# Deliverable 7: Data Analysis as a Service (DAaaS) blueprint

Última modificación: 31/08/2023

Fuentes: El proyecto latex editable de este documento se encuentra en overleaf, en: https://www.overleaf.com/project/644b578f41a2ee8dfbed911a

## 1. Introducción

El proyecto **ANIMaLICoS**: *Advanced Networkmetrics: Interpretable Machine Learning for Intelligent Communication Systems* [1], tiene dos componentes principales a diseñar/implementar (ver Deliverable 5): un **laboratorio de redes *SDN* (*Software-Defined Networking*)** y un servicio ***DAaaS*** (*Data Analysis as a Service*). El laboratorio *SDN* es un piloto para la investigación en este tipo de redes, mientras que el *DAaaS* es un demostrador de técnicas multivariantes de procesado y análisis de estos datos. Este documento se centra en el DAaaS. La estructura general del sistema propuesto en el proyecto *ANIMaLICoS* se muestra en la Figura 1. Este documento toma como base el sistema especificado en el Deliverable 5.

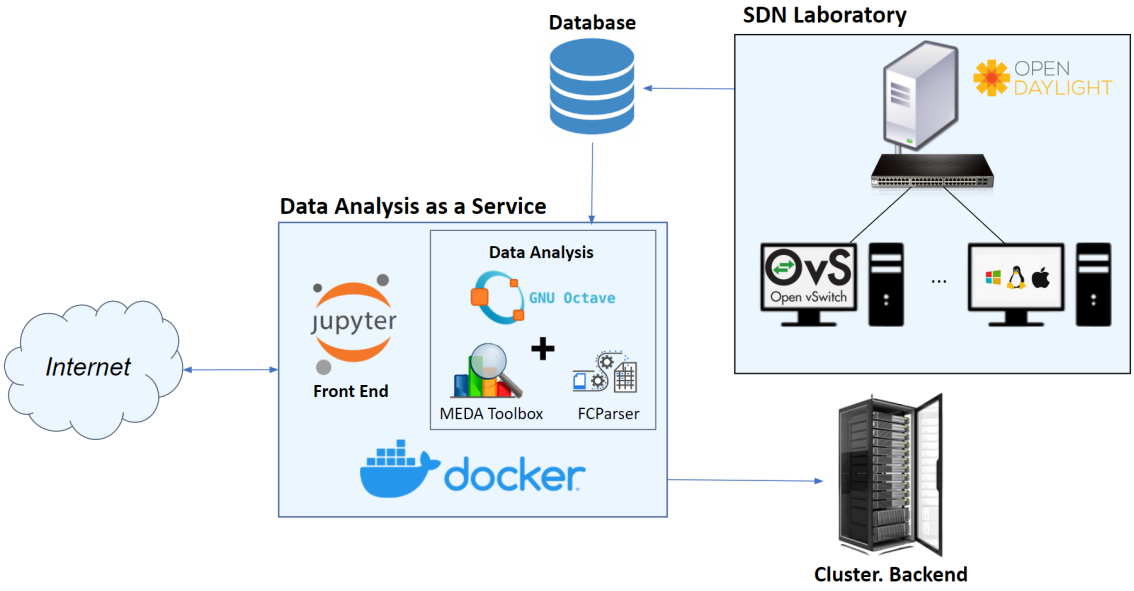


Figura 1: Esquema del sistema: Data Analysis as a Service (*DAaaS*), Laboratorio SDN y su conexión.

## 2. Data Analysis as a Service

El **Data Analysis as a Service** (***DAaaS***) es un servicio online que facilita el análisis e interpretación de datos de cualquier índole, incluyendo: **i**) conjuntos de datos offline proporcionados (como *UGR'16* [2]), **ii**) datos generados en el laboratorio *SDN*, y **iii**) datos que carguen los usuarios en su área personal.

### 2.1. Arquitectura del servicio

El esquema propuesto, donde aparecen las diferentes herramientas y tecnologías a utilizar, se muestra en la Figura 2.

