Trabajo Fin de Grado
Ingeniería de Telecomunicación

Desarrollo de un entorno de virtualización y pruebas de RPL (Routing Protocol for Low-Power and Lossy Networks) con Contiki y Vagrant

Autor: Javier Esteban Expósito

Tutor: Juan Antonio Ternero Muñiz

Dpto. Ingeniería Telemática
Escuela Técnica Superior de Ingeniería
Universidad de Sevilla

Sevilla, 2023

Departamento de Teoría de la Señal y Comunicaciones

Trabajo Fin de Grado
Ingeniería de Telecomunicación

Desarrollo de un entorno de virtualización y pruebas de RPL (Routing Protocol for Low-Power and Lossy Networks) con Contiki y Vagrant

Autor:

Javier Esteban Expósito

Tutor:

Juan Antonio Ternero Muñiz

Profesor Titular

Dpto. Ingeniería Telemática
Escuela Técnica Superior de Ingeniería
Universidad de Sevilla

Sevilla, 2023

Trabajo Fin de Grado: Desarrollo de un entorno de virtualización y pruebas de RPL (Routing Protocol for Low-Power and Lossy Networks) con Contiki y Vagrant

Autor: Javier Esteban Expósito
Tutor: Juan Antonio Ternero Muñiz

El tribunal nombrado para juzgar el trabajo arriba indicado, compuesto por los siguientes profesores:

Presidente:

Vocal/es:

Secretario:

acuerdan otorgarle la calificación de:

El Secretario del Tribunal

Fecha:

Agradecimientos

A mi madre.

Javier Esteban Expósito
Sevilla, 2023

Resumen

Contiki-ng es un sistema operativo de código abierto altamente eficiente diseñado para ser utilizado en motas, dispositivos con recursos limitados que funcionan con baja potencia, para desarrollar aplicaciones de forma sencilla y eficiente que resuelvan distintos casos de uso que pueden presentarse en escenarios IoT. Estos casos de uso incluyen la monitorización de sensores, el control de dispositivos y la comunicación en red.

Para el desarrollo productivo de estas aplicaciones, es fundamental verificar que los resultados obtenidos de los diferentes desarrollos no dependen de la forma en que se haya instalado Contiki-ng. Por lo tanto, el objetivo de este trabajo de fin de grado es realizar una comparativa de las métricas obtenidas al realizar simulaciones de tres escenarios distintos en las tres versiones oficiales en las que se soporta el despliegue de Contiki-ng. Estas versiones son la imagen de Docker, la máquina virtual de Vagrant y la instalación nativa.

Abstract

Contiki-ng is a highly efficient open-source operating system designed to be used on motes, devices with limited resources that operate on low power, to develop applications easily and efficiently that solve different use cases that may arise in IoT scenarios. These use cases include sensor monitoring, device control, and network communication.
For the productive development of these applications, it is essential to verify that the results obtained from different developments do not depend on the way Contiki-ng has been installed. Therefore, the aim of this project is to perform a comparison of the metrics obtained by conducting simulations of three different scenarios in the three official versions in which Contiki-ng deployment is supported. These versions are the Docker image, the Vagrant virtual machine, and the native installation.

Índice Abreviado

Índice

1 Introducción

Este trabajo de fin de grado es una expansión del trabajo [1] . En [1] el objetivo consistía en comparar los resultados de simular tres escenarios utilizando el simulador Cooja, que viene integrado por defecto en Contiki-ng, ejecutando las simulaciones desde la versión de Contiki-ng que se puede instalar de forma nativa y desde la imagen de Docker. En este trabajo de fin de grado, el objetivo será realizar una comparativa similar, pero incluyendo también los resultados de ejecutar las simulaciones desde el entorno desplegado en una máquina virtual con Vagrant.

Adicionalmente, para la comparación de los resultados se ha optado por utilizar una versión autogestionada de Elasticsearch, que, junto con Kibana, permitirá la ingesta de grandes cantidades de datos. Se ha optado por utilizar Elasticsearch para este propósito en lugar de otras soluciones por ser una herramienta puntera, ampliamente utilizada en entornos empresariales productivos.

1.1 Motivación

El Internet de las cosas -IoT- está transformando a pasos agigantados un gran número de industrias: Desde la sanidad hasta la manufactura, pasando por la agricultura. La necesidad de acceder a cantidades masivas de datos obtenibles en tiempo real para optimizar procesos empresariales y tomas de decisiones ha hecho que el coste de la tecnología utilizada en la fabricación de motas haya decrecido, lo que a su vez ha permitido democratizar aún más esta tecnología. Hoy en día los negocios tienen la posibilidad de investigar esta tecnología para explorar posibles implementaciones que les ayuden a incrementar su productividad.

En este contexto se hace necesaria la utilización de un entorno en el que se puedan desarrollar estos proyectos IoT de manera fiable. Existen varios posibles entornos utilizados a nivel empresarial para este propósito, de los cuales hablaremos en el capítulo 2, pero para las pruebas realizadas en este trabajo se ha utilizado Contiki-ng. Contiki-ng provee de tres formas de instalación: nativa -también llamada toolchain-, mediante imagen de Docker y mediante máquina virtual en Vagrant. ¿Debería un proyecto verse afectado por la forma en que se ha desplegado el entorno de desarrollo? La respuesta es no, y este trabajo va a enfocarse en realizar una comparativa de los datos obtenidos mediante los tres métodos de despliegue para demostrarlo.

1.2 Objetivos

Como ya se ha mencionado en el anterior apartado, el objetivo de este trabajo es confirmar que los resultados de las simulaciones de tres escenarios no varían dependiendo de la forma en que hemos instalado Contiki-ng. Para la realización de este trabajo no fue posible conseguir acceso a los escenarios utilizados en [1], por lo que fue necesario que crear escenarios completamente diferentes.

Para ello, se han creado tres escenarios nuevos, que se han ejecutado durante tres horas. Una vez finalizadas las simulaciones, se ha hecho uso de un script que convierte las trazas generadas en un archivo .csv que Elasticsearch puede ingerir directamente. Con Elasticsearch y Kibana se han creado gráficas que comparan distintas métricas -media, desviación típica y mediana- para los tres escenarios, ejecutados en los tres entornos. Con estas gráficas se puede obtener una comparativa visual de los resultados, haciendo evidente que no dependen del método de despliegue del entorno.

1.3 Estructura de la memoria

La memoria se dive en las siguientes partes:

- Introducción: Es el capítulo en el que nos encontramos, aquí se presenta la motivación original de este trabajo de fin de grado, el punto del que se ha partido, las decisiones que se han tomado de cara a la valoración de los resultados y una descripción del resto del contenido de la memoria.
- Estudio de las tecnologías: Debido a que este trabajo es una continuación de otro anterior, la selección del entorno para el desarrollo de los escenarios viene heredada de [1]. Sin embargo, en esta sección se elaborarán las razones por las que se ha decidido utilizar Elasticsearch con Kibana y un script para transformar las trazas que genera el módulo Energest, que es el utilizado para la verificación de los resultados, en lugar de otras alternativas.
- Despliegues: Se explican aquí los detalles que se han tenido en cuenta para el despliegue de Contiki-ng y del entorno de visualizacón de resultados, además de decisiones particulares que fue necesario tomar y no aparecían directamente en las guías de Contiki-ng, Docker o Vagrant.
- Fase de pruebas: En esta sección, se detallan los escenarios utilizados para comparar los resultados en los tres tipos de despliegue distintos. Se describen las pruebas realizadas y los resultados obtenidos. Se especifica información detallada sobre los criterios de evaluación utilizados y cómo se han medido los resultados.
- Conclusiones: En esta sección, se resumen los hallazgos del trabajo y se ofrecen posibles continuaciones al mismo. Se incluye una evaluación crítica del trabajo realizado y una reflexión sobre las limitaciones y posibles mejoras futuras. Además, se destaca la contribución del trabajo al campo de estudio y su relevancia para futuras investigaciones.

2 Estudio de las tecnologías

La implementación de proyectos de IoT (Internet de las cosas) requiere el uso de motas, que son dispositivos electrónicos diseñados específicamente para ser muy eficientes a la hora de realizar labores de medición y transmisión de datos en tiempo real. Para disminuir costes en estos proyectos, es necesario poder realizar simulaciones previas al despliegue de la aplicación desarrollada en las motas. En este episodio, se dará un breve repaso a las características de Contiki-ng, la razón por la que se ha utilizado el simulador Cooja para la realización de las pruebas y la justificación de la elección de las tecnologías utilizadas para la visualización de los resultados.

2.1 Contiki-ng

Contiki-ng es un sistema operativo de código abierto para dispositivos IoT (Internet de las cosas). Está diseñado específicamente para ser utilizado en motas y ofrece una plataforma de desarrollo para construir aplicaciones de IoT en estos dispositivos de baja potencia.

Contiki-ng ofrece una amplia variedad de características, como la gestión de energía, la comunicación inalámbrica, el enrutamiento y la seguridad, lo que lo hace adecuado para aplicaciones de IoT que requieran la conexión de múltiples dispositivos. Además, este sistema operativo es altamente modular y configurable, lo que significa que los desarrolladores pueden seleccionar los componentes que se ajusten a sus necesidades y limitaciones de hardware.

Otra de las características destacadas de Contiki-ng es su capacidad para ejecutar múltiples procesos en paralelo, lo que permite una mayor eficiencia en la gestión de recursos. Además, cuenta con una amplia comunidad de desarrolladores y una documentación extensa, lo que facilita su uso y su integración en diferentes proyectos.

2.1.1 Estructura de carpetas de Contiki-ng

Si comparamos la estructura de directorios de Contiki-ng con aquella que podría tener otra distribución basada en Linux, como Ubuntu, nos daremos cuenta de que difieren debido a la especificidad que se buscaba al crear Contiki-ng. Contiki-ng sigue una arquitectura modular, donde cada módulo se encuentra en su carpeta correspondiente, y dentro de dicha carpeta nos encontramos con el código fuente del módulo y los ficheros Makefile que se utilizan para su compilación.

Código 2.1 Listado de los contenidos del directorio principal de Contiki-ng.

```
javier@TFG:~/Contiki-ng$ ll -h ./*
-rw-rw-r-- 1 javier javier 3,3K feb 13 23:40 ./CODE_OF_CONDUCT.md
-rw-rw-r-- 1 javier javier 143 feb 13 23:40 ./CONTRIBUTING.md
-rw-rw-r-- 1 javier javier 957 feb 13 23:40 ./lgtm.yml
-rw-rw-r-- 1 javier javier 1,5K feb 13 23:40 ./LICENSE.md
-rw-rw-r-- 1 javier javier 3,6K feb 13 23:40 ./Makefile.dir-variables
-rw-rw-r-- 1 javier javier 1,1K feb 13 23:40 ./Makefile.embedded
```

```
-rw-rw-r-- 1 javier javier 1,8K feb 13 23:40 ./Makefile.help
-rw-rw-r-- 1 javier javier 922 feb 13 23:40 ./Makefile.identify-target
-rw-rw-r-- 1 javier javier 21K feb 13 23:40 ./Makefile.include
-rw-rw-r-- 1 javier javier 687 feb 13 23:40 ./Makefile.tools
-rw-rw-r-- 1 javier javier 2,8K feb 13 23:40 ./README.md
-rw-rw-r-- 1 javier javier 861 feb 13 23:40 ./SECURITY.md

./arch:
total 20K
drwxrwxr-x 5 javier javier 4,0K feb 13 23:40 ./
drwxrwxr-x 10 javier javier 4,0K feb 13 23:40 ../
drwxrwxr-x 11 javier javier 4,0K feb 13 23:40 cpu/
drwxrwxr-x 7 javier javier 4,0K feb 13 23:40 dev/
drwxrwxr-x 14 javier javier 4,0K feb 13 23:40 platform/

./doc:
total 40K
drwxrwxr-x 7 javier javier 4,0K feb 13 23:40 ./
drwxrwxr-x 10 javier javier 4,0K feb 13 23:40 ../
drwxrwxr-x 2 javier javier 4,0K feb 13 23:40 getting-started/
-rw-rw-r-- 1 javier javier 7,0K feb 13 23:40 Home.md
-rw-rw-r-- 1 javier javier 282 feb 13 23:40 index.rst
drwxrwxr-x 3 javier javier 4,0K feb 13 23:40 platforms/
drwxrwxr-x 3 javier javier 4,0K feb 13 23:40 programming/
drwxrwxr-x 2 javier javier 4,0K feb 13 23:40 project/
drwxrwxr-x 3 javier javier 4,0K feb 13 23:40 tutorials/

./examples:
total 88K
drwxrwxr-x 22 javier javier 4,0K feb 13 23:40 ./
drwxrwxr-x 10 javier javier 4,0K feb 13 23:40 ../
drwxrwxr-x 10 javier javier 4,0K feb 13 23:40 6tisch/
drwxrwxr-x 5 javier javier 4,0K feb 13 23:40 benchmarks/
drwxrwxr-x 5 javier javier 4,0K feb 13 23:40 coap/
drwxrwxr-x 2 javier javier 4,0K feb 13 23:40 cplusplus/
drwxrwxr-x 7 javier javier 4,0K feb 13 23:40 dev/
drwxrwxr-x 3 javier javier 4,0K mar 12 11:28 hello-world/
drwxrwxr-x 2 javier javier 4,0K feb 13 23:40 ip64-router/
drwxrwxr-x 14 javier javier 4,0K feb 13 23:40 libs/
drwxrwxr-x 3 javier javier 4,0K feb 13 23:40 lwm2m-ipso-objects/
drwxrwxr-x 3 javier javier 4,0K feb 13 23:40 mqtt-client/
drwxrwxr-x 4 javier javier 4,0K feb 13 23:40 multicast/
drwxrwxr-x 2 javier javier 4,0K feb 13 23:40 nullnet/
drwxrwxr-x 6 javier javier 4,0K feb 13 23:40 platform-specific/
drwxrwxr-x 4 javier javier 4,0K feb 13 23:40 rpl-border-router/
drwxrwxr-x 3 javier javier 4,0K feb 14 00:10 rpl-udp/
drwxrwxr-x 10 javier javier 4,0K feb 13 23:40 sensniff/
drwxrwxr-x 3 javier javier 4,0K feb 13 23:40 slip-radio/
drwxrwxr-x 3 javier javier 4,0K feb 13 23:40 snmp-server/
drwxrwxr-x 5 javier javier 4,0K feb 13 23:40 storage/
drwxrwxr-x 3 javier javier 4,0K feb 13 23:40 websocket/

./os:
total 64K
drwxrwxr-x 8 javier javier 4,0K feb 13 23:40 ./
drwxrwxr-x 10 javier javier 4,0K feb 13 23:40 ../
-rw-rw-r-- 1 javier javier 9,2K feb 13 23:40 contiki-default-conf.h
```

```
-rw-rw-r-- 1 javier javier 2,0K feb 13 23:40 contiki.h
-rw-rw-r-- 1 javier javier 1,9K feb 13 23:40 contiki-lib.h
-rw-rw-r-- 1 javier javier 5,6K feb 13 23:40 contiki-main.c
-rw-rw-r-- 1 javier javier 2,2K feb 13 23:40 contiki-net.h
drwxrwxr-x 2 javier javier 4,0K feb 13 23:40 dev/
drwxrwxr-x 7 javier javier 4,0K feb 13 23:40 lib/
drwxrwxr-x 8 javier javier 4,0K feb 13 23:40 net/
drwxrwxr-x 15 javier javier 4,0K feb 13 23:40 services/
drwxrwxr-x 4 javier javier 4,0K feb 13 23:40 storage/
drwxrwxr-x 2 javier javier 4,0K feb 13 23:40 sys/

./tests:
total 104K
drwxrwxr-x 20 javier javier 4,0K feb 13 23:40 ./
drwxrwxr-x 10 javier javier 4,0K feb 13 23:40 ../
drwxrwxr-x 2 javier javier 4,0K feb 13 23:40 00-documentation/
drwxrwxr-x 2 javier javier 4,0K feb 13 23:40 01-compile-base/
drwxrwxr-x 2 javier javier 4,0K feb 13 23:40 02-compile-arm-ports/
drwxrwxr-x 2 javier javier 4,0K feb 13 23:40 05-compile-tools/
drwxrwxr-x 5 javier javier 4,0K feb 13 23:40 06-script-base/
drwxrwxr-x 6 javier javier 4,0K feb 13 23:40 07-simulation-base/
drwxrwxr-x 5 javier javier 4,0K feb 13 23:40 08-native-runs/
drwxrwxr-x 4 javier javier 4,0K feb 13 23:40 09-ipv6/
drwxrwxr-x 3 javier javier 4,0K feb 13 23:40 10-ipv6-nbr/
drwxrwxr-x 6 javier javier 4,0K feb 13 23:40 13-ieee802154/
drwxrwxr-x 4 javier javier 4,0K feb 13 23:40 14-rpl-lite/
drwxrwxr-x 4 javier javier 4,0K feb 13 23:40 15-rpl-classic/
drwxrwxr-x 2 javier javier 4,0K feb 13 23:40 17-tun-rpl-br/
drwxrwxr-x 4 javier javier 4,0K feb 13 23:40 18-coap-lwm2m/
drwxrwxr-x 2 javier javier 4,0K feb 13 23:40 19-out-of-tree-build/
drwxrwxr-x 3 javier javier 4,0K feb 13 23:40 20-packet-parsing/
-rwxrwxr-x 1 javier javier 229 feb 13 23:40 check-test.sh*
drwxrwxr-x 2 javier javier 4,0K feb 13 23:40 compile-all/
-rw-rw-r-- 1 javier javier 1,8K feb 13 23:40 Makefile
-rw-rw-r-- 1 javier javier 4,0K feb 13 23:40 Makefile.compile-test
-rw-rw-r-- 1 javier javier 524 feb 13 23:40 Makefile.script-test
-rw-rw-r-- 1 javier javier 3,2K feb 13 23:40 Makefile.simulation-test
drwxrwxr-x 2 javier javier 4,0K feb 13 23:40 scan_build/
-rwxrwxr-x 1 javier javier 3,2K feb 13 23:40 utils.sh*

./tools:
total 76K
drwxrwxr-x 19 javier javier 4,0K feb 13 23:40 ./
drwxrwxr-x 10 javier javier 4,0K feb 13 23:40 ../
drwxrwxr-x 3 javier javier 4,0K feb 13 23:49 cc2538-bsl/
drwxrwxr-x 2 javier javier 4,0K feb 13 23:40 code-style/
drwxrwxr-x 3 javier javier 4,0K feb 13 23:40 coffee-manager/
drwxrwxr-x 15 javier javier 4,0K feb 25 12:01 cooja/
drwxrwxr-x 3 javier javier 4,0K feb 13 23:40 docker/
drwxrwxr-x 2 javier javier 4,0K feb 13 23:40 doxygen/
drwxrwxr-x 2 javier javier 4,0K feb 13 23:40 git/
drwxrwxr-x 2 javier javier 4,0K feb 13 23:40 ip64/
drwxrwxr-x 3 javier javier 4,0K feb 13 23:40 jn516x/
drwxrwxr-x 3 javier javier 4,0K feb 13 23:49 motelist/
drwxrwxr-x 3 javier javier 4,0K feb 13 23:40 readthedocs/
drwxrwxr-x 2 javier javier 4,0K feb 13 23:49 sensniff/
drwxrwxr-x 2 javier javier 4,0K feb 13 23:40 serial-io/
```

```
drwxrwxr-x 2 javier javier 4,0K feb 13 23:40 sky/
drwxrwxr-x 3 javier javier 4,0K feb 23 21:13 vagrant/
drwxrwxr-x 2 javier javier 4,0K feb 13 23:40 viewconf/
drwxrwxr-x 2 javier javier 4,0K feb 13 23:40 zolertia/
```

