensuales desde 2009 hasta la fecha.

## 5.3 Grafana

### 5.3.1 Introducción a Grafana

Grafana es una aplicación de código abierto que facilita el desarrollo de cuadros de mando sobre métricas basadas en series de tiempo, tanto para mostrar y analizar datos de infraestructuras (servidores, routers, etc.) como aplicaciones propias. Grafana da soporte a múltiples aplicaciones y bases de datos como las analizadas en el estudio de las bases de datos temporales (véase capítulo 2.1.2). Por defecto trae incorporadas gráficas y datasources, y permite también la instalación mediante plugins de más opciones de visualización y datasources que no se encuentren instaladas por defecto siempre y cuando les ofrezca soporte.

### 5.3.2 Configuración de Grafana

Grafana se instala por defecto utilizando el puerto 3000 de la máquina local (localhost) y accederemos a la aplicación a través de un navegador web [17], introduciendo la dirección <http://localhost:3000>. Para poder acceder a la aplicación instalada en la máquina virtual del IaaS desde una máquina local debemos cambiar el puerto 3000 por 8087 o por otro puerto que permita la configuración de conexión de la máquina virtual ya que la máquina virtual no tiene habilitado el puerto 3000. También para poder establecer conexión debemos estar conectados mediante VPN a la red de la Universidad de La Laguna.

Al acceder se nos solicitará usuario y contraseña de acceso como se muestra en la siguiente captura:

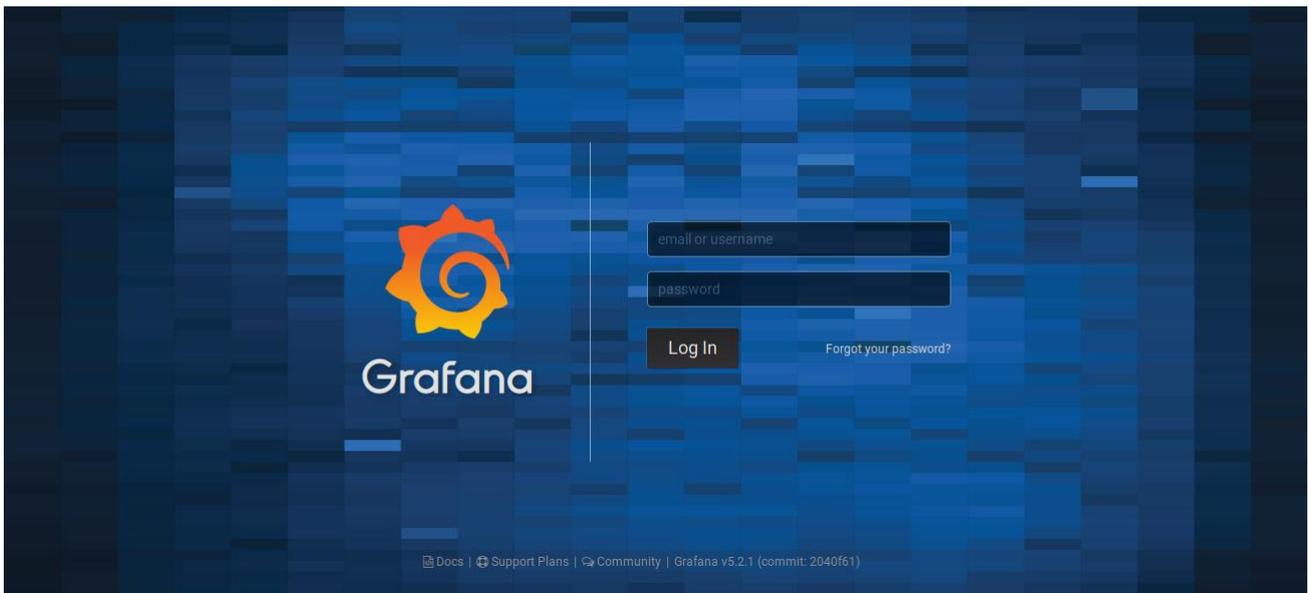


Figura 5.8 Pantalla principal de Grafana.