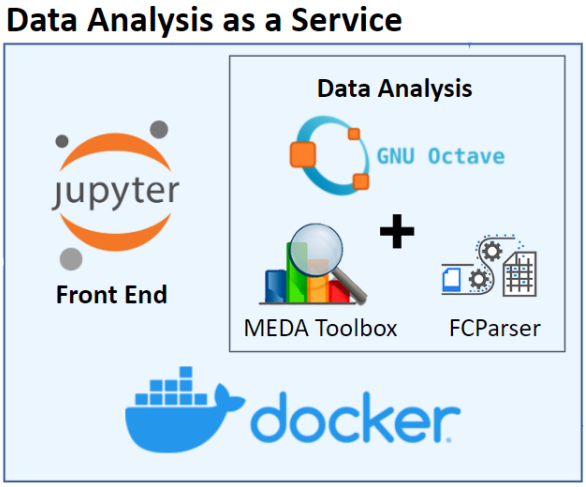


Figura 2: Esquema del Data Analysis as a Service (*DAaaS*).

El *DAaaS* utiliza distintas tecnologías, incluyendo *Docker* como gestor de contenedores y *Notebook de Jupyter* como entorno de *scripting* y programación, en el que se instalarán las herramientas *MEDA Toolbox* y *FCParser*. El objetivo de estas tecnologías es doble:

- **Preprocesado de datos**, por medio de programación en *Python* con la herramienta *FCParser* (`https://github.com/josecamachop/FCParser`).
- **Análisis de datos**, teniendo integradas las herramientas de *Octave* y la *MEDA Toolbox* (`https://github.com/josecamachop/MEDAToolbox`).

En primer lugar, se ha instalado en el servidor un gestor de contenedores, ***Docker*** [3]. Se trata de una plataforma de software abierta diseñada para crear y ejecutar aplicaciones de un modo ágil y versátil. Esto permite que cada usuario ejecute un contenedor con la misma imagen de

base, sin interferir en el desarrollo de los demás, además de permitir la personalización de un entorno para cada usuario. Esto es idóneo para el objeto del DAaaS.

Además, se necesita tener un entorno de análisis de datos para utilizar las herramientas nombradas anteriormente. Para esto se utiliza ***Jupyter Hub*** [4], una aplicación web para el análisis interactivo de datos. Recientemente se ha añadido a ***Jupyter Hub*** la compatibilidad con *Octave*, lo que permite el acceso a comandos y visualizaciones en *MEDA Toolbox*.

La diferencia fundamental de ***Jupyter Hub*** con respecto a su precedesor *Jupyter Notebook* es que incluye una gestión y autenticación de usuarios por parte de un administrador. Esto permite una gestión centralizada para dar acceso restringido y controlado al servidor, lo cual es necesario en este proyecto. Además, permite la utilización de un *notebook* para cada usuario a la misma vez o la actualización de los cambios en tiempo real para el caso de utilizar el mismo usuario.

***FCParser*** [5] es una biblioteca que permite un análisis sintáctico cómodo, general y altamente configurable de los datos procedentes de diferentes fuentes. Todo ello, a partir de una previa decisión por parte del analista para conocer qué fuentes de datos incluye, qué información es relevante, qué criterios utiliza para la agregación y las características de salida. En otras palabras, es una forma de extraer características de un conjunto de datos para poder procesaros e interpretarlos de una manera más simplificada.

Una vez preprocesados los datos, estos se podrán analizar por medio de ***Multivariate Exploratory Data Analysis*** (***MEDA Toolbox***) [6] que es un conjunto de herramientas de análisis multivariante para la exploración de conjuntos de datos. Así, *MEDA Toolbox* incluye los gráficos exploratorios tradicionales basados en el Análisis de Componentes Principales (***PCA***, del inglés, *Principal Component Analysis*) o en los Mínimos Cuadrados Parciales (***PLS***, del inglés, *Partial Least Squares*), como son los gráficos de *scores*, *loadings* y *residuals* y otras visualizaciones como *MEDA, oMEDA, SVI plots, ADICOV, EKF & CKF cross-validation, CSP, GPCA*, entre otros. *MEDA Toolbox* se puede utilizar tanto en *Matlab* [7] como en *Octave* [8]. En este proyecto se preveé utilizar *Octave*, por ser software libre.

## 2.2. Funcionalidades del servidor

En esta sección, se pretende dar otra perspectiva más visual de lo que se ha explicado en la sección anterior, incluyendo una serie de capturas que muestra la estructura del servicio y la manera que se pretende que funcione.

En primer lugar, se encuentra el gestor de contenedores *Docker*, donde mediante el comando que se muestra en la Figura 3, se pueden observar todos los contenedores instalados y que se pueden utilizar independientemente.