Los contenidos de las carpetas listadas son los siguientes:

- contiki-ng: Es la carpeta principal, contiene al resto de subcarpetas, los ficheros Makefile que se utilizan para compilar Contiki-ng y ejecutar sus aplicaciones y los ficheros README, utilizados para dirigirnos a enlaces de interés, como la documentación.
- arch: Este directorio contiene tres subdirectorios con código específico para cada arquitectura que soporta Contiki-ng. Los subdirectorios son:
  - arch/cpu: Contiene el código fuente que implementa las instrucciones cada arquitectura de CPU en la que se puede desarrollar con Contiki-ng.
  - arch/dev: Contiene subcarpetas donde cada una es un módulo que se puede utilizar en desarrollos para interactuar con los periféricos de las motas, como LEDs, dispositivos de entrada y salida de radio y Ethernet...
  - arch/platform: Contiene las librerías que seran utilizadas cuando se desarrolle para una plataforma específica.
- doc: Bajo este directorio se encuentra la documentación, que incluye tutoriales, instrucciones sobre el uso de los diferentes módulos y enlaces a la web con la documentación oficial. Los contenidos de esta carpeta son prácticamente los mismos que los de la página de documentación, se descargan por la utilidad de no necesitar estar conectado a Internet para acceder a ellos.
- examples: Contiene ejemplos que pueden resultar útiles como plantillas iniciales para desarrollos.
- os: Contiene el código fuente de Contiki-ng.
- tests: Contiene pruebas unitarias y los ficheros Makefile necesarios para ejecutarlas y comprobar que no hay fallos en la instalación de Contiki-ng.
- tools: Contiene herramientas que pueden facilitar el desarrollo de nuevas aplicaciones. Bajo este escritorio se encuentran los métodos de instalación por Docker y por Vagrant, además del simulador Cooja, que es el que se ha utilizado en este trabajo.

2.2 Cooja

Cooja es un simulador de red incluido en Contiki-ng, sirve para realizar simulaciones de escenarios en los que las motas estarán ejecutando la aplicación cargada en Contiki-ng.

Se ha utilizado el simulador Cooja por ser intuitivo a la hora de crear los escenarios, por venir incluido en la descarga de Contiki-ng y, por último, por haber sido la elección hecha en [1] para simular los escenarios.

2.3 Visualización de resultados: Elasticsearch y Kibana

Para la representación visual de los resultados de las simulaciones de los escenarios, se ha optado por utilizar Elasticsearch para la ingesta de datos y Kibana para su visualización. Una selección habitual en entornos empresariales es utilizar Logstash, Elasticsearch y Kibana, ya que Logstash permite que Elasticsearch ingiera ficheros de trazas independientemente de su formato. En este trabajo, por el contrario, en lugar de Logstash se ha optado por crear un script que transforme las trazas del módulo Energest a un fichero .csv, que Elasticsearch sabe tratar directamente. La razón de esta decisión es por la utilidad de aprender AWK, un lenguaje de programación enfocado al procesamiento de texto, pero Logstash es la opción más potente y versátil. Para este caso, como la cantidad de datos que se obtienen de las simulaciones no es exageradamente grande, es viable aún el uso del script que se ha desarrollado.

2.3.1 Elasticsearch

Elasticsearch es un motor de búsqueda capaz de indexar datos de todo tipo: texto estructurado y no estructurado, datos numéricos, trazas de errores... Con Elasticsearch también es posible realizar análisis de

los datos ingeridos para encontrar patrones, tendencias y encontrar relaciones en estos datos para realizar búsquedas veloces. Debido a que es un motor de búsqueda distribuido, es fácil escalar Elasticsearch cuando el volumen de datos que ingiere pasa a ser mayor que el que puede almacenar el servidor o servidores donde se encuentra.

2.3.2 Kibana

Kibana es una herramienta de visualización de datos que permite interactuar con los datos ingeridos por Elasticsearch. Con Kibana es posible crear gráficas y representaciones de los datos que ingiere Elasticsearch y, a partir de ese punto, es posible sacar conclusiones de los datos, descubrir vulnerabilidades en los datos introducidos, monitorizar nuestras aplicaciones y descubrir relaciones en la información introducida.

3 Despliegues

Contiki-ng ofrece la posibilidad de desplegarlo de tres maneras distintas: Realizando una instalación nativa en el sistema operativo anfitrión, utilizando una imagen de Docker o creando una máquina virtual y desplegándola con Vagrant. En esta sección se especificarán todos los pasos que hay que seguir para obtener una instalación limpia de cada una de las tres posibilidades, todos los pasos han sido obtenidos de la documentación de contiki-ng. También se especificarán los pasos a seguir para realizar la instalación de Elasticsearch con Kibana, de acuerdo con la documentación de Elastic.

Para el despliegue se ha utilizado un ordenador personal, que utiliza Ubuntu 22.04.

3.1 Instalación nativa o toolchain

1. Se actualiza el administrador de paquetes y se instalan las dependencias:

```
sudo apt update
sudo apt install build-essential doxygen git git-lfs curl wireshark python3-serial srecord rlwrap
```

2. Se instala Java para poder utilizar Cooja:

```
sudo apt install default-jdk ant
update-alternatives –config java
```

3. Se clona el repositorio de contiki-ng:

```
git clone https://github.com/contiki-ng/contiki-ng.git;cd contiki-ng
git submodule update –init –recursive
```

4. Para ejecutar Cooja, es necesario navegar a /home/javier/contiki-ng/tools/cooja y ejecutar el simulador, se hace asi:

```
cd /home/javier/contiki-ng/tools/cooja/
./gradlew run
```

3.2 Instalación utilizando la imagen de Docker

Docker es una plataforma que permite a los programadores crear aplicaciones y desplegarlas mediante el uso de contenedores, que son paquetes ejecutables que contienen las dependencias y componentes

necesarios para que la aplicación funcione independientemente del entorno que la hospede. Contiki-ng tiene una imagen oficial de Docker, que será la utilizada para ejecutar Cooja.

Este método de instalación es el más directo ya que se necesita instalar Docker, descargar la imagen oficial y configurar los alias para poder llamar directamente a Cooja, se puede hacer esto sin necesidad de acceder al contenedor que está ejecutando Contiki-ng.

1. Se instala Docker en la máquina anfitriona, en caso de que haya sido instalado previamente, y se añade el usuario con el que se está trabajando al grupo *docker*:

```
sudo apt-get install docker-ce
sudo usermod -aG docker javier
```

2. Es necesario salir del sistema y volver a entrar como el mismo usuario. Tras ello, hará falta descargar la imagen oficial de Docker de Contiki-ng:

```
docker pull contiker/contiki-ng
```

   Este paso descargará la imagen más reciente de Contiki-ng, es posible escoger una imagen distinta haciendo pull de contiker/contiki-ng:<ETIQUETA>

3. Se clona el repositorio de Contiki-ng, si ya se hizo para la instalación previa este paso puede obviarse:

```
git clone https://github.com/contiki-ng/contiki-ng.git
cd contiki-ng
git submodule update –init –recursive
```

4. Se añaden los alias para facilitar el uso de Cooja con las opciones necesarias. Para ello, se edita el fichero .bashrc para añadir las siguientes líneas:

```
export CNG_PATH=/home/javier/contiki-ng
alias contiker="docker run –privileged –sysctl net.ipv6.conf.all.disable_ipv6=0 –mount type=bind,
source=$CNG_PATH,destination=/home/user/contiki-ng -e DISPLAY=$DISPLAY -v /tmp/.X11-
unix:/tmp/.X11-unix -v /dev/bus/usb:/dev/bus/usb -ti contiker/contiki-ng"
```

   Ahora es posible salir y entrar como nuestro usuario o actualizar el fichero .bashrc:

```
source /home/javier/.bashrc
```

5. Para ejecutar Cooja, es posible utilizar los alias que han sido configurados en el anterior paso:

```
contiker cooja
```

3.3 Instalación en máquina virtual utilizando Vagrant

En esta sección, además de explicarse los pasos a seguir para poder realizar la instalación de Contiki-ng utilizando Vagrant, se explicará qué es Vagrant, haciendo una descripción de las principales características de esta novedosa tecnología. Se hace una explicación más pormenorizada de esta tecnología por no haber sido cubiert previamente en [1].

3.3.1 Vagrant: Definición y características

Vagrant es una herramienta de código abierto que permite configurar y administrar máquinas virtuales de forma consistente, reproducible e independiente del sistema operativo que utilice la máquina anfitriona, unificando en un único proceso de trabajo todas las acciones anteriores. Debido a su alto grado de automatización, se disminuye el tiempo que se tarda en montar un entorno de desarrollo para nuevos usuarios y desaparecen los problemas que pueden darse para un desarrollador en su máquina, pero que otros desarrolladores no pueden reproducir.

Vagrant se utiliza para provisionar entornos de desarrollo que son desplegados en software de virtualización como VirtualBox, VMWare o AWS, aislando las dependencias de la máquina anfitriona y haciendo fácil y rápido el despliegue de nuevos entornos. Para automatizar el provisionamiento de las dependencias, Vagrant es utilizado junto con las herramientas más actuales que tiene la industria a día de hoy, como son scripts bash, módulos de Puppet o Ansible o recetas de Chef. Al utilizarlo en conjunto con estas tecnologías, es posible reservar los recursos necesarios para la máquina virtual -con el software de virtualización correspondiente- y descargar todas las dependencias -con el software de provisionamiento escogido- en un único proceso automatizado que será idéntico en cada máquina donde se utilice. Al conjunto de todos los elementos necesarios para crear un entorno específico en Vagrant -sistema operativo, configuraciones, dependencias software, etc.- se le llama caja. Una caja es el equivalente a una imagen de Docker, generalmente está compuesta por una imagen base de un sistema operativo y un fichero de configuraciones -el Vagrantfile, que será explicado más adelante- que será el que se encargue del redireccionamiento de puertos y la configuración de las carpetas compartidas para comunicar la caja con la máquina anfitriona, la descarga de las dependencias software, la creación de las variables de entorno necesarias, etc. Vagrant tiene un amplio catálogo de cajas disponibles, tanto oficiales como creadas por la comunidad, por lo que es sencillo utilizar una caja preexistente como base para crear la caja que se ajuste a las necesidades de cada proyecto.

Previamente se ha mencionado el Vagrantfile. El Vagrantfile es un fichero de texto en formato Ruby que describe las dependencias y recursos necesarios para crear una caja de forma agnóstica al sistema operativo y dependencias de la máquina anfitriona, por lo que los ficheros Vagrantfile se pueden utilizar en cualquier plataforma que soporte Vagrant.

Puede surgir la duda de por qué utilizar Vagrant si, por lo general, el software de virtualización ya trae scripts que permiten provisionar las máquinas virtuales que se vayan a crear. Vagrant utiliza estos scripts también, pero, además, extiende la utilidad de los mismos haciéndose cargo de la creación de túneles HTTP/S, de las carpetas compartidas para poder acceder a ficheros tanto desde la máquina virtual como desde la anfitriona y detectando y resolviendo problemas específicos que pudiera tener la versión del software de vrtualización que estamos utilizando. Por lo general, los scripts de provisionamiento incluidos con el software de virtualización pueden contener errores que se arreglan en versiones posteriores, pero Vagrant es capaz de detectar estos errores y proporcionar soluciones a los mismos para que, si se intenta utilizar Vagrant sobre, por ejemplo, dos instalaciones de VirtualBox con versiones diferentes, el funcionamiento del entorno desplegado sea consistente en ambas.