Una vez hayamos iniciado sesión correctamente en la aplicación debemos configurar las datasources para que Grafana pueda establecer conexión con las bases de datos.

Para configurar una datasource se nos solicitará un nombre de conexión, tipo de base de datos a las que se ofrezca soporte (en nuestro caso InfluxDB) y en detalles de bases de datos debemos indicar el nombre de la base de datos a consultar (en nuestro caso INDICADORES\_(ISTAC)), el nombre de un usuario y contraseña. Si se ha configurado correctamente debería mostrar el contenido de la siguiente captura:

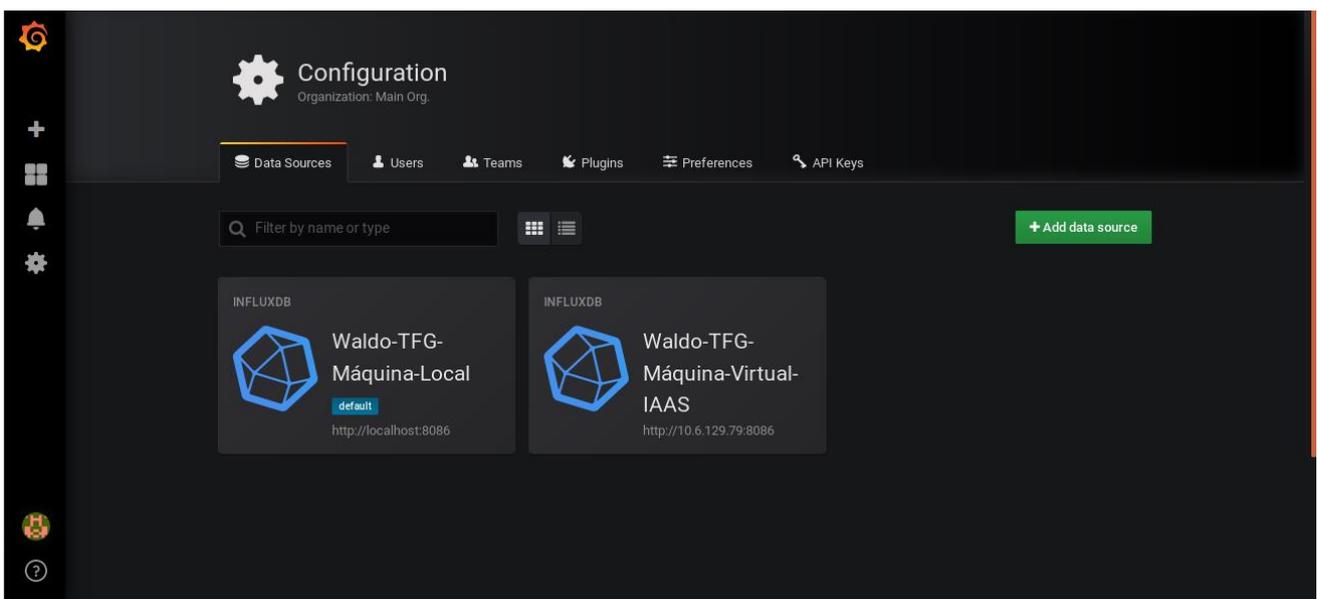


Figura 5.9 Configuración de datasources en Grafana.

### 5.3.3 Caso de ejemplo en Grafana

Para la creación de un dashboard se debe seleccionar el intervalo de tiempo de los datos que deseamos consultar, en la parte superior derecha de la aplicación, por ejemplo 2009 como fecha base de consulta. En la opción dashboard se puede elegir graph para crear una gráfica, para la cual es necesario crear una consulta, como por ejemplo, la configuración que aparece en la siguiente captura:

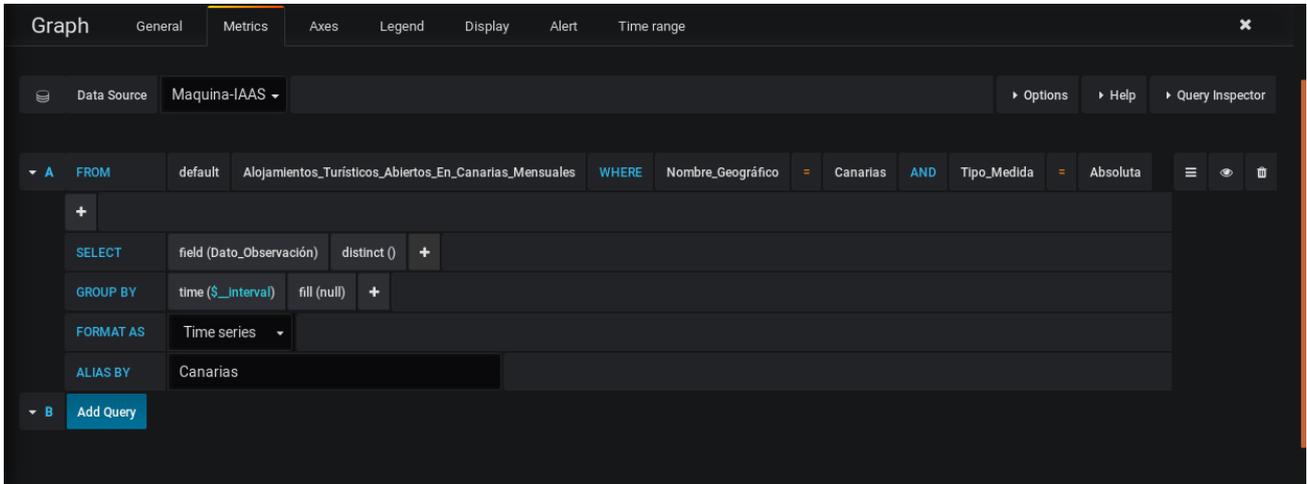


Figura 5.10 Creación de query con Grafana.

En la Figura 5.11 podemos observar la creación de una gráfica comparando todos los indicadores estadísticos correspondientes al área de Turismo en Canarias utilizando como fecha base el año 2009.