Como se puede observar en la Figura 3, se pueden instalar contenedores de todo tipo, desde


MINISTERIO DE CIENCIA E INNOVACIÓN
AGENCIA ESTATAL DE INVESTIGACIÓN

```
$ docker image ls
REPOSITORY                TAG              IMAGE ID       CREATED         SIZE
jupyterhub_octave         latest           b5386fe8baac   10 days ago     702MB
jupyterhub/jupyterhub     latest           91497f74f9e4   3 weeks ago     361MB
ubuntu                    latest           d73974364e66   4 weeks ago     675MB
jupyter/scipy-notebook    latest           cfbd0d404880   4 weeks ago     3.01GB
mtmiller/octavemeda       latest           69c4e4888002   2 months ago    2.53GB
gnuoctave/octave          6.4.0            54464f5ab6ed   3 months ago    3.52GB
nvcr.io/nvidia/pytorch    20.12-py3-IPy    a5125789318b   5 months ago    14.2GB
r-bigmemory               latest           34eb1acc825d   12 months ago   924MB
```

Figura 3: Contenedores disponibles en el *Docker*.

máquinas virtuales de *Ubuntu* hasta *Octave*. En este caso, para el proyecto se va a hacer uso de la que se ha creado con *JupyterHub* y *Octave* con sus herramientas integradas. Haciendo uso del comando *run* se puede ejecutar cada contenedor e, indicando algunos *flags*, se pueden añadir algunas funcionalidades, como la de utilizar los directorios y ficheros que se pretenden analizar.

Una vez ejecutado, se obtendría una salida como la que se muestra en la Figura 4 y para visualizar el área de trabajo de *Jupyter* solo habría que abrir un navegador y acceder a la dirección y puerto donde hemos lanzado el servicio.

```
root@b797d643ebdf:/home# sh launch.sh
[I 2022-06-03 11:31:28.736 JupyterHub app:2769] Running JupyterHub version 2.3.0
[I 2022-06-03 11:31:28.737 JupyterHub app:2799] Using Authenticator: nativeauthenticator.nativeauthenticator.NativeAuthenticator
[I 2022-06-03 11:31:28.737 JupyterHub app:2799] Using Spawner: jupyterhub.spawner.LocalProcessSpawner-2.3.0
[I 2022-06-03 11:31:28.737 JupyterHub app:2799] Using Proxy: jupyterhub.proxy.ConfigurableHTTPProxy-2.3.0
[I 2022-06-03 11:31:28.799 JupyterHub app:1606] Loading cookie_secret from /home/jupyterhub_cookie_secret
[I 2022-06-03 11:31:28.909 JupyterHub proxy:496] Generating new CONFIGPROXY_AUTH_TOKEN
[I 2022-06-03 11:31:28.930 JupyterHub app:1924] Not using allowed_users. Any authenticated user will be allowed.
[I 2022-06-03 11:31:28.970 JupyterHub app:2838] Initialized 0 spawners in 0.005 seconds
[W 2022-06-03 11:31:28.974 JupyterHub proxy:687] Running JupyterHub without SSL.  I hope there is SSL termination happening somewhere else...
[I 2022-06-03 11:31:28.975 JupyterHub proxy:691] Starting proxy @ http://:8000
```

Figura 4: Ejecución de *JupyterHub*.

La pantalla que se encontraría el usuario al acceder a la dirección sería el inicio de sesión, tal y como se muestra en la Figura 5. En ella, tendría la opción de crear un nuevo usuario o de acceder si ya tuviese las credenciales y el acceso verificado por el administrador (sólo puede dar acceso a nuevos usuarios).

Por último, una vez autenticado, accedería a la interfaz que se muestra en la Figura 6, donde se podrían ejecutar archivos de *python* para el procesado de datos (como el caso de *FCParser*), o utilizar *Octave* para el uso de *MEDA Toolbox* para el análisis de datos.