3.3.2 Instrucciones para la instalación de Contiki-ng utilizando Vagrant

Contiki-ng ya contiene bajo la carpeta *tools* un fichero Vagrantfile, por lo que, para ejecutar Cooja, los únicos requisitos serán instalar Vagrant, levantar la máquina virtual y, una vez dentro de la misma, instalar Java para poder utilizar Cooja. Los pasos para esto son los siguientes:

1. Se instala Vagrant, en la documentación de Vagrant aparecen los pasos para instalar este software en distintos tipos de distribuciones Linux, en el caso que atañe a este trabajo se utilizarán las instrucciones para Ubuntu/Debian:

```
wget -O- https://apt.releases.hashicorp.com/gpg | sudo gpg –dearmor -o /usr/share/keyrings/hashicorp-archive-keyring.gpg

echo "deb [signed-by=/usr/share/keyrings/hashicorp-archive-keyring.gpg]

https://apt.releases.hashicorp.com $(lsb_release -cs) main"|sudo tee /etc/apt/sources.list.d/hashicorp.list
```

```
sudo apt update && sudo apt install vagrant
```

2. Se instala Virtualbox, lo más sencillo será descargar el fichero .deb correspondiente a la distribución de la máquina anfitriona, abrirlo con el instalador de paquetes y seguir los pasos que nos pida el instalador.
3. Se clona el repositorio de Contiki-ng, si lo ya se hizo para las instalaciones previas este paso puede obviarse:

```
sudo apt-get install git git-lfs
git clone https://github.com/contiki-ng/contiki-ng.git
```

4. Es necesario hacer algunas modificaciones al fichero Vagrantfile que existe en /home/javier/contiki-ng/tools/vagrant para poder utilizar la GUI de VirtualBox, para tener suficientes recursos reservados a nuestra MV cuando se use Cooja y para configurar una carpeta compartida entre la MV y la máquina anfitriona. El fichero Vagrantfile quedará así:

Código 3.1 Contenido del fichero Vagrantfile que se encuentra en /contiki-ng/tools/vagrant.

```
# -*- mode: ruby -*-
# vi: set ft=ruby :

# All Vagrant configuration is done below. The "2" in Vagrant.configure
# configures the configuration version (we support older styles for
# backwards compatibility). Please don't change it unless you know what
# you're doing.
Vagrant.configure("2") do |config|
  # The most common configuration options are documented and commented below.

  # For a complete reference, please see the online documentation at
  # https://docs.vagrantup.com.

  # Every Vagrant development environment requires a box. You can search for
  # boxes at https://vagrantcloud.com/search.
  config.vm.box = "ubuntu/bionic64"

  # Disable automatic box update checking. If you disable this, then
  # boxes will only be checked for updates when the user runs
  # 'vagrant box outdated'. This is not recommended.
  # config.vm.box_check_update = false

  # Create a forwarded port mapping which allows access to a specific port
  # within the machine from a port on the host machine. In the example below,

  # accessing "localhost:8080" will access port 80 on the guest machine.
  # NOTE: This will enable public access to the opened port
  # config.vm.network "forwarded_port", guest: 80, host: 8080

  # Share an additional folder to the guest VM. The first argument is
  # the path on the host to the actual folder. The second argument is
  # the path on the guest to mount the folder. And the optional third
  # argument is a set of non-required options.
  config.vm.synced_folder "../../", "/home/vagrant/contiki-ng"

  # Provider-specific configuration so you can fine-tune various
  # backing providers for Vagrant. These expose provider-specific options.
```

```
  # Example for VirtualBox:
  #
  config.vm.provider "virtualbox" do |vb|
  #   # Display the VirtualBox GUI when booting the machine
   vb.gui = true
  #
  #   # Customize the amount of memory on the VM:
     vb.memory = "4096"
     vb.cpu = "2"
  end
  #
  # View the documentation for the provider you are using for more
  # information on available options.
end
```

5. Se arranca la máquina:

```
cd /home/javier/contiki-ng/tools/vagrant
vagrant up
```

6. Ahora, se instalarán todas las dependencias necesarias para correr Contiki-ng, de momento sin utilizar la interfaz gráfica:

```
vagrant ssh
./contiki-ng/tools/vagrant/bootstrap.sh
exit
```

7. Se carga la imagen y nos conectamos a ella para instalar las dependencias de la interfaz gráfica, la imagen de Contiki-ng utiliza XFCE:

```
vagrant reload
vagrant ssh
sudo ./contiki-ng/tools/vagrant/bootstrap-vbox-with-x.sh
exit
```

8. Ahora se vuelve a cargar la imagen y ya será posible acceder a la máquina:

```
vagrant reload
vagrant up
```

Esto cargará la máquina virtual, si se quisiera utilizar la interfaz gráfica será necesario conectarse utilizando el usuario y contraseña predeterminados, ambos son 'vagrant', y una vez conectados, habrá que ejecutar este comando:

```
sudo startx
```

3.4 Elasticsearch con Kibana

Para desplegar el entorno de visualización se ha creado un fichero docker-compose.yml que automatiza todo el proceso, será necesario situarse en la carpeta donde resida el fichero docker-compose.yml. Este es el contenido del fichero:

Código 3.2 Contenido del fichero docker-compose.yml.

```
name: tfg-visualizacion
services:
  elasticsearch:
    image: docker.elastic.co/elasticsearch/elasticsearch:8.6.0
    environment:
      xpack.security.enabled: "false"
      discovery.type: "single-node"
    networks:
      - elastic
    volumes:
      - esdata:/home/javier/TFG
    ports:
      - 9200:9200
      - 9300:9300
  kibana:
    image: docker.elastic.co/kibana/kibana:8.6.0
    networks:
      - elastic
    ports:
      - 5601:5601
    depends_on:
      - elasticsearch
    #La linea de command es util para resetear el token para cofigurar
        Elasticsearch al desplegar las imagenes del docker compose, si fuese
        necesariohabría que descomentarla para poder generar uno nuevo para el
        usuario elastic, ,que es el predeterminado.
    #command:docker exec -it tfg-visualizacion-elasticsearch-1 /usr/share/
        elasticsearch/bin/elasticsearch-create-enrollment-token -s kibana;
        docker exec -it tfg-visualizacion-elasticsearch-1 /usr/share/
        elasticsearch/bin/elasticsearch-reset-password -u elastic

volumes:
  esdata:
    driver: local

networks:
  elastic:
    driver: bridge
```

Para arrancarar Elasticsearch y Kibana, basta con ejecutar el siguiente comando en la carpeta donde se encuentre el docker-compose.yml :

```
docker compose up
```

Y al cabo de un rato el servicio quedaría desplegado, pudiendo accederse a él desde localhost:5061 .

4 Fase de pruebas

Este capítulo describe las pruebas que se han llevado a cabo para verificar que los resultados obtenidos son independientes del método de despliegue de la aplicación. Siguiendo el método de [1], se utilizará el mismo código de ejemplo que se utilizó previamente en [1], el ejemplo *rdp-udp* , localizado bajo /home/javier/contiki-ng/examples/rpl-udp .

Las pruebas se basarán en simular tres escenarios distintos, cada uno durante tres horas, en cada uno de los tres despliegues hechos. Estas simulaciones generarán trazas relacionadas con el consumo de potencia de cada una de las motas gracias al módulo Energest, lo que se va a comparar serán los valores conseguidos para cada una de las motas en cada tipo de despliegue.

Los escenarios serán distintos a los que se utilizaron en [1], se han generado otros tres escenarios aleatorios, los resultados aun así deben ser consistentes y mostrar que no hay diferencias dependiendo de la forma en que se despliega Contiki-ng.

4.1 Módulo Energest

El módulo Energest puede ser utilizado para implementar estimaciones de la potencia consumida por las motas. Este módulo mide el número de ticks del reloj de la mota, pero los desarrolladores pueden utilizar la información de consumo de potencia que aparece en los catálogos de las motas para las que estén desarrollando para producir una estimación del consumo de potencia de las mismas. El módulo Energest imprimirá trazas correspondientes a los siguientes estados:

- ENERGEST_TYPE_CPU : Muestra el tiempo que la CPU pasa activa.
- ENERGEST_TYPE_LPM : Muestra el tiempo que la CPU pasa trabajando en modo de bajo consumo.
- ENERGEST_TYPE_DEEP_LPM : Muestra el tiempo que la CPU pasa trabajando en modo de bajo consumo profundo. Las motas Z1, que son las que se van a simular, no soportan este modo, por lo que su valor será 0 siempre.
- ENERGEST_TYPE_TRANSMIT : Muestra el tiempo que la radio pasa transmitiendo.
- ENERGEST_TYPE_LISTEN : Muestra el tiempo que la radio pasa escuchando.

Para poder utilizar este módulo en las pruebas del trabajo, va a ser necesario añadir la siguiente línea al comienzo del Makefile que se encuentra en la carpeta *rpl-udp*:

```
MODULES += os/services/simple-energest
```

En [1] también se modificó la constante SIMPLE_ENERGEST_PERIOD dentro de este fichero Makefile para que los resultados se imprimiesen cada hora en lugar de hacerlo cada minuto, como es el valor por defecto. Esto se debe a que en [1] hubo que tratar luego los resultados manualmente, como en este trabajo se dispone de Elasticsearch para realizar cálculos con todos los datos, es posible mantener el periodo cada cuanto se escriben las trazas de Energest en su valor por defecto.

4.2 Creación de los escenarios

En esta sección se detallan todos los pasos que hay que seguir para crear los ficheros .csc que describen los escenarios, y los pasos necesarios para ejecutar las simulaciones.

1. Se accede a Cooja, no importa en cual de las tres instalaciones se entre porque esta sección se va a enfocar en la creación de los escenarios, en la siguiente sección se darán los pasos a seguir para ejecutar las simulaciones.

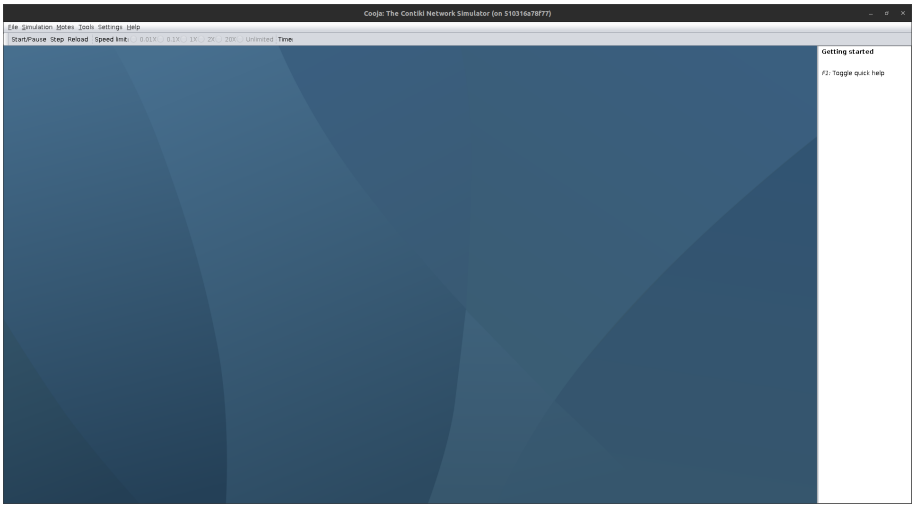

Figura 4.1 Panel inicial de Cooja.

2. Se crea una nueva simulación haciendo click en File/New simulation y se mantienen todos los valores que aparecen por defecto, para cada escenario cambiará únicamente el nombre.

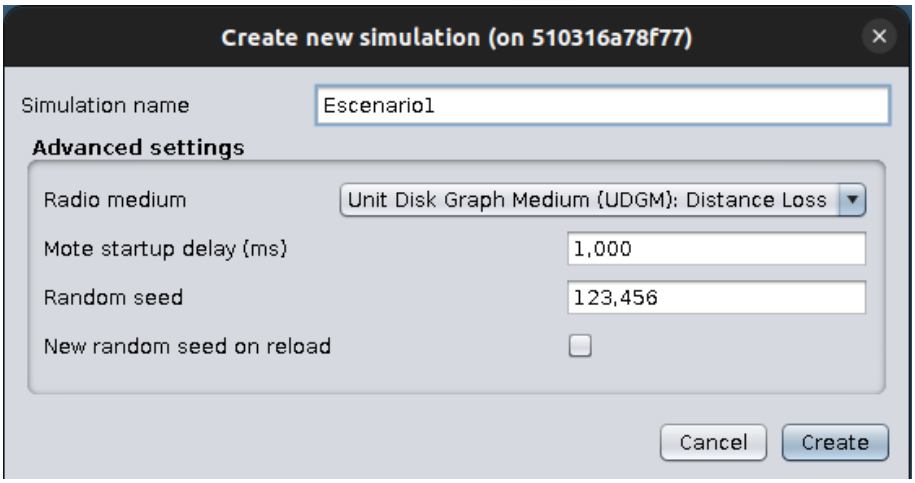

Figura 4.2 Nueva simulación.

3. Se añaden las motas a simular, para ello hay que hacer click en los menús Motes/Add Motes/Create New Mote Type/Z1 Mote. Hará falta crear 1 mota que actúe como servidor y 14 que actúen como clientes, para ello habrá que escoger y compilar el código fuente de cada uno de los dos tipos de motas, en la imagen correspondiente se puede ver un ejemplo de cómo hacerlo para la mota servidor.
4. Tras hacer lo mismo para las motas cliente, se puede ejecutar la simulación. Una vez hayan pasado tres horas se detiene la simulación y, en la ventana de Mote output, se hace click en File/Save to file para guardar las trazas que se van a utilizar.

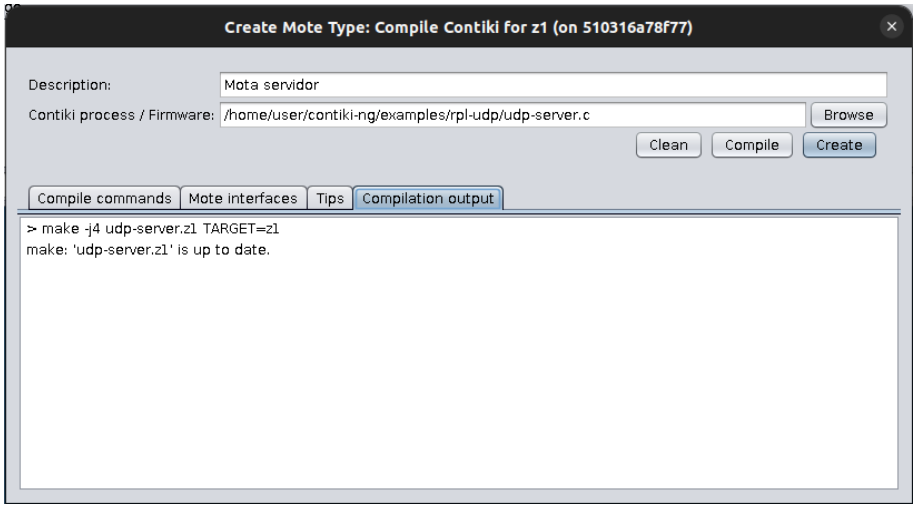

Figura 4.3 Ejemplo de compilación del código fuente de la mota servidor.