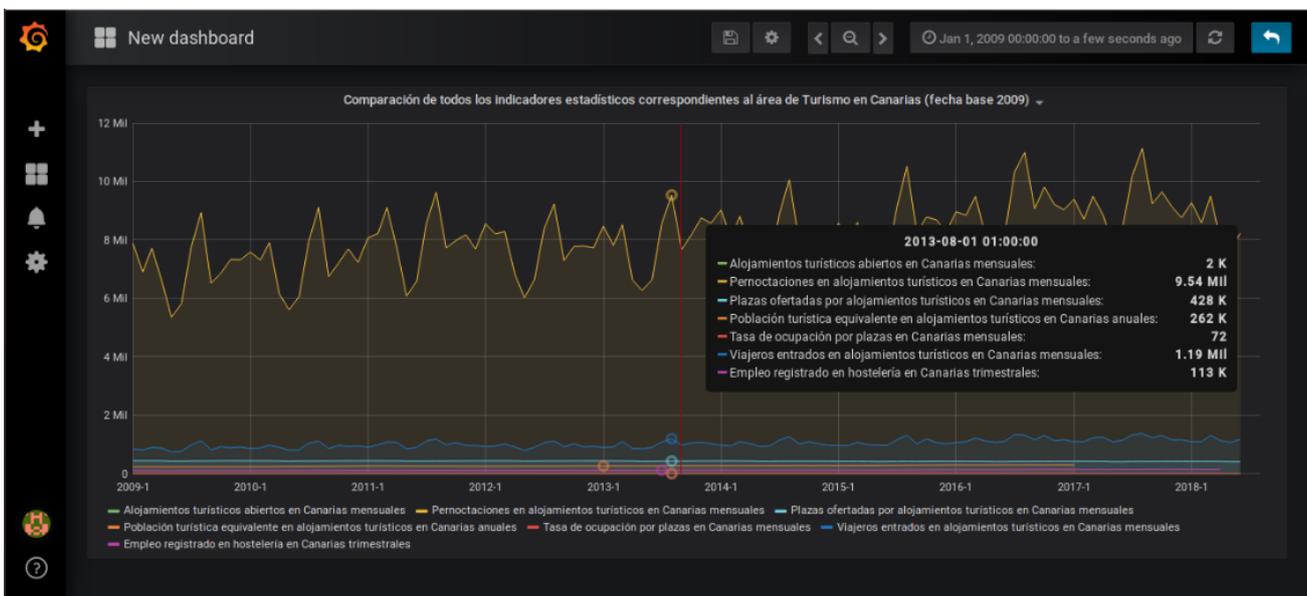


Figura 5.11 Ejemplo de visualización con Grafana.

## 5.4 Chronograf vs Grafana

Como hemos podido observar Chronograf y Grafana pueden mostrar resultados similares pero son aplicaciones diferentes.

El uso de Chronograf está pensado para usuarios administradores de las bases de datos InfluxDB ya que Chronograf desde la propia aplicación permite la gestión y administración de las bases de datos InfluxDB (creación, modificación y eliminación de bases de datos). Chronograf está pensada también como una alternativa para aquellos usuarios que les resulte más cómodo gestionar y administrar las bases de datos desde una interfaz gráfica administrativa en lugar de hacerlo por la terminal de la consola (véase apartado 4.2.2). Teniendo en cuenta el funcionamiento de Chronograf no es recomendable conceder acceso a la aplicación a usuarios que no sean administradores o miembros de la organización ya que si se concediera acceso a terceros las bases de datos podrían quedar expuestas a cambios o modificaciones que comprometan los datos del sistema. Para este proyecto las pruebas en Chronograf se realizan desde la máquina local del alumno como administrador del sistema.

Grafana, a diferencia de Chronograf, da soporte a InfluxDB y a todas las bases de datos que hemos analizado en el estudio sobre bases de datos temporales (véase apartado 2.1.2). Grafana al contrario de Chronograf no permite la gestión y administración de base de datos sino que solamente realiza consultas para su visualización sin riesgo alguno de comprometer los datos, lo que la convierte en la mejor opción si deseamos conceder acceso a los datos a terceros o usuarios ajenos a nuestra organización. Para este proyecto se han establecido cuentas de usuarios para que las tutoras del trabajo y usuarios invitados puedan hacer uso de la aplicación.

Se podrá hacer uso de Grafana desde la máquina local del alumno como administrador del sistema, desde la máquina virtual del IaaS requiriendo estar conectado mediante VPN a la red de la Universidad de La Laguna, y por último todo usuario podrá acceder a Grafana de la máquina local del alumno administrador del sistema a través de una dirección web. Para este último caso se ha solicitado la creación de un DNS dinámico gratuito obtenido mediante el servicio que ofrece el sitio [www.no-ip.com](http://www.no-ip.com) para poder hacer público el acceso a Grafana de la máquina local del administrador del sistema desde cualquier red.

# Capítulo 6

## Desarrollo

### 6.1 Plan de trabajo

En la tabla siguiente podemos observar el plan seguido durante el desarrollo de este proyecto:

Distribución de tareas	
<b>Semana 1-3</b>	Revisión de bibliografía, estado del arte, estudio de bases de datos temporales e indicadores estadísticos.
<b>Semana 4-6</b>	Estudio de base de datos temporal seleccionada (InfluxDB) para el desarrollo del proyecto y selección de lenguaje de programación (Python) para el tratamiento y almacenamiento de los datos en la base de datos.
<b>Semana 7-10</b>	Obtención y almacenamiento de los datos.
<b>Semana 11-13</b>	Estudio de aplicaciones de visualización de datos Chronograf y Grafana.
<b>Semana 14</b>	Visualización de los datos con Chronograf y Grafana.
<b>Semana 15</b>	Pruebas de visualización con usuarios y realización de mejoras en el almacenamiento de los datos en la base de datos.
<b>Semana 16-18</b>	Elaboración de memoria y presentación.

Tabla 6.1 Plan de trabajo del proyecto.