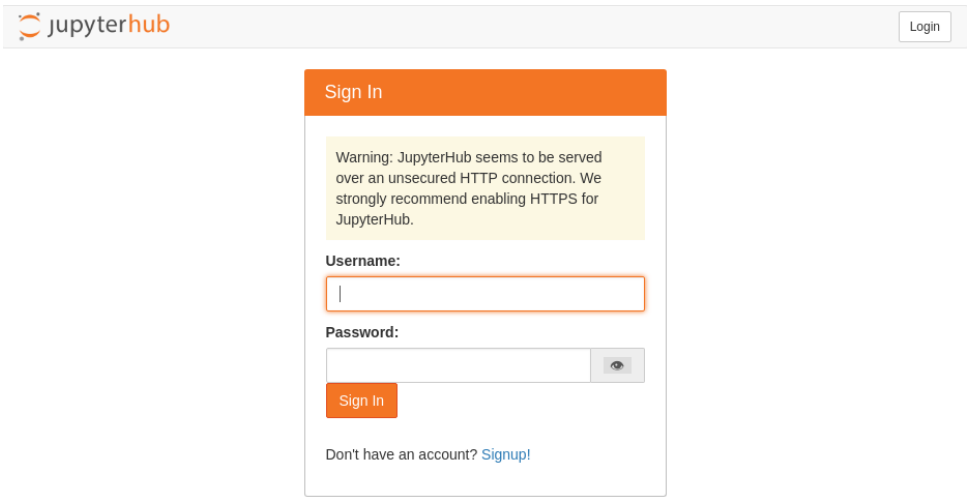


Figura 5: Inicio de sesión en *JupyterHub*.

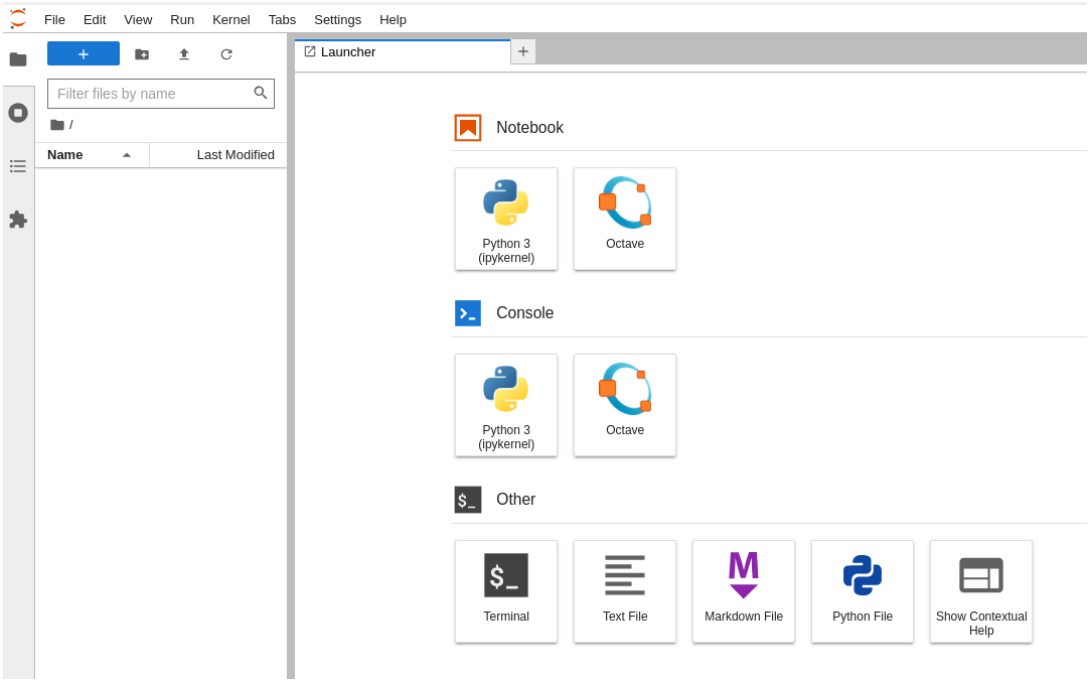


Figura 6: Pantalla de inicio de *JupyterHub*.

## Referencias

[1] ANIMaLICoS. Advanced networkmetrics: Interpretable machine learning for intelligent communication systems. https://www.codas.ugr.es/animalicos/en.

[2] Gabriel Maciá Fernández, José Camacho, Roberto Magán-Carrión, Pedro García-Teodoro, Roberto Theron, Ugr'16: a new dataset for the evaluation of cyclostationarity-based network IDSs, In Computers Security, 2017.

[3] Docker. Available online: https://www.docker.com/.

[4] Project jupyter, jupyterhub. Available online: https://jupyter.org/.

[5] Feature as a counter parser for networkmetrics. Available online: https://github.com/josecamachop/FCParser.

[6] Multivariate exploratory data analysis (meda) toolbox. Available online: https://github.com/josecamachop/MEDA-Toolbox.

[7] Matlab. Available online: https://es.mathworks.com/products/matlab.html.

[8] Octave. Available online: https://www.gnu.org/software/octave/index.