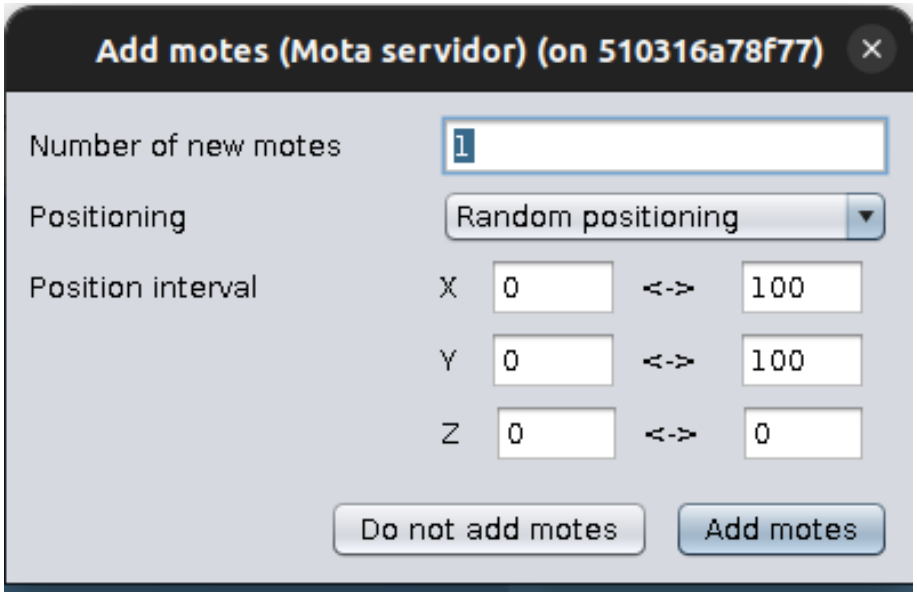

Figura 4.4 Número de motas servidor y posición, dejamos los valores por defecto y añadimos una mota servidor.

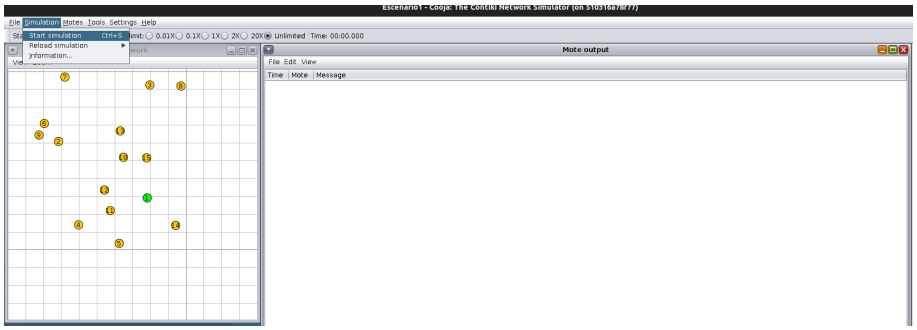

Figura 4.5 Layout de Cooja una vez el escenario ha sido cargado.

Estos pasos nos servirán para crear los tres escenarios que vamos a necesitar y para ejecutarlos una primera vez en el entorno que hayamos elegido primero, después de esto tendremos que cargar los escenarios desde los otros dos entornos y volver a ejecutar las simulaciones.

Los escenarios estarán guardados en ficheros .csc, que realmente son ficheros en formato XML que describen la posición de cada mota, el tipo de mota y demás información pertinente de manera que Cooja sea capaz de crear el escenario a partir de dicha información. Los contenidos de los ficheros de los escearios se encuentran en los anexos.

4.3 Resultados de las pruebas

Una vez obtenidas las trazas de las simulaciones, habrá que generar los ficheros .csv para cada escenario. Cada uno de estos ficheros será el resultado de correr el script *transformador.sh* escogiendo el escenario que se desee. La estructura de carpetas que es necesaria para el correcto funcionamiento del script es la siguiente:

Código 4.1 Estructura de directorios utilizada para ejecutar el script que transforma las trazas Energest a un fichero en formato .csv.

```
javier@TFG:~/contiki-ng/tools/cooja/TFG$ ll -h ./*
-rw-rw-r-- 1 javier javier 2,7M mar 12 18:33 ./Metricas_Escenario1_2023-03-12.
    csv
-rw-rw-r-- 1 javier javier 2,7M mar 12 18:33 ./Metricas_Escenario2_2023-03-12.
    csv
-rw-rw-r-- 1 javier javier 2,7M mar 12 18:33 ./Metricas_Escenario3_2023-03-12.
    csv
-rwxrw-r-- 1 javier javier 2,0K mar 12 18:32 ./transformador.sh

./Escenarios:
total 44K
drwxrwxr-x 2 javier javier 4,0K feb 25 12:01 ./
drwxrwxr-x 6 javier javier 4,0K mar 12 19:00 ../
-rw-rw-r-- 1 javier javier 12K ene 6 12:21 Escenario1.csc
-rw-rw-r-- 1 javier javier 12K feb 8 22:28 Escenario2.csc
-rw-rw-r-- 1 javier javier 12K feb 8 22:29 Escenario3.csc

./Output_docker:
total 6,6M
drwxrwxr-x 2 javier javier 4,0K mar 12 14:12 ./
drwxrwxr-x 6 javier javier 4,0K mar 12 19:00 ../
-rw-r--r-- 1 javier javier 2,2M mar 6 12:20 Escenario1_docker.txt
-rw-r--r-- 1 javier javier 2,1M mar 6 13:02 Escenario2_docker.txt
-rw-r--r-- 1 javier javier 2,3M mar 10 10:37 Escenario3_docker.txt

./Output_toolchain:
total 7,2M
drwxrwxr-x 2 javier javier 4,0K mar 12 11:33 ./
drwxrwxr-x 6 javier javier 4,0K mar 12 19:00 ../
-rw-rw-r-- 1 javier javier 3,0M mar 3 08:39 Escenario1_toolchain.txt
-rw-rw-r-- 1 javier javier 2,1M mar 3 11:47 Escenario2_toolchain.txt
-rw-rw-r-- 1 javier javier 2,2M mar 3 10:47 Escenario3_toolchain.txt

./Output_vagrant:
total 6,5M
drwxrwxr-x 2 javier javier 4,0K mar 12 11:33 ./
drwxrwxr-x 6 javier javier 4,0K mar 12 19:00 ../
```

```
-rw-r--r-- 1 javier javier 2,3M mar 10 11:45 Escenario1_vagrant.txt
-rw-r--r-- 1 javier javier 2,1M mar 10 12:18 Escenario2_vagrant.txt
-rw-r--r-- 1 javier javier 2,1M mar 10 12:54 Escenario3_vagrant.txt
```

Donde las carpetas Output_* contienen las trazas para cada escenario, desde cada uno de los despliegues. El código del script transformador.sh estará adjuntado en el anexo, los ficheros Metricas_Escenario* son los ficheros generados por el script, que se subirán a Elasticsearch.

Los ficheros generados contendrán los siguientes campos:

- timestamp: Marca de tiempo, en formato [h:]mm:ss , que indica el instante de tiempo en que el módulo Energest envió la traza.
- id: Identificador de la mota, en el caso de nuestros escenarios se identifican con un número del 1 al 15.
- metrica: El tipo de dato, este campo es equivalente a cada uno de los estados de Energest, vistos antes en este capítulo.
- valor_medido: La medida registrada por la mota en el último periodo, es un número de ticks del reloj de la mota.
- total: Total de ticks de reloj que ha realizado la mota en el último periodo.
- origen: El despliegue del que viene el dato, es el nombre del fichero del que se ha recopilado la información.

4.4 Visualización de los resultados

Un vez generados los ficheros .csv , es hora de crearlas visualizaciones en Kibana. Para ello, inicializamos el entorno de visualización ejecutando el comando

```
docker compose up
```

en el directorio en que se encuentre docker-compose.yml . Para acceder a Elasticsearch, se debe utilizar un navegador y acceder a localhost:5601

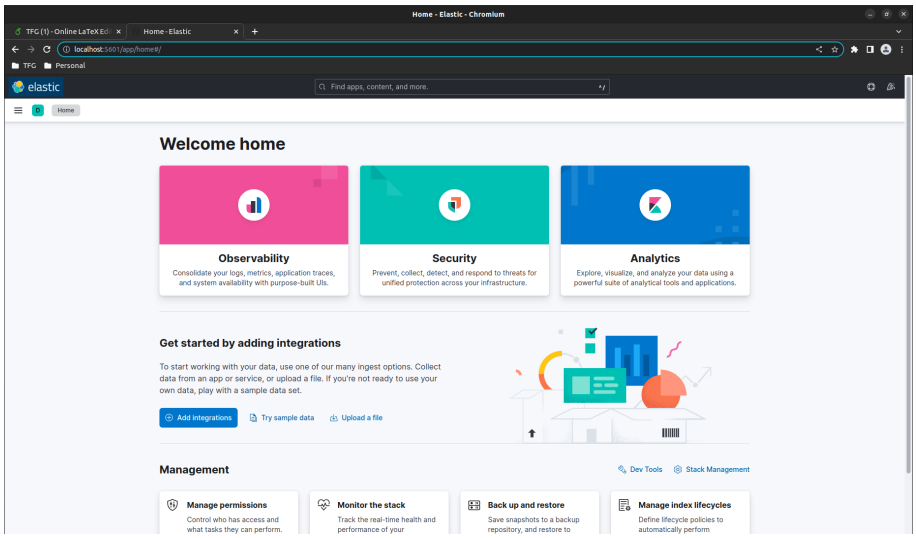

Figura 4.6 Página principal de Elasticsearch.

Para subir nuestros ficheros, deberemos hacer click en el botón *Upload a file* . Tras seleccionar el fichero del escenario a subir aparecerá una pantalla con resumenes de la información subida. Se puede obviar todo esto y darle directaemente al botón de importar, en el campo *index name* se deberá poner un identificador de nuestros datos, en este caso se va a llamar a cada escenario escenario1 , escenario2 o escenario3.

Una vez hecho esto, tenemos los datos subidos y podemos crear las visualizaciones de los mismos. Para hacer esto, navegamos a Analytics/Dashboard y crearemos un nuevo dashboard. En el dashboard, crearemos

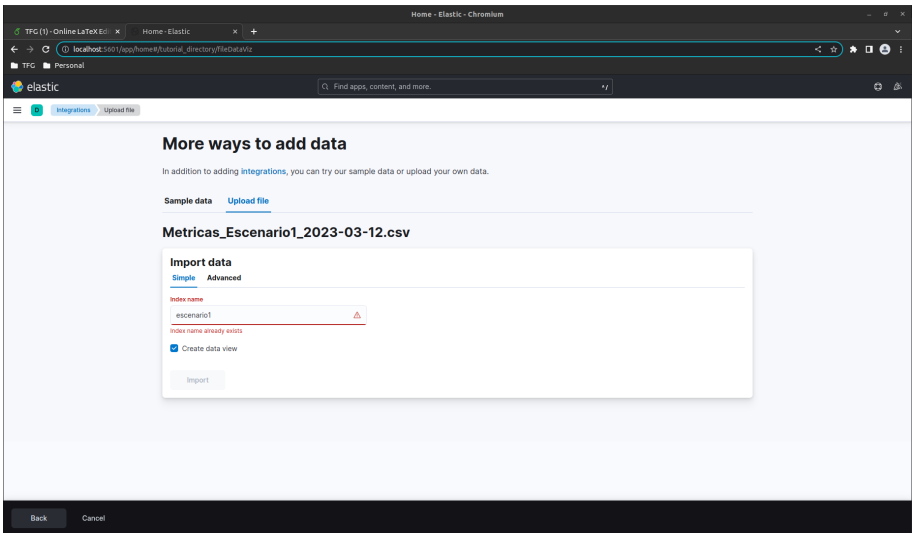

Figura 4.7 Pantalla final para añadir datos a Elasticsearch.

una visualización para la media, otra para la desviación estándar y otra para la mediana de los valores de cada mota en cada escenario. En el eje vertical mostraremos el número de ticks de reloj que ha pasado cada mota en el estado que represente el dashboard, en el eje horizontal representaremos el identificador de cada mota, habrá una barra vertical para cada mota con los valores en cada entorno. Configurar esto es sencillo, ya que puede hacerse mediante *drag and drop*, arrastrando los campos de la izquierda al eje deseado, en el lado derecho.

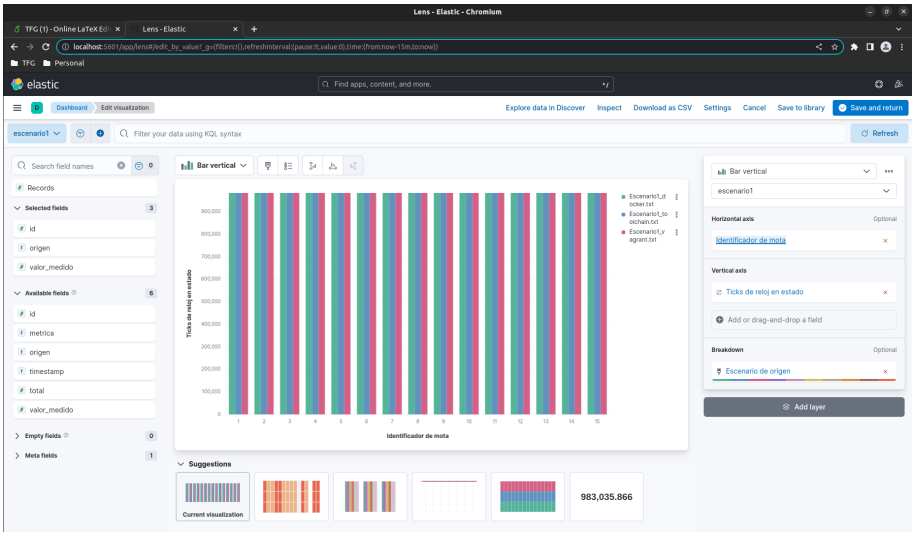

Figura 4.8 Ejemplo de gráfica, utilizada para mostrar la media en el escenario 1.

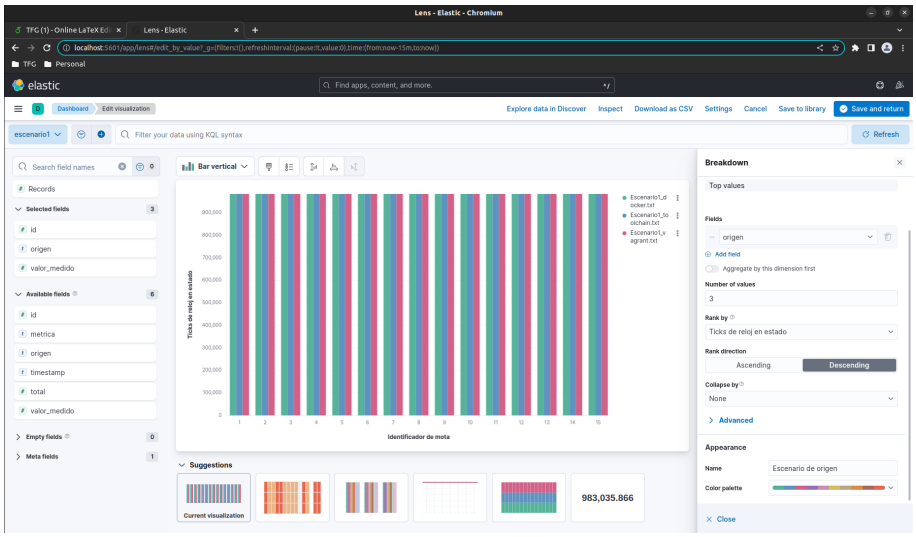

Figura 4.9 Configuración del apartado *Breakdown* para mostrar una columna por el valor de cada despliegue.

En el nivel de dashboard es posible crear un filtro que aplique a todas nuestras gráficas, donde el campo métrica mostrará el valor del estado que se desee ver. Habrá valores para CPU, LPM, Radio RX y Radio TX, pero no para Deep LPM porque, por especificaciones de las motas simuladas, este valor es de 0 siempre.