### 6.2 Problemas encontrados

- **Escasez de información de bases de datos temporales:** Uno de los problemas encontrados ha sido la escasez de información bibliográfica y técnica sobre las bases de datos estudiadas y la opción seleccionada para este proyecto (InfluxDB) más allá de la que ofrecen los sitios web oficiales de cada una de ellas, tal vez la escasez de información sea debida a que son tecnologías relativamente nuevas. La búsqueda de información referida a InfluxDB se ha realizado por medio del sitio web oficial, foros y artículos en su mayoría.
- **Múltiples versiones de InfluxDB:** Durante la búsqueda de información de InfluxDB, uno de los problemas encontrados ha sido los reportes a problemas surgidos a usuarios que trabajan en múltiples versiones distintas de InfluxDB con características que incluye o deja de incluir dependiendo de la versión con la que se está trabajando. InfluxDB ofrecía una interfaz gráfica de usuario administrativa (anterior a Chronograf) que solo era usable en versiones antiguas que se llegaron a usar en este proyecto, y que al actualizar dejaban de estar operativas ya que el equipo de InfluxData migró todas las funciones de dicha interfaz a la nueva aplicación Chronograf. El desarrollo de este proyecto se ralentizó algún tiempo hasta averiguar que Chronograf sería la aplicación

que heredaría todas las funciones que ofrecía la antigua interfaz gráfica de usuario administrativa.

- **Escasez de información sobre lenguajes de programación para el tratamiento de datos en InfluxDB:** Otro problema ha sido la escasez de información y ejemplos en la red a la hora de tratar los datos en InfluxDB usando diferentes lenguajes de programación (Python en nuestro caso). La información y ejemplos de uso encontrados son muy básicos.
- **Problemas con el almacenamiento de datos en la máquina virtual del IaaS:** Al ejecutar los scripts de Python para el almacenamiento de los datos recopilados del ISTAC en la base de datos InfluxDB instalada en la máquina virtual del IaaS surgió un problema que hacía que los datos se insertaran de manera incorrecta y desordenada. El problema era causado por la versión de Python3 de la máquina virtual, que hubo que actualizar a Python3.6.

# **Parte III**

## **Conclusiones**

# Capítulo 7

## Conclusiones y líneas futuras

Durante el desarrollo de este proyecto, he podido comprobar la importancia de la visualización de datos y su correcta representación a la hora de analizar datos estadísticos. La precisión a la hora de mostrar los datos es de vital importancia ya que podría marcar la diferencia en caso de que los datos a representar sean la base de una toma de decisiones.

Se ha prestado especial atención a la granularidad de los datos ofrecidos, para que su representación sea lo más cómoda y comprensible para el público que quiera utilizar el sistema.

El desarrollo de este proyecto ha sido una experiencia muy gratificante y enriquecedora, he aprendido nuevas tecnologías como son las bases de datos temporales, concretamente la seleccionada para este proyecto InfluxDB, aplicaciones de visualización de datos de métricas basadas en series de tiempo como son Chronograf y Grafana. También he tenido la oportunidad de aprender un lenguaje de programación nuevo para mí como es Python, despertando un gran interés en mí de cara a mi futuro profesional.

Otra característica a destacar en el desarrollo de este proyecto ha sido la utilización de la API del ISTAC. Los datos de la API se actualizarán continuamente y los scripts desarrollados en Python cada vez que sean ejecutados actualizarán la base de datos InfluxDB.

Finalmente, en cuanto a posibles líneas futuras de trabajo, podría considerarse el uso de distintas bases de datos temporales alternativas a InfluxDB como las que se han estudiado previamente en el desarrollo de este proyecto y desarrollar una aplicación de visualización de datos estadísticos basada Grafana, una aplicación sencilla y accesible pensada para cualquier tipo de usuario.

# Capítulo 8

## Summary and Conclusions

During the development of this project, I have been able to verify the importance of data visualization and its correct representation when analyzing statistical data. The accuracy when displaying the data is of vital importance as it could make a difference in case the data to be represented is the basis of decision making.