Una vez creados los filtros se creará un dashboard para cada estado, para cada escenario, por lo que al final tendremos un total de 12 dashboards. La decisión de utilizar los dashboards ha sido dada porque la otra opción de visualización disponible, los canvas, están pensados para ser utilizados en datos que se aactualizan en tiempo real. Como los datos no van a cambiar una vez han sido subidos, la mejor opción será utilizar dashboards, que permiten la representación del contenido de una manera más estática.

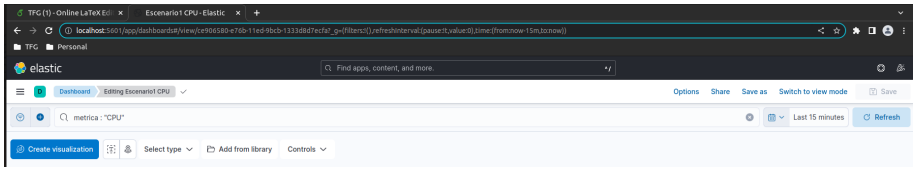

Figura 4.10 Filtro global que hay que aplicar para mostrar el estado en todas las gráficas del dashboard, en este caso para la CPU del escenario 1.

A continuación se han añadido imágenes con los dashboards que he generado para la comparación de los resultados, en ellos se pueden ver tres columnas para cada mota, cada una de ellas corresponde al despliegue en el que se ha ejecutado la simulación de la que se han obtenido los datos.

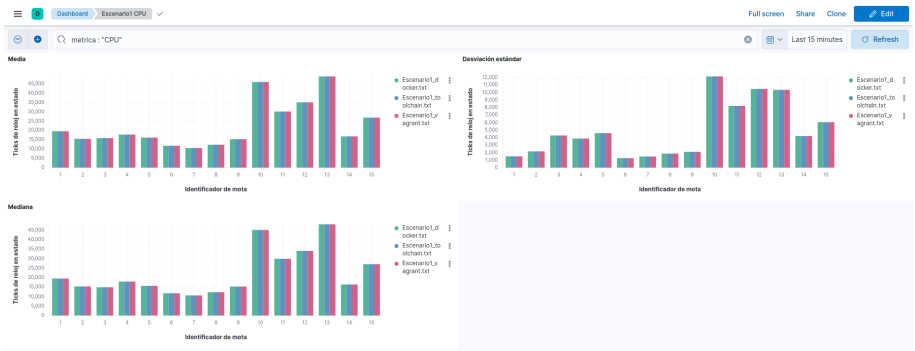

Figura 4.11 Dashboard que muestra el tiempo pasado con la CPU activa en el escenario 1, en ticks de reloj.

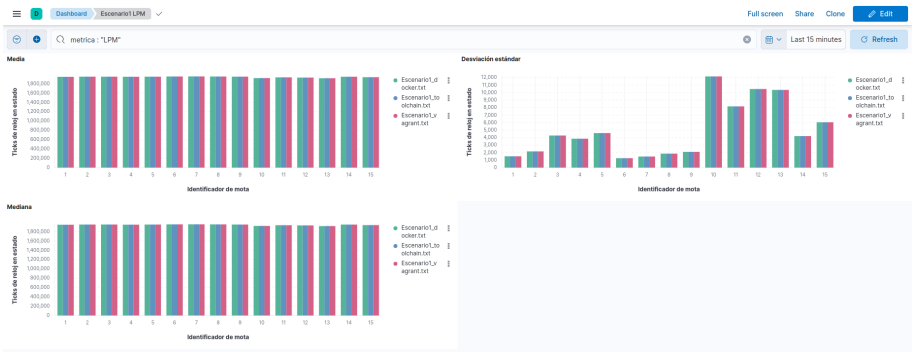

Figura 4.12 Dashboard que muestra el tiempo pasado en modo de baja potencia en el escenario 1, en ticks de reloj.

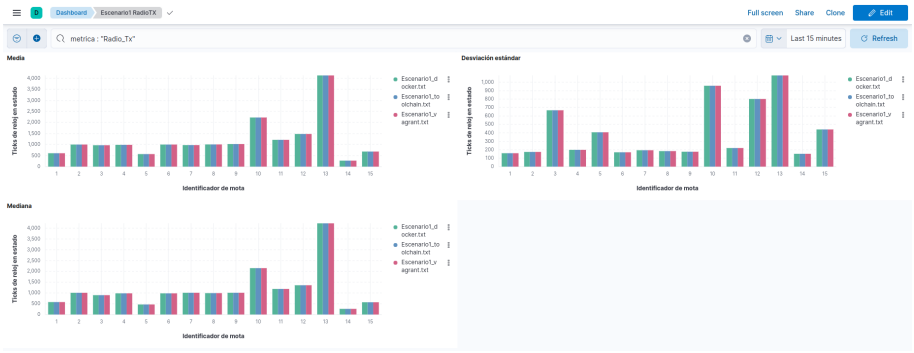

Figura 4.13 Dashboard que muestra el tiempo pasado transmitiendo por radio en el escenario 1, en ticks de reloj.

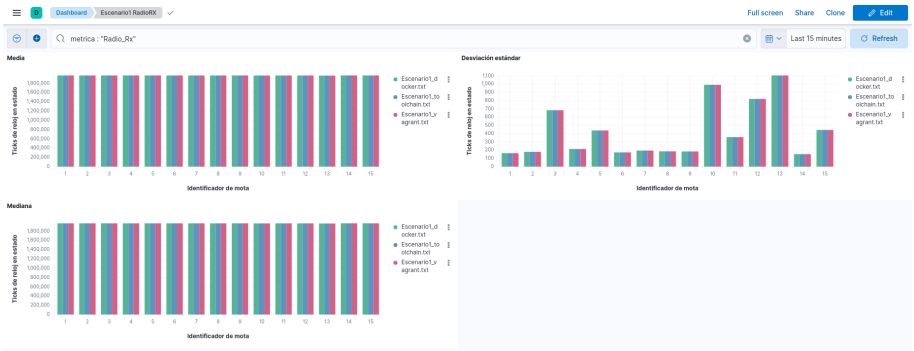

Figura 4.14 Dashboard que muestra el tiempo pasado escuchando por radio en el escenario 1, en ticks de reloj.

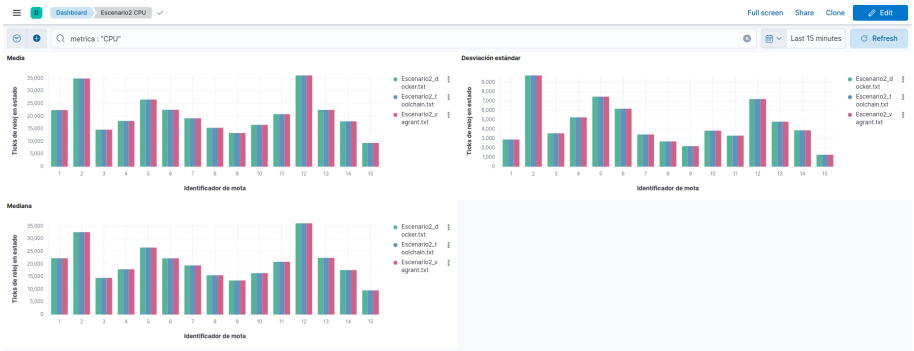

Figura 4.15 Dashboard que muestra el tiempo pasado con la CPU activa en el escenario 2, en ticks de reloj.

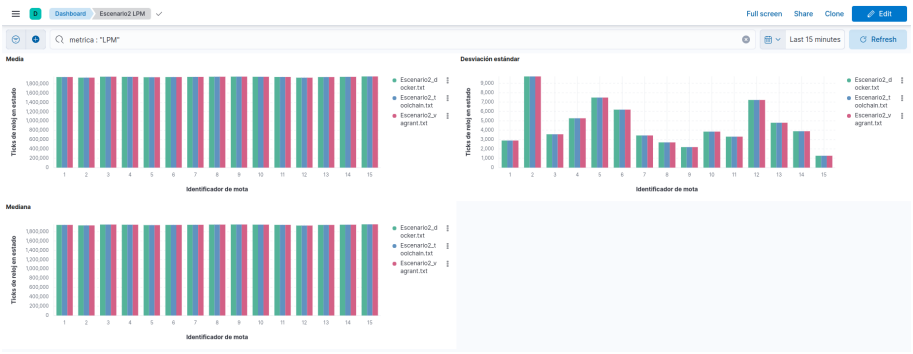

Figura 4.16 Dashboard que muestra el tiempo pasado en modo de baja potencia en el escenario 2, en ticks de reloj.

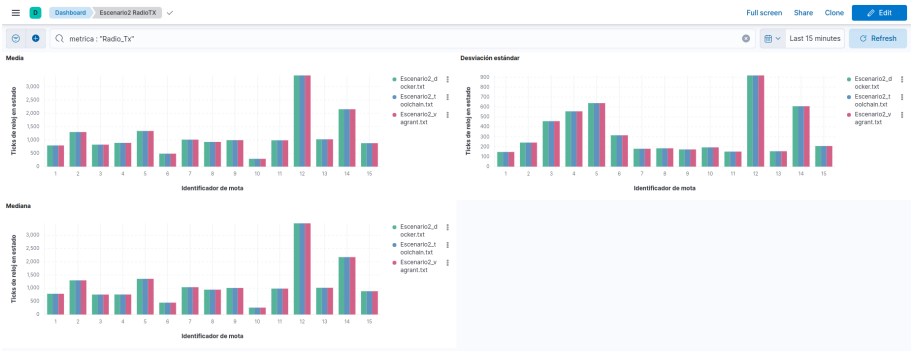

Figura 4.17 Dashboard que muestra el tiempo pasado transmitiendo por radio en el escenario 2, en ticks de reloj.

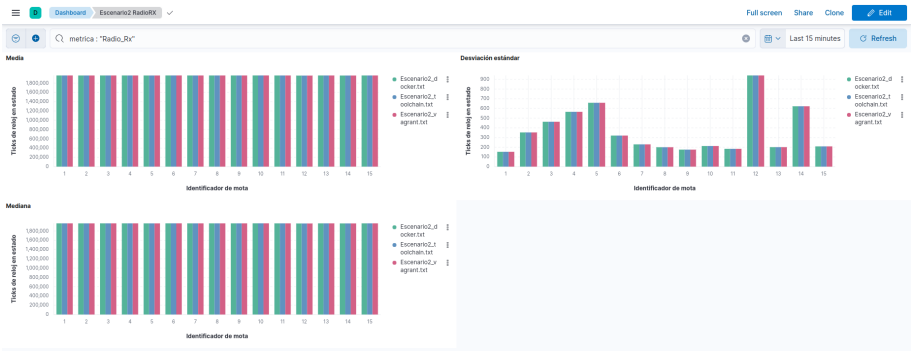

Figura 4.18 Dashboard que muestra el tiempo pasado escuchando por radio en el escenario 2, en ticks de reloj.

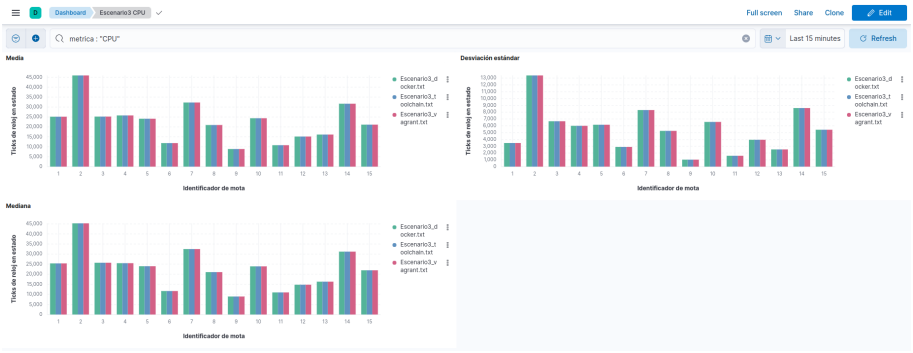

Figura 4.19 Dashboard que muestra el tiempo pasado con la CPU activa en el escenario 3, en ticks de reloj.

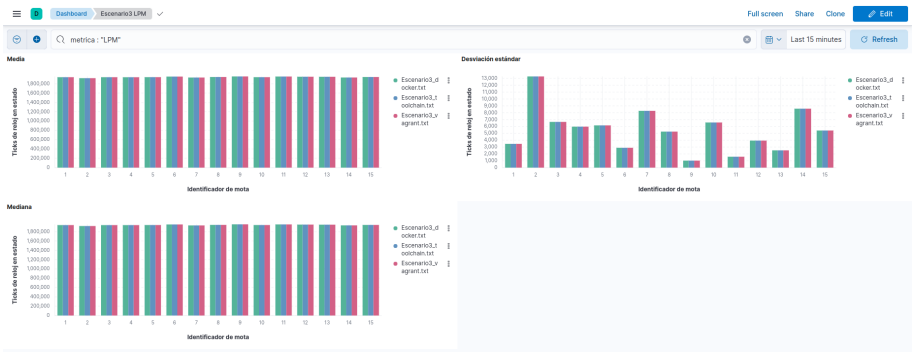

Figura 4.20 Dashboard que muestra el tiempo pasado en modo de baja potencia en el escenario 3, en ticks de reloj.

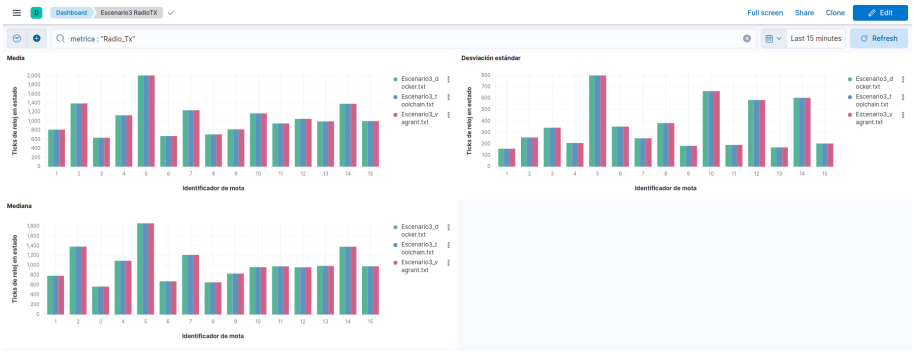

Figura 4.21 Dashboard que muestra el tiempo pasado transmitiendo por radio en el escenario 3, en ticks de reloj.

Figura 4.22 Dashboard que muestra el tiempo pasado escuchando por radio en el escenario 3, en ticks de reloj.

5 Conclusiones

Habiendo obtenido los resultados, lo único que resta es sacar conclusiones de los mismos. En este apartado se analiza el significado de los resultados obtenidos, se harán algunos apuntes referentes a las decisiones tomadas durante el trabajo y se ofrecen ideas sobre posibles continuaciones para el mismo.

5.1 Análisis de los resultados

Una vez revisamos las gráficas generadas para cada escenario, se puede observar que no existen diferencias en los resultados entre los tres tipos de despliegues.

¿A qué puede deberse entonces que en [1] se encontrasen diferencias en los resultados de hasta el 6 %? Esto puede haber sido una consecuencia de haber cambiado el periodo cada cuanto se imprimen las trazas del módulo Energest, de 1 minuto a 1 hora. Cooja debe realizar las operaciones de agregación para devolver a cada hora el valor del número de ticks que pasa una mota en un determinado estado, y para devolver este valor debe promediar los valores obtenidos a cada minuto. Si los valores son ligeramente distintos y en la operaciones realizadas para obtener el promedio hay errores de redondeo debido a la precisión de la mantisa con la que trabaja Cooja, es posible que los resultados finales presenten unas diferencias como las experimentadas en [1], de hasta el 6 %. Esto no sucede con la nueva forma de obtener los resultados porque Elasticsearch es muy preciso a la hora de realizar estas operaciones, es una de sus características, la de obtener conclusiones sobre un conjunto de datos de forma rápida y precisa. El promedio que se obtiene con el nuevo método es el mismo independientemente del despliegue con el que se obtengan los datos porque no hay errores de redondeo, lo que nos permite afirmar que podemos utilizar Contiki-ng independientemente del método que escojamos para su instalación, ya que los resultados no son dependientes del método de despliegue.

5.2 Valoración de los métodos de despliegue

Sabiendo que la instalación de Contiki-ng no afecta a los resultados, puede surgir la pregunta de cual de los tres métodos preferimos utilizar.

- Toolchain: Este método es el menos aconsejable, es laborioso instalar Contiki-ng siguiendo unos pasos que no están automatizados y, de los tres, es el único que puede dar problemas de dependencias, ya que depende de los paquetes que tengas instalados en tu máquina personal.
- Vagrant: Si bien el uso de esta herramienta es muy útil para minimizar el tiempo que deben pasar los programadores preparando el entorno de desarrollo, esta opción tiene algunos inconvenientes. Si se quisiera utilizar Cooja desde la imagen de Vagrant, hará falta añadir las siguientes líneas al fichero /home/javier/contiki-ng/tools/vagrant/bootstrap.sh

```
sudo apt install default-jdk ant
update-alternatives –config java
```

De otro modo, la imagen no trae la máquina virtual de Java, por lo que aparecerán errores cuando se intente ejecutar Cooja.

Aparte de este inconveniente, si se desea utilizar Cooja también será necesario que la máquina virtual tenga una interfaz gráfica de usuario, por lo que será necesario reservar más recursos, ya que se simulará también la interfaz gráfica XFCE. Si el ordenador anfitrión tiene suficientes recursos esto no será un problema, pero en caso contrario puede ser un grave inconveniente.

Por último, leyendo nuevamente el fichero /home/javier/contiki-ng/tools/vagrant/bootstrap.sh se puede observar que la máquina virtual de Vagrant utiliza la imagen de Docker, por lo que se podría considerar redundante utilizar Vagrant si existiese la posibilidad de utilizar simplemente la imagen de Docker.

Otro punto a tener en cuenta es que la maquina virtual utiliza el teclado con la disposición de las teclas estadounidense, es otra configuración que potencialmente debería cambiarse si los desarrolladores no utilizan la misma disposición.

- Docker: Esta es la opción más aconsejable, el único prerrequisito es tener Docker instalado en nuestro ordenador, nos ahorraremos problemas de dependencias como con el método de la toolchain o de recursos adicionales como potencialmente podría suceder con Vagrant. Además, sería posible automatizar el despliegue de la imagen de Docker y del entorno de visualización añadiendo la configuración correspondiente al fichero docker-compose.yml que ya se ha creado. La ejecución con Docker ha sido la más rápida, sencilla y la que menos problemas ha dado, por lo que, si tuviésemos que escoger un método de entre los tres, este sería el más aconsejable dadas las circunstancias.

5.3 Proyectos futuros

Teniendo en cuenta la forma más ventajosa de instalar Contiki-ng y desplegar un entorno de visualización de los resultados, la continuación más obvia de este trabajo sería realizar un desarrollo con estas herramientas.

Otra posibilidad sería completar la automatización total del despliegue, utilizando Ansible para ello, y además poder dar rienda suelta a todas las capacidades de Elasticsearch utilizando el conjunto de servicios Elasticsearch, Logstash y Kibana, tanto para automatizar la ingesta de los resultados como para monitorizar el rendimiento del entorno, creando visualizaciones de métricas en tiempo real como podrían ser el uso de memoria, de CPU o de disco de nuestro ordenador mientras ejecutamos una simulación.

Una última idea sería configurar también un servidor de integración continua, como Jenkins, para automatizar la compilación, las pruebas y la validación del código.

5.4 Reflexión personal

Este trabajo de fin de grado demuestra que con las herramientas adecuadas y una metodología de trabajo eficiente, es posible minimizar el tiempo que se tarda en desarrollar soluciones tecnológicas de alta calidad a un coste razonable. Además, la automatización de procesos y el uso de herramientas de integración de los resultados son clave para acelerar el desarrollo y mejorar la calidad del software producido, por lo que este trabajo aporta una contribución significativa al campo de la informática y puede servir de inspiración para futuros proyectos de desarrollo de software.

Apéndice A
Escenarios de las pruebas

Código A.1 Contenido del fichero Escenario1.csc.

```
<?xml version="1.0" encoding="UTF-8"?>
<simconf>
  <simulation>
    <title>Escenario1</title>
    <randomseed>123456</randomseed>
    <motedelay_us>1000000</motedelay_us>
    <radiomedium>
      org.contikios.cooja.radiomediums.UDGM
      <transmitting_range>50.0</transmitting_range>
      <interference_range>100.0</interference_range>
      <success_ratio_tx>1.0</success_ratio_tx>
      <success_ratio_rx>1.0</success_ratio_rx>
    </radiomedium>
    <events>
      <logoutput>40000</logoutput>
    </events>
    <motetype>
      org.contikios.cooja.mspmote.Z1MoteType
      <identifier>z1232598741</identifier>
      <description>Z1 server</description>
      <source>[CONTIKI_DIR]/examples/rpl-udp/udp-server.c</source>
      <commands>make -j$(CPUS) udp-server.z1 TARGET=z1</commands>
      <firmware>[CONTIKI_DIR]/examples/rpl-udp/build/z1/udp-server.z1</firmware
          >
      <moteinterface>org.contikios.cooja.interfaces.Position</moteinterface>
      <moteinterface>org.contikios.cooja.interfaces.RimeAddress</moteinterface>
      <moteinterface>org.contikios.cooja.interfaces.IPAddress</moteinterface>
      <moteinterface>org.contikios.cooja.interfaces.Mote2MoteRelations</
          moteinterface>
      <moteinterface>org.contikios.cooja.interfaces.MoteAttributes</
          moteinterface>
      <moteinterface>org.contikios.cooja.mspmote.interfaces.MspClock</
          moteinterface>
      <moteinterface>org.contikios.cooja.mspmote.interfaces.MspMoteID</
          moteinterface>
      <moteinterface>org.contikios.cooja.mspmote.interfaces.MspButton</
          moteinterface>
```

```
      <moteinterface>org.contikios.cooja.mspmote.interfaces.Msp802154Radio</
          moteinterface>
      <moteinterface>org.contikios.cooja.mspmote.interfaces.MspDefaultSerial</
          moteinterface>
      <moteinterface>org.contikios.cooja.mspmote.interfaces.MspLED</
          moteinterface>
      <moteinterface>org.contikios.cooja.mspmote.interfaces.MspDebugOutput</
          moteinterface>
    </motetype>
    <motetype>
      org.contikios.cooja.mspmote.Z1MoteType
      <identifier>z1270566932</identifier>
      <description>Z1 Client</description>
      <source>[CONTIKI_DIR]/examples/rpl-udp/udp-client.c</source>
      <commands>make -j$(CPUS) udp-client.z1 TARGET=z1</commands>
      <firmware>[CONTIKI_DIR]/examples/rpl-udp/build/z1/udp-client.z1</firmware
          >
      <moteinterface>org.contikios.cooja.interfaces.Position</moteinterface>
      <moteinterface>org.contikios.cooja.interfaces.RimeAddress</moteinterface>
      <moteinterface>org.contikios.cooja.interfaces.IPAddress</moteinterface>
      <moteinterface>org.contikios.cooja.interfaces.Mote2MoteRelations</
          moteinterface>
      <moteinterface>org.contikios.cooja.interfaces.MoteAttributes</
          moteinterface>
      <moteinterface>org.contikios.cooja.mspmote.interfaces.MspClock</
          moteinterface>
      <moteinterface>org.contikios.cooja.mspmote.interfaces.MspMoteID</
          moteinterface>
      <moteinterface>org.contikios.cooja.mspmote.interfaces.MspButton</
          moteinterface>
      <moteinterface>org.contikios.cooja.mspmote.interfaces.Msp802154Radio</
          moteinterface>
      <moteinterface>org.contikios.cooja.mspmote.interfaces.MspDefaultSerial</
          moteinterface>
      <moteinterface>org.contikios.cooja.mspmote.interfaces.MspLED</
          moteinterface>
      <moteinterface>org.contikios.cooja.mspmote.interfaces.MspDebugOutput</
          moteinterface>
    </motetype>
    <mote>
      <interface_config>
        org.contikios.cooja.interfaces.Position
        <x>78.34493832597914</x>
        <y>70.79232895470153</y>
        <z>0.0</z>
      </interface_config>
      <interface_config>
        org.contikios.cooja.mspmote.interfaces.MspMoteID
        <id>1</id>
      </interface_config>
      <motetype_identifier>z1232598741</motetype_identifier>
    </mote>
    <mote>
      <interface_config>
        org.contikios.cooja.interfaces.Position
        <x>28.484206374034848</x>
        <y>39.26230570298483</y>
```

```
      <z>0.0</z>
    </interface_config>
    <interface_config>
      org.contikios.cooja.mspmote.interfaces.MspMoteID
      <id>2</id>
    </interface_config>
    <motetype_identifier>z1270566932</motetype_identifier>
  </mote>
  <mote>
    <interface_config>
      org.contikios.cooja.interfaces.Position
      <x>79.63561266177469</x>
      <y>7.577351208640293</y>
      <z>0.0</z>
    </interface_config>
    <interface_config>
      org.contikios.cooja.mspmote.interfaces.MspMoteID
      <id>3</id>
    </interface_config>
    <motetype_identifier>z1270566932</motetype_identifier>
  </mote>
  <mote>
    <interface_config>
      org.contikios.cooja.interfaces.Position
      <x>39.80347763411617</x>
      <y>86.04594372172095</y>
      <z>0.0</z>
    </interface_config>
    <interface_config>
      org.contikios.cooja.mspmote.interfaces.MspMoteID
      <id>4</id>
    </interface_config>
    <motetype_identifier>z1270566932</motetype_identifier>
  </mote>
  <mote>
    <interface_config>
      org.contikios.cooja.interfaces.Position
      <x>62.56355164968419</x>
      <y>96.28465209388494</y>
      <z>0.0</z>
    </interface_config>
    <interface_config>
      org.contikios.cooja.mspmote.interfaces.MspMoteID
      <id>5</id>
    </interface_config>
    <motetype_identifier>z1270566932</motetype_identifier>
  </mote>
  <mote>
    <interface_config>
      org.contikios.cooja.interfaces.Position
      <x>20.402519828751597</x>
      <y>29.141447133772957</y>
      <z>0.0</z>
    </interface_config>
    <interface_config>
      org.contikios.cooja.mspmote.interfaces.MspMoteID
      <id>6</id>
```

```
    </interface_config>
    <motetype_identifier>z1270566932</motetype_identifier>
  </mote>
  <mote>
    <interface_config>
      org.contikios.cooja.interfaces.Position
      <x>32.11298110513808</x>
      <y>3.1389019147770814</y>
      <z>0.0</z>
    </interface_config>
    <interface_config>
      org.contikios.cooja.mspmote.interfaces.MspMoteID
      <id>7</id>
    </interface_config>
    <motetype_identifier>z1270566932</motetype_identifier>
  </mote>
  <mote>
    <interface_config>
      org.contikios.cooja.interfaces.Position
      <x>97.19766503627028</x>
      <y>8.214536251040993</y>
      <z>0.0</z>
    </interface_config>
    <interface_config>
      org.contikios.cooja.mspmote.interfaces.MspMoteID
      <id>8</id>
    </interface_config>
    <motetype_identifier>z1270566932</motetype_identifier>
  </mote>
  <mote>
    <interface_config>
      org.contikios.cooja.interfaces.Position
      <x>17.45021662313915</x>
      <y>35.681153376311</y>
      <z>0.0</z>
    </interface_config>
    <interface_config>
      org.contikios.cooja.mspmote.interfaces.MspMoteID
      <id>9</id>
    </interface_config>
    <motetype_identifier>z1270566932</motetype_identifier>
  </mote>
  <mote>
    <interface_config>
      org.contikios.cooja.interfaces.Position
      <x>64.86427282128533</x>
      <y>48.20819669712445</y>
      <z>0.0</z>
    </interface_config>
    <interface_config>
      org.contikios.cooja.mspmote.interfaces.MspMoteID
      <id>10</id>
    </interface_config>
    <motetype_identifier>z1270566932</motetype_identifier>
  </mote>
  <mote>
    <interface_config>
```

```
      org.contikios.cooja.interfaces.Position
      <x>57.517880050669945</x>
      <y>78.1077145881477</y>
      <z>0.0</z>
    </interface_config>
    <interface_config>
      org.contikios.cooja.mspmote.interfaces.MspMoteID
      <id>11</id>
    </interface_config>
    <motetype_identifier>z1270566932</motetype_identifier>
  </mote>
  <mote>
    <interface_config>
      org.contikios.cooja.interfaces.Position
      <x>54.08554235297236</x>
      <y>66.38215809921086</y>
      <z>0.0</z>
    </interface_config>
    <interface_config>
      org.contikios.cooja.mspmote.interfaces.MspMoteID
      <id>12</id>
    </interface_config>
    <motetype_identifier>z1270566932</motetype_identifier>
  </mote>
  <mote>
    <interface_config>
      org.contikios.cooja.interfaces.Position
      <x>63.098017694342865</x>
      <y>33.41393640357406</y>
      <z>0.0</z>
    </interface_config>
    <interface_config>
      org.contikios.cooja.mspmote.interfaces.MspMoteID
      <id>13</id>
    </interface_config>
    <motetype_identifier>z1270566932</motetype_identifier>
  </mote>
  <mote>
    <interface_config>
      org.contikios.cooja.interfaces.Position
      <x>94.06709523314973</x>
      <y>86.20787013920193</y>
      <z>0.0</z>
    </interface_config>
    <interface_config>
      org.contikios.cooja.mspmote.interfaces.MspMoteID
      <id>14</id>
    </interface_config>
    <motetype_identifier>z1270566932</motetype_identifier>
  </mote>
  <mote>
    <interface_config>
      org.contikios.cooja.interfaces.Position
      <x>78.0397952465744</x>
      <y>48.52616592488786</y>
      <z>0.0</z>
    </interface_config>
```

```
      <interface_config>
        org.contikios.cooja.mspmote.interfaces.MspMoteID
        <id>15</id>
      </interface_config>
      <motetype_identifier>z1270566932</motetype_identifier>
    </mote>
  </simulation>
  <plugin>
    org.contikios.cooja.plugins.Visualizer
    <plugin_config>
      <moterelations>true</moterelations>
      <skin>org.contikios.cooja.plugins.skins.IDVisualizerSkin</skin>
      <skin>org.contikios.cooja.plugins.skins.GridVisualizerSkin</skin>
      <skin>org.contikios.cooja.plugins.skins.TrafficVisualizerSkin</skin>
      <skin>org.contikios.cooja.plugins.skins.UDGMVisualizerSkin</skin>
      <skin>org.contikios.cooja.plugins.skins.MoteTypeVisualizerSkin</skin>
      <viewport>3.3769168635243383 0.0 0.0 3.3769168635243383
          0.42181752849880594 5.127461918313188</viewport>
    </plugin_config>
    <width>400</width>
    <z>0</z>
    <height>400</height>
    <location_x>1</location_x>
    <location_y>1</location_y>
  </plugin>
  <plugin>
    org.contikios.cooja.plugins.LogListener
    <plugin_config>
      <filter />
      <formatted_time />
      <coloring />
    </plugin_config>
    <width>1320</width>
    <z>3</z>
    <height>240</height>
    <location_x>400</location_x>
    <location_y>160</location_y>
  </plugin>
  <plugin>
    org.contikios.cooja.plugins.TimeLine
    <plugin_config>
      <mote>0</mote>
      <mote>1</mote>
      <mote>2</mote>
      <mote>3</mote>
      <mote>4</mote>
      <mote>5</mote>
      <mote>6</mote>
      <mote>7</mote>
      <mote>8</mote>
      <mote>9</mote>
      <mote>10</mote>
      <mote>11</mote>
      <mote>12</mote>
      <mote>13</mote>
      <mote>14</mote>
      <showRadioRXTX />
```

```
      <showRadioHW />
      <showLEDs />
      <zoomfactor>500.0</zoomfactor>
    </plugin_config>
    <width>1720</width>
    <z>2</z>
    <height>166</height>
    <location_x>0</location_x>
    <location_y>800</location_y>
  </plugin>
  <plugin>
    org.contikios.cooja.plugins.Notes
    <plugin_config>
      <notes>Enter notes here</notes>
      <decorations>true</decorations>
    </plugin_config>
    <width>1040</width>
    <z>1</z>
    <height>160</height>
    <location_x>680</location_x>
    <location_y>0</location_y>
  </plugin>
</simconf>
```