Special attention has been paid to the granularity of the data offered, so that its representation is as comfortable and understandable for the public that wants to use the system.

The development of this project has been a very rewarding and enriching experience, I have learned new technologies such as temporary databases, specifically the one selected for this project InfluxDB, data visualization applications for metrics based on time series such as Chronograf and Grafana. I have also had the opportunity to learn a new programming language for me such as Python, awakening a great interest in me in the face of my professional future.

Another feature to highlight in the development of this project has been the use of the ISTAC API. The data of the API will be updated continuously and the scripts developed in Python every time they are executed will update the InfluxDB database.

Finally, regarding possible future lines of work, the use of different temporary databases alternative to InfluxDB such as those that have been studied previously in the development of this project and to develop an application for visualization of statistical data based on Grafana, could be considered, simple and accessible application designed for any type of user.

# Capítulo 9

## Presupuesto

Para la elaboración del presupuesto nos hemos basado en el cronograma de trabajo de este proyecto, y en una estimación de los costes para su desarrollo se ha elaborado el siguiente presupuesto:

- Serán requeridos los servicios de un analista que realizará los trabajos de investigación que supone la obtención de los datos estadísticos de los indicadores requeridos, y la mejor forma de disponer esos datos en la aplicación de visualización de datos, según las preferencias de los usuarios finales de la aplicación.
  - Coste estimado: 14,95€/hora laboral.
- Serán requeridos los servicios de un desarrollador de software que deberá desarrollar programas que recopilen y almacenar en la base de datos temporal los datos de los indicadores estadísticos basándonos en las preferencias de los usuarios finales detalladas en la investigación del analista.
  - Coste estimado: 12,95€/hora laboral.

Basándonos en las líneas del cronograma, las tareas asignadas a cada uno de los usuarios seleccionados anteriormente constarán de las siguientes horas laborales o sesiones:

Estimación temporal del desarrollo del proyecto		
Tipo de actividad	Descripción	Tiempo o sesiones dedicados
Análisis	Revisión de antecedentes, estado del arte, consulta de información de la API ISTAC y visualización de los datos.	8 semanas
Desarrollo de software	Desarrollo de los programas que recopilen y almacenen en la base de datos temporal los datos de la API ISTAC	7 semanas

Tabla 10.1 Estimación temporal de desarrollo del proyecto.

Asumiendo que cada semana cuenta con cinco jornadas laborales (de lunes a viernes) y cada jornada supone ocho horas. Asumiendo los costes del trabajo realizado por los empleados, el presupuesto de este proyecto es el siguiente:

<b>Presupuesto total del proyecto.</b>	
<b>Concepto</b>	<b>Precio</b>
Salario analista (14,95€ x 8 horas x 40 jornadas laborales)	4.784€
Salario desarrollador de software (12,95€ x 8 horas x 35 jornadas laborales)	3.626€
<b>TOTAL</b>	<b>8.410€</b>

Tabla 10.2 Presupuesto total del proyecto.