Código A.2 Contenido del fichero Escenario2.csc.

```
<?xml version="1.0" encoding="UTF-8"?>
<simconf>
  <simulation>
    <title>Escenario_2</title>
    <randomseed>123456</randomseed>
    <motedelay_us>1000000</motedelay_us>
    <radiomedium>
      org.contikios.cooja.radiomediums.UDGM
      <transmitting_range>50.0</transmitting_range>
      <interference_range>100.0</interference_range>
      <success_ratio_tx>1.0</success_ratio_tx>
      <success_ratio_rx>1.0</success_ratio_rx>
    </radiomedium>
    <events>
      <logoutput>40000</logoutput>
    </events>
    <motetype>
      org.contikios.cooja.mspmote.Z1MoteType
      <identifier>z1956630178</identifier>
      <description>Z1 Mote Type #1</description>
      <source>[CONTIKI_DIR]/examples/rpl-udp/udp-server.c</source>
      <commands>make -j$(CPUS) udp-server.z1 TARGET=z1</commands>
      <firmware>[CONTIKI_DIR]/examples/rpl-udp/build/z1/udp-server.z1</firmware
          >
      <moteinterface>org.contikios.cooja.interfaces.Position</moteinterface>
      <moteinterface>org.contikios.cooja.interfaces.RimeAddress</moteinterface>
      <moteinterface>org.contikios.cooja.interfaces.IPAddress</moteinterface>
      <moteinterface>org.contikios.cooja.interfaces.Mote2MoteRelations</
          moteinterface>
```

```
      <moteinterface>org.contikios.cooja.interfaces.MoteAttributes</
          moteinterface>
      <moteinterface>org.contikios.cooja.mspmote.interfaces.MspClock</
          moteinterface>
      <moteinterface>org.contikios.cooja.mspmote.interfaces.MspMoteID</
          moteinterface>
      <moteinterface>org.contikios.cooja.mspmote.interfaces.MspButton</
          moteinterface>
      <moteinterface>org.contikios.cooja.mspmote.interfaces.Msp802154Radio</
          moteinterface>
      <moteinterface>org.contikios.cooja.mspmote.interfaces.MspDefaultSerial</
          moteinterface>
      <moteinterface>org.contikios.cooja.mspmote.interfaces.MspLED</
          moteinterface>
      <moteinterface>org.contikios.cooja.mspmote.interfaces.MspDebugOutput</
          moteinterface>
    </motetype>
    <motetype>
      org.contikios.cooja.mspmote.Z1MoteType
      <identifier>z1649067866</identifier>
      <description>Z1 Mote Type #2</description>
      <source>[CONTIKI_DIR]/examples/rpl-udp/udp-client.c</source>
      <commands>make -j$(CPUS) udp-client.z1 TARGET=z1</commands>
      <firmware>[CONTIKI_DIR]/examples/rpl-udp/build/z1/udp-client.z1</firmware
          >
      <moteinterface>org.contikios.cooja.interfaces.Position</moteinterface>
      <moteinterface>org.contikios.cooja.interfaces.RimeAddress</moteinterface>
      <moteinterface>org.contikios.cooja.interfaces.IPAddress</moteinterface>
      <moteinterface>org.contikios.cooja.interfaces.Mote2MoteRelations</
          moteinterface>
      <moteinterface>org.contikios.cooja.interfaces.MoteAttributes</
          moteinterface>
      <moteinterface>org.contikios.cooja.mspmote.interfaces.MspClock</
          moteinterface>
      <moteinterface>org.contikios.cooja.mspmote.interfaces.MspMoteID</
          moteinterface>
      <moteinterface>org.contikios.cooja.mspmote.interfaces.MspButton</
          moteinterface>
      <moteinterface>org.contikios.cooja.mspmote.interfaces.Msp802154Radio</
          moteinterface>
      <moteinterface>org.contikios.cooja.mspmote.interfaces.MspDefaultSerial</
          moteinterface>
      <moteinterface>org.contikios.cooja.mspmote.interfaces.MspLED</
          moteinterface>
      <moteinterface>org.contikios.cooja.mspmote.interfaces.MspDebugOutput</
          moteinterface>
    </motetype>
    <mote>
      <interface_config>
        org.contikios.cooja.interfaces.Position
        <x>56.65666789510336</x>
        <y>45.72432020405961</y>
        <z>0.0</z>
      </interface_config>
      <interface_config>
        org.contikios.cooja.mspmote.interfaces.MspMoteID
        <id>1</id>
```

```
    </interface_config>
    <motetype_identifier>z1956630178</motetype_identifier>
  </mote>
  <mote>
    <interface_config>
      org.contikios.cooja.interfaces.Position
      <x>34.52958566048924</x>
      <y>49.36899264983572</y>
      <z>0.0</z>
    </interface_config>
    <interface_config>
      org.contikios.cooja.mspmote.interfaces.MspMoteID
      <id>2</id>
    </interface_config>
    <motetype_identifier>z1649067866</motetype_identifier>
  </mote>
  <mote>
    <interface_config>
      org.contikios.cooja.interfaces.Position
      <x>80.85696069589078</x>
      <y>7.5803776945479235</y>
      <z>0.0</z>
    </interface_config>
    <interface_config>
      org.contikios.cooja.mspmote.interfaces.MspMoteID
      <id>3</id>
    </interface_config>
    <motetype_identifier>z1649067866</motetype_identifier>
  </mote>
  <mote>
    <interface_config>
      org.contikios.cooja.interfaces.Position
      <x>59.76057676094925</x>
      <y>80.8086606958045</y>
      <z>0.0</z>
    </interface_config>
    <interface_config>
      org.contikios.cooja.mspmote.interfaces.MspMoteID
      <id>4</id>
    </interface_config>
    <motetype_identifier>z1649067866</motetype_identifier>
  </mote>
  <mote>
    <interface_config>
      org.contikios.cooja.interfaces.Position
      <x>62.13847511943567</x>
      <y>25.850342939401692</y>
      <z>0.0</z>
    </interface_config>
    <interface_config>
      org.contikios.cooja.mspmote.interfaces.MspMoteID
      <id>5</id>
    </interface_config>
    <motetype_identifier>z1649067866</motetype_identifier>
  </mote>
  <mote>
    <interface_config>
```

```
      org.contikios.cooja.interfaces.Position
      <x>74.27138867762703</x>
      <y>70.75660297148264</y>
      <z>0.0</z>
    </interface_config>
    <interface_config>
      org.contikios.cooja.mspmote.interfaces.MspMoteID
      <id>6</id>
    </interface_config>
    <motetype_identifier>z1649067866</motetype_identifier>
  </mote>
  <mote>
    <interface_config>
      org.contikios.cooja.interfaces.Position
      <x>35.15727284885902</x>
      <y>23.88551709295903</y>
      <z>0.0</z>
    </interface_config>
    <interface_config>
      org.contikios.cooja.mspmote.interfaces.MspMoteID
      <id>7</id>
    </interface_config>
    <motetype_identifier>z1649067866</motetype_identifier>
  </mote>
  <mote>
    <interface_config>
      org.contikios.cooja.interfaces.Position
      <x>40.7156741464228</x>
      <y>3.0407383477749517</y>
      <z>0.0</z>
    </interface_config>
    <interface_config>
      org.contikios.cooja.mspmote.interfaces.MspMoteID
      <id>8</id>
    </interface_config>
    <motetype_identifier>z1649067866</motetype_identifier>
  </mote>
  <mote>
    <interface_config>
      org.contikios.cooja.interfaces.Position
      <x>16.441014973566116</x>
      <y>97.82608782563848</y>
      <z>0.0</z>
    </interface_config>
    <interface_config>
      org.contikios.cooja.mspmote.interfaces.MspMoteID
      <id>9</id>
    </interface_config>
    <motetype_identifier>z1649067866</motetype_identifier>
  </mote>
  <mote>
    <interface_config>
      org.contikios.cooja.interfaces.Position
      <x>82.90145270379429</x>
      <y>46.90110747486293</y>
      <z>0.0</z>
    </interface_config>
```

```
    <interface_config>
      org.contikios.cooja.mspmote.interfaces.MspMoteID
      <id>10</id>
    </interface_config>
    <motetype_identifier>z1649067866</motetype_identifier>
  </mote>
  <mote>
    <interface_config>
      org.contikios.cooja.interfaces.Position
      <x>33.12684432179617</x>
      <y>29.92239776554577</y>
      <z>0.0</z>
    </interface_config>
    <interface_config>
      org.contikios.cooja.mspmote.interfaces.MspMoteID
      <id>11</id>
    </interface_config>
    <motetype_identifier>z1649067866</motetype_identifier>
  </mote>
  <mote>
    <interface_config>
      org.contikios.cooja.interfaces.Position
      <x>34.05662757684597</x>
      <y>67.58471279645613</y>
      <z>0.0</z>
    </interface_config>
    <interface_config>
      org.contikios.cooja.mspmote.interfaces.MspMoteID
      <id>12</id>
    </interface_config>
    <motetype_identifier>z1649067866</motetype_identifier>
  </mote>
  <mote>
    <interface_config>
      org.contikios.cooja.interfaces.Position
      <x>28.992984004122203</x>
      <y>81.72779172022115</y>
      <z>0.0</z>
    </interface_config>
    <interface_config>
      org.contikios.cooja.mspmote.interfaces.MspMoteID
      <id>13</id>
    </interface_config>
    <motetype_identifier>z1649067866</motetype_identifier>
  </mote>
  <mote>
    <interface_config>
      org.contikios.cooja.interfaces.Position
      <x>4.173231405836175</x>
      <y>66.17062268605882</y>
      <z>0.0</z>
    </interface_config>
    <interface_config>
      org.contikios.cooja.mspmote.interfaces.MspMoteID
      <id>14</id>
    </interface_config>
    <motetype_identifier>z1649067866</motetype_identifier>
```

```
    </mote>
    <mote>
      <interface_config>
        org.contikios.cooja.interfaces.Position
        <x>8.512032805041237</x>
        <y>24.393953932739176</y>
        <z>0.0</z>
      </interface_config>
      <interface_config>
        org.contikios.cooja.mspmote.interfaces.MspMoteID
        <id>15</id>
      </interface_config>
      <motetype_identifier>z1649067866</motetype_identifier>
    </mote>
  </simulation>
  <plugin>
    org.contikios.cooja.plugins.Visualizer
    <plugin_config>
      <moterelations>true</moterelations>
      <skin>org.contikios.cooja.plugins.skins.IDVisualizerSkin</skin>
      <skin>org.contikios.cooja.plugins.skins.GridVisualizerSkin</skin>
      <skin>org.contikios.cooja.plugins.skins.TrafficVisualizerSkin</skin>
      <skin>org.contikios.cooja.plugins.skins.UDGMVisualizerSkin</skin>
      <skin>org.contikios.cooja.plugins.skins.MoteTypeVisualizerSkin</skin>
      <viewport>3.3185028728402215 0.0 0.0 3.3185028728402215
          49.521205315268254 5.636573784626141</viewport>
    </plugin_config>
    <width>400</width>
    <z>0</z>
    <height>400</height>
    <location_x>1</location_x>
    <location_y>1</location_y>
  </plugin>
  <plugin>
    org.contikios.cooja.plugins.LogListener
    <plugin_config>
      <filter />
      <formatted_time />
      <coloring />
    </plugin_config>
    <width>1246</width>
    <z>3</z>
    <height>240</height>
    <location_x>400</location_x>
    <location_y>160</location_y>
  </plugin>
  <plugin>
    org.contikios.cooja.plugins.TimeLine
    <plugin_config>
      <mote>0</mote>
      <mote>1</mote>
      <mote>2</mote>
      <mote>3</mote>
      <mote>4</mote>
      <mote>5</mote>
      <mote>6</mote>
      <mote>7</mote>
```

```
      <mote>8</mote>
      <mote>9</mote>
      <mote>10</mote>
      <mote>11</mote>
      <mote>12</mote>
      <mote>13</mote>
      <mote>14</mote>
      <showRadioRXTX />
      <showRadioHW />
      <showLEDs />
      <zoomfactor>500.0</zoomfactor>
    </plugin_config>
    <width>1646</width>
    <z>2</z>
    <height>166</height>
    <location_x>0</location_x>
    <location_y>827</location_y>
  </plugin>
  <plugin>
    org.contikios.cooja.plugins.Notes
    <plugin_config>
      <notes>Enter notes here</notes>
      <decorations>true</decorations>
    </plugin_config>
    <width>966</width>
    <z>1</z>
    <height>160</height>
    <location_x>680</location_x>
    <location_y>0</location_y>
  </plugin>
</simconf>
```