# Apéndice A

## Almacenamiento de datos de indicadores estadísticos en InfluxDB

### A.1 Ejemplo de script en Python

```
#!/*****  
#*  
#* AlojaturAbiertos.py  
#*  
#*****  
#*  
#* AUTOR: Waldo Nazco Torres  
#*  
#*  
#* FECHA: 10/7/2018  
#*  
#*  
#* DESCRIPCION: "Número de hoteles y apartamentos turísticos abiertos."  
#* NOTAS: "Se entiende por establecimiento abierto aquel que ha prestado servicios de alojamiento de pago algún día del mes  
de referencia."  
#*  
#***/  
  
#!/usr/bin/python3  
  
#Se importa influxdb  
from influxdb import InfluxDBClient  
  
#Se define un json con los datos a insertar  
  
import urllib.request, json  
  
import itertools  
  
import datetime
```

```

#Almacenamos los datos correspondientes a las medidas absolutas relacionadas al tiempo "anual"
with
urllib.request.urlopen("https://www3.gobiernodecanarias.org/istac/api/indicators/v1.0/indicators/ALOJATUR_ABIERTOS/data?representation=MEASURE%5BABSOLUTE%5D&granularity=TIME%5BYEARLY%5D&fields=-observationsMetadata") as url:
    data = json.loads(url.read().decode())
    print(data)

Obs_Absolutas = []
i=0
for d in data["observation"]:
    float_observation = data["observation"][i]
    if float_observation == '.':
        Obs_Absolutas.append(None)
    else:
        float_observation = float(float_observation)
        print(float_observation)
        Obs_Absolutas.append(float_observation)
    i = i + 1

print (i)
print (Obs_Absolutas)

#Almacenamos los datos de todos los metadatos
with
urllib.request.urlopen("https://www3.gobiernodecanarias.org/istac/api/indicators/v1.0/indicators?q=id%20EQ%20%22ALOJATUR_ABIERTOS%22&fields=%2Bmetadata") as url:
    data2 = json.loads(url.read().decode())
    #print(data)

#Almacenamos los metadatos correspondientes al nombre de ubicación y código de ubicación según el código a la isla, región o comunidad.
Codigos_Ubicaciones = []
Ubicaciones = []
j=0
for d in data["dimension"]["GEOGRAPHICAL"]["representation"]["index"]:
    j=0
    for dt in data2["items"][0]["metadata"]["dimension"]["GEOGRAPHICAL"]["representation"]:
        if d == data2["items"][0]["metadata"]["dimension"]["GEOGRAPHICAL"]["representation"][j]["code"]:
            Ubicaciones.append(data2["items"][0]["metadata"]["dimension"]["GEOGRAPHICAL"]["representation"][j]["title"]["es"])
            Codigos_Ubicaciones.append(data2["items"][0]["metadata"]["dimension"]["GEOGRAPHICAL"]["representation"][j]["code"])
        j = j + 1

print (Codigos_Ubicaciones)

```

```
print (Ubicaciones)
```

```
#Almacenamos los metadatos correspondientes al nombre de tiempo y código de tiempo según el tiempo (año o mes) de la isla, región o comunidad.
```

```
Codigos_Tiempos = []
```

```
Tiempos = []
```

```
j=0
```

```
for d in data["dimension"]["TIME"]["representation"]["index"]:
```

```
    j=0
```

```
    for dt in data2["items"][0]["metadata"]["dimension"]["TIME"]["representation"]:
```

```
        if d == data2["items"][0]["metadata"]["dimension"]["TIME"]["representation"][j]["code"]:
```

```
            Tiempos.append(data2["items"][0]["metadata"]["dimension"]["TIME"]["representation"][j]["title"]["es"])
```

```
            Codigos_Tiempos.append(data2["items"][0]["metadata"]["dimension"]["TIME"]["representation"][j]["code"])
```

```
                j = j + 1
```

```
print (Codigos_Tiempos)
```

```
print (Tiempos)
```

```
#Modificamos fechas para años completos añadiendo al año '01-01' refiriéndose al mes y día respectivamente (casos anuales)
```

```
Fechas = []
```

```
Union = []
```

```
cadena = '-01-01'
```

```
j=0
```

```
for d in data["dimension"]["TIME"]["representation"]["index"]:
```

```
    j=0
```

```
    for dt in data2["items"][0]["metadata"]["dimension"]["TIME"]["representation"]:
```

```
        if d == data2["items"][0]["metadata"]["dimension"]["TIME"]["representation"][j]["code"]:
```

```
            Union = data2["items"][0]["metadata"]["dimension"]["TIME"]["representation"][j]["code"] + cadena
```

```
            Fechas.append(Union)
```

```
                j = j + 1
```

```
print (Fechas)
```

```
#Se conecta a la base de datos
```

```
client = InfluxDBClient('localhost', 8086, 'admin', 'admin', 'INDICADORES_(ISTAC)')
```

```
#Se crea la base de datos INDICADORES_(ISTAC)
```

```
client.create_database('INDICADORES_(ISTAC)')
```

```

#Se inserta los metadatos y datos en InfluxDB
i = 0
z = 0
while i < len(Ubicaciones):
    j = 0
    while j < len(Tiempos):
        if Obs_Absolutas[z] == None:
            j = j + 1
            z = z + 1
        else:
            json_data = [
                {
                    "measurement": "Alojamientos_Turísticos_Abiertos_En_Canarias_Anuales",

                    "tags": {
                        "Código_Geográfico": Codigos_Ubicaciones[i],
                        "Nombre_Geográfico": Ubicaciones[i],
                        "Código_Año": Codigos_Tiempos[j],
                        "Nombre_Año": Tiempos[j],
                        "Tipo_Medida": "Absoluta"

                    },

                    "time": Fechas[j],

                    "fields":{
                        "Dato_Observación": Obs_Absolutas[z]

                    }

                }
            ]

            client.write_points(json_data)

            print("dato geográfico n°:({}) = código Geográfico ({} ) y Nombre Geográfico ({})"
                .format(i,Codigos_Ubicaciones[i],Ubicaciones[i]))

            print("dato tiempo anual n°: ({} ) = código año ({} ) y nombre año ({})"
                .format(j,Codigos_Tiempos[j],Tiempos[j]))

            print("dato medida anual n°: ({} ) = Dato de Medida Absoluta ({})" .format(z,Obs_Absolutas[z]))

            j = j + 1
            z = z + 1

        i = i + 1

```

# Bibliografía

- [1] Rankings comparativo de bases de datos temporales (2018). Db-engines: <https://db-engines.com/en/ranking/time+series+dbms>
- [2] Sistema de indicadores estadísticos del ISTAC (2018). Gobierno de Canarias: <https://www3.gobiernodecanarias.org/istac/api/indicators/v1.0/#%21/indicators>
- [3] Trabajo de fin de grado Pulsec. Alumno Beneharo González González. Universidad de La Laguna. Curso 2013/2014: <https://github.com/beneharo/pulsec>
- [4] Trabajo de fin de grado Vistac IMAS-can. Alumno Eliasib Jesús García Martín. Universidad de La Laguna. Curso 2014/2015: <https://riull.ull.es/xmlui/bitstream/handle/915/866/VIStac-IMAS-Can.pdf?sequence=1&isAllowed=y>
- [5] InfluxData. Sitio web oficial (2018). InfluxData: <https://www.influxdata.com/>
- [6] Base de datos temporal InfluxDB. Sitio web oficial (2018). InfluxData: <https://docs.influxdata.com/influxdb/v1.6/>
- [7] Herramienta Chronograf. Sitio web oficial (2018). InfluxData: <https://docs.influxdata.com/chronograf/v1.6/>
- [8] Herramienta Telegraf. Sitio web oficial (2018). InfluxData: <https://docs.influxdata.com/telegraf/v1.7/>
- [9] Herramienta Kapacitor. Sitio web oficial (2018). InfluxData: <https://docs.influxdata.com/kapacitor/v1.5/>
- [10] Herramienta de visualización de datos Grafana. Sitio web oficial (2018). Grafana: <https://grafana.com/>
- [11] Python (Lenguaje de programación utilizado para la obtención, tratamiento y almacenamiento de datos en la base de datos). Sitio web oficial (2018). Python: <https://www.python.org/>
- [12] API BBVA MARKET estudiada y descartada en el desarrollo del proyecto. Sitio web oficial (2018). BBVA: <https://www.bbvaapimarket.com/documentation/bbva/paystats>
- [13] JSON. Sitio web oficial (2018). JSON: <https://www.json.org/>
- [14] Librería de Python (python-influxdb) con soporte para InfluxDB (2018). Github: <https://github.com/influxdata/influxdb-python>
- [15] Time series database. En Wikipedia actualizado el 11 de Julio de 2018. [https://en.wikipedia.org/wiki/Time\\_series\\_database](https://en.wikipedia.org/wiki/Time_series_database)
- [16] Lenguaje de programación Go. Sitio web oficial (2018). Golang: <https://golang.org/>

[17] Grafana. Guía de instalación del sitio web oficial (2018). Grafana:  
<http://docs.grafana.org/installation/debian/>