Código A.3 Contenido del fichero Escenario3.csc.

```
<?xml version="1.0" encoding="UTF-8"?>
<simconf>
  <simulation>
    <title>Simu3</title>
    <randomseed>123456</randomseed>
    <motedelay_us>1000000</motedelay_us>
    <radiomedium>
      org.contikios.cooja.radiomediums.UDGM
      <transmitting_range>50.0</transmitting_range>
      <interference_range>100.0</interference_range>
      <success_ratio_tx>1.0</success_ratio_tx>
      <success_ratio_rx>1.0</success_ratio_rx>
    </radiomedium>
    <events>
      <logoutput>40000</logoutput>
    </events>
    <motetype>
      org.contikios.cooja.mspmote.Z1MoteType
      <identifier>z1891346872</identifier>
      <description>Z1 Mote Type #1</description>
      <source>[CONTIKI_DIR]/examples/rpl-udp/udp-server.c</source>
```

```
      <commands>make -j$(CPUS) udp-server.z1 TARGET=z1</commands>
      <firmware>[CONTIKI_DIR]/examples/rpl-udp/build/z1/udp-server.z1</firmware
          >
      <moteinterface>org.contikios.cooja.interfaces.Position</moteinterface>
      <moteinterface>org.contikios.cooja.interfaces.RimeAddress</moteinterface>
      <moteinterface>org.contikios.cooja.interfaces.IPAddress</moteinterface>
      <moteinterface>org.contikios.cooja.interfaces.Mote2MoteRelations</
          moteinterface>
      <moteinterface>org.contikios.cooja.interfaces.MoteAttributes</
          moteinterface>
      <moteinterface>org.contikios.cooja.mspmote.interfaces.MspClock</
          moteinterface>
      <moteinterface>org.contikios.cooja.mspmote.interfaces.MspMoteID</
          moteinterface>
      <moteinterface>org.contikios.cooja.mspmote.interfaces.MspButton</
          moteinterface>
      <moteinterface>org.contikios.cooja.mspmote.interfaces.Msp802154Radio</
          moteinterface>
      <moteinterface>org.contikios.cooja.mspmote.interfaces.MspDefaultSerial</
          moteinterface>
      <moteinterface>org.contikios.cooja.mspmote.interfaces.MspLED</
          moteinterface>
      <moteinterface>org.contikios.cooja.mspmote.interfaces.MspDebugOutput</
          moteinterface>
    </motetype>
    <motetype>
      org.contikios.cooja.mspmote.Z1MoteType
      <identifier>z1503335864</identifier>
      <description>Z1 Mote Type #2</description>
      <source>[CONTIKI_DIR]/examples/rpl-udp/udp-client.c</source>
      <commands>make -j$(CPUS) udp-client.z1 TARGET=z1</commands>
      <firmware>[CONTIKI_DIR]/examples/rpl-udp/build/z1/udp-client.z1</firmware
          >
      <moteinterface>org.contikios.cooja.interfaces.Position</moteinterface>
      <moteinterface>org.contikios.cooja.interfaces.RimeAddress</moteinterface>
      <moteinterface>org.contikios.cooja.interfaces.IPAddress</moteinterface>
      <moteinterface>org.contikios.cooja.interfaces.Mote2MoteRelations</
          moteinterface>
      <moteinterface>org.contikios.cooja.interfaces.MoteAttributes</
          moteinterface>
      <moteinterface>org.contikios.cooja.mspmote.interfaces.MspClock</
          moteinterface>
      <moteinterface>org.contikios.cooja.mspmote.interfaces.MspMoteID</
          moteinterface>
      <moteinterface>org.contikios.cooja.mspmote.interfaces.MspButton</
          moteinterface>
      <moteinterface>org.contikios.cooja.mspmote.interfaces.Msp802154Radio</
          moteinterface>
      <moteinterface>org.contikios.cooja.mspmote.interfaces.MspDefaultSerial</
          moteinterface>
      <moteinterface>org.contikios.cooja.mspmote.interfaces.MspLED</
          moteinterface>
      <moteinterface>org.contikios.cooja.mspmote.interfaces.MspDebugOutput</
          moteinterface>
    </motetype>
    <mote>
      <interface_config>
```

```
      org.contikios.cooja.interfaces.Position
      <x>39.20137152606162</x>
      <y>50.022623245169854</y>
      <z>0.0</z>
    </interface_config>
    <interface_config>
      org.contikios.cooja.mspmote.interfaces.MspMoteID
      <id>1</id>
    </interface_config>
    <motetype_identifier>z1891346872</motetype_identifier>
  </mote>
  <mote>
    <interface_config>
      org.contikios.cooja.interfaces.Position
      <x>48.07398304059429</x>
      <y>37.492162996866554</y>
      <z>0.0</z>
    </interface_config>
    <interface_config>
      org.contikios.cooja.mspmote.interfaces.MspMoteID
      <id>2</id>
    </interface_config>
    <motetype_identifier>z1503335864</motetype_identifier>
  </mote>
  <mote>
    <interface_config>
      org.contikios.cooja.interfaces.Position
      <x>84.51263451896884</x>
      <y>70.66718608090176</y>
      <z>0.0</z>
    </interface_config>
    <interface_config>
      org.contikios.cooja.mspmote.interfaces.MspMoteID
      <id>3</id>
    </interface_config>
    <motetype_identifier>z1503335864</motetype_identifier>
  </mote>
  <mote>
    <interface_config>
      org.contikios.cooja.interfaces.Position
      <x>37.60719621507002</x>
      <y>19.63063089386298</y>
      <z>0.0</z>
    </interface_config>
    <interface_config>
      org.contikios.cooja.mspmote.interfaces.MspMoteID
      <id>4</id>
    </interface_config>
    <motetype_identifier>z1503335864</motetype_identifier>
  </mote>
  <mote>
    <interface_config>
      org.contikios.cooja.interfaces.Position
      <x>26.800556897337124</x>
      <y>14.593967058140878</y>
      <z>0.0</z>
    </interface_config>
```

```
    <interface_config>
      org.contikios.cooja.mspmote.interfaces.MspMoteID
      <id>5</id>
    </interface_config>
    <motetype_identifier>z1503335864</motetype_identifier>
  </mote>
  <mote>
    <interface_config>
      org.contikios.cooja.interfaces.Position
      <x>84.8970632526681</x>
      <y>27.19481738723628</y>
      <z>0.0</z>
    </interface_config>
    <interface_config>
      org.contikios.cooja.mspmote.interfaces.MspMoteID
      <id>6</id>
    </interface_config>
    <motetype_identifier>z1503335864</motetype_identifier>
  </mote>
  <mote>
    <interface_config>
      org.contikios.cooja.interfaces.Position
      <x>37.90354368124873</x>
      <y>61.574869736557226</y>
      <z>0.0</z>
    </interface_config>
    <interface_config>
      org.contikios.cooja.mspmote.interfaces.MspMoteID
      <id>7</id>
    </interface_config>
    <motetype_identifier>z1503335864</motetype_identifier>
  </mote>
  <mote>
    <interface_config>
      org.contikios.cooja.interfaces.Position
      <x>57.512439971280195</x>
      <y>80.85373190217834</y>
      <z>0.0</z>
    </interface_config>
    <interface_config>
      org.contikios.cooja.mspmote.interfaces.MspMoteID
      <id>8</id>
    </interface_config>
    <motetype_identifier>z1503335864</motetype_identifier>
  </mote>
  <mote>
    <interface_config>
      org.contikios.cooja.interfaces.Position
      <x>0.39566125969311416</x>
      <y>75.4716556206694</y>
      <z>0.0</z>
    </interface_config>
    <interface_config>
      org.contikios.cooja.mspmote.interfaces.MspMoteID
      <id>9</id>
    </interface_config>
    <motetype_identifier>z1503335864</motetype_identifier>
```

```
    </mote>
    <mote>
      <interface_config>
        org.contikios.cooja.interfaces.Position
        <x>57.54703962562686</x>
        <y>71.61142936269667</y>
        <z>0.0</z>
      </interface_config>
      <interface_config>
        org.contikios.cooja.mspmote.interfaces.MspMoteID
        <id>10</id>
      </interface_config>
      <motetype_identifier>z1503335864</motetype_identifier>
    </mote>
    <mote>
      <interface_config>
        org.contikios.cooja.interfaces.Position
        <x>15.176884200412621</x>
        <y>2.4258443857216094</y>
        <z>0.0</z>
      </interface_config>
      <interface_config>
        org.contikios.cooja.mspmote.interfaces.MspMoteID
        <id>11</id>
      </interface_config>
      <motetype_identifier>z1503335864</motetype_identifier>
    </mote>
    <mote>
      <interface_config>
        org.contikios.cooja.interfaces.Position
        <x>19.79565120948852</x>
        <y>7.6895774110570585</y>
        <z>0.0</z>
      </interface_config>
      <interface_config>
        org.contikios.cooja.mspmote.interfaces.MspMoteID
        <id>12</id>
      </interface_config>
      <motetype_identifier>z1503335864</motetype_identifier>
    </mote>
    <mote>
      <interface_config>
        org.contikios.cooja.interfaces.Position
        <x>10.492491621577072</x>
        <y>56.96209055685636</y>
        <z>0.0</z>
      </interface_config>
      <interface_config>
        org.contikios.cooja.mspmote.interfaces.MspMoteID
        <id>13</id>
      </interface_config>
      <motetype_identifier>z1503335864</motetype_identifier>
    </mote>
    <mote>
      <interface_config>
        org.contikios.cooja.interfaces.Position
        <x>61.956663380130614</x>
```

```
      <y>58.31926731116107</y>
      <z>0.0</z>
    </interface_config>
    <interface_config>
      org.contikios.cooja.mspmote.interfaces.MspMoteID
      <id>14</id>
    </interface_config>
    <motetype_identifier>z1503335864</motetype_identifier>
  </mote>
  <mote>
    <interface_config>
      org.contikios.cooja.interfaces.Position
      <x>87.38190461453279</x>
      <y>64.44661495504171</y>
      <z>0.0</z>
    </interface_config>
    <interface_config>
      org.contikios.cooja.mspmote.interfaces.MspMoteID
      <id>15</id>
    </interface_config>
    <motetype_identifier>z1503335864</motetype_identifier>
  </mote>
</simulation>
<plugin>
  org.contikios.cooja.plugins.Visualizer
  <plugin_config>
    <moterelations>true</moterelations>
    <skin>org.contikios.cooja.plugins.skins.IDVisualizerSkin</skin>
    <skin>org.contikios.cooja.plugins.skins.GridVisualizerSkin</skin>
    <skin>org.contikios.cooja.plugins.skins.TrafficVisualizerSkin</skin>
    <skin>org.contikios.cooja.plugins.skins.UDGMVisualizerSkin</skin>
    <skin>org.contikios.cooja.plugins.skins.MoteTypeVisualizerSkin</skin>
    <viewport>4.010632754572825 0.0 0.0 4.010632754572825 17.978209594077907
        5.998101776401064</viewport>
  </plugin_config>
  <width>400</width>
  <z>0</z>
  <height>400</height>
  <location_x>1</location_x>
  <location_y>1</location_y>
</plugin>
<plugin>
  org.contikios.cooja.plugins.LogListener
  <plugin_config>
    <filter />
    <formatted_time />
    <coloring />
  </plugin_config>
  <width>1320</width>
  <z>3</z>
  <height>240</height>
  <location_x>400</location_x>
  <location_y>160</location_y>
</plugin>
<plugin>
  org.contikios.cooja.plugins.TimeLine
  <plugin_config>
```

```
      <mote>0</mote>
      <mote>1</mote>
      <mote>2</mote>
      <mote>3</mote>
      <mote>4</mote>
      <mote>5</mote>
      <mote>6</mote>
      <mote>7</mote>
      <mote>8</mote>
      <mote>9</mote>
      <mote>10</mote>
      <mote>11</mote>
      <mote>12</mote>
      <mote>13</mote>
      <mote>14</mote>
      <showRadioRXTX />
      <showRadioHW />
      <showLEDs />
      <zoomfactor>500.0</zoomfactor>
    </plugin_config>
    <width>1720</width>
    <z>2</z>
    <height>166</height>
    <location_x>0</location_x>
    <location_y>800</location_y>
  </plugin>
  <plugin>
    org.contikios.cooja.plugins.Notes
    <plugin_config>
      <notes>Enter notes here</notes>
      <decorations>true</decorations>
    </plugin_config>
    <width>1040</width>
    <z>1</z>
    <height>160</height>
    <location_x>680</location_x>
    <location_y>0</location_y>
  </plugin>
</simconf>
```

Apéndice B
Script de transformación de trazas

Código B.1 Contenido del fichero transformador.sh.

```
#!/bin/bash

# Directorios
DIR=$HOME/contiki-ng/tools/cooja/TFG;
DIR_TOOLCHAIN=$DIR/Output_toolchain/;
DIR_VAGRANT=$DIR/Output_vagrant/;
DIR_DOCKER=$DIR/Output_docker/;

if [[ "$1" =~ ^Escenario[123]$ ]]; then
  # Escenario a transformar, de cada método de instalación de Contiki:
  ESC_TOOLCHAIN=$(ls $DIR_TOOLCHAIN | grep $1);
  ESC_VAGRANT=$(ls $DIR_VAGRANT | grep $1);
  ESC_DOCKER=$(ls $DIR_DOCKER | grep $1);
  #Tablas con las rutas y nombres de los ficheros, seran usadas al escribir en
      el .csv de salida.
  DIRECTORIOS=($DIR_TOOLCHAIN $DIR_VAGRANT $DIR_DOCKER);
  ESCENARIOS=($ESC_TOOLCHAIN $ESC_VAGRANT $ESC_DOCKER);

  CSV="$DIR/Metricas_$1_$(date -u +"%Y-%m-%d").csv";
  #Crea la cabecera del .csv
  echo "timestamp,id,metrica,valor_medido,total,origen" > $CSV;
  #Rellena las filas del .csv con los contenidos de los logs de cada entorno si
       existe cada uno de los logs de simulaciones.
  for (( i=0; i<${#DIRECTORIOS[@]}; i++ )); do
    if [ -f "${DIRECTORIOS[$i]}/${ESCENARIOS[$i]}" ]; then

      awk -v OFS=',' -v ORIGEN="${ESCENARIOS[$i]}" '
        BEGIN{}
        {
          if(NF==11 && $4=="Energest" && $6!="---" && $6!="Total"){gsub("ID
              :","",$2);print (substr($1,0,length($1)-4),$2,"\""$6"\"",(substr
              ($8,0,length($8)-1)),$9,ORIGEN)};
          if(NF==12 && $4=="Energest" && $6!="---" && $6!="Total"){gsub("ID
              :","",$2);print (substr($1,0,length($1)-4),$2,"\""$6"_"$7"\"",(
              substr($9,0,length($9)-1)),$10,ORIGEN)}
        }
        END{}
```

```
      ' "${DIRECTORIOS[$i]}/${ESCENARIOS[$i]}" >> "$CSV"
    fi
  done
  #Filtro para que se guarde únicamente la información de las 3 primeras horas
      y la acbecera. Debe hacerse en dos pasos: Escribimos la cabecera y los
      datos en un fichero variable temp y despues sobreescribimos nuestro
      fichero destino con los contenidos de temp.
  echo "timestamp,id,metrica,valor_medido,total,origen" > temp;
  cat "$CSV" | grep -E '^[^3-9]*[0-5][0-9]:[0-5][0-9]' >> temp;
  cat temp > $CSV;
  rm temp;
else
  echo "Forma de utilizar el script (desde $DIR): ./transformador.sh EscenarioX
       , siendo X 1, 2 o 3.";
fi
```

Índice de Figuras

Índice de Códigos

Bibliografía

[1] Mariana Reyes Enríquez. Desarrollo de un entorno de virtualización y pruebas de RPL. Último acceso: 26 de junio de 2023, desde *https://biblus.us.es/bibing/proyectos/abreproy/94018/fichero/TFG-4018+REYES+HENRIQUEZ%2C+MARIANA.pdf*

[2] Elastic. Elasticsearch - Guía de introducción. Último acceso: 1 de mayo de 2023, desde *https://www.elastic.co/guide/en/elasticsearch/reference/current/elasticsearch-intro.html*

[3] Contiki-NG. Documentación de Contiki-NG. Último acceso: 1 de mayo de 2023, desde *https://docs.contiki-ng.org/en/develop/*

[4] Contiki-NG. Instrumenting Contiki-NG applications with energy usage estimation. Último acceso: 1 de mayo de 2023, desde *https://docs.contiki-ng.org/en/develop/doc/tutorials/Instrumenting-Contiki-NG-applications-with-energy-usage-estimation.html?highlight=simple-ennergest#interpreting-simple-energest-output*

[5] Free Software Foundation. GAWK: Effective AWK Programming. Último acceso: 1 de mayo de 2023, desde *https://www.gnu.org/software/gawk/manual/gawk.html#Getting-Started*

[6] Universidad de Murcia. Programación de la Shell. Awk. Último acceso: 1 de mayo de 2023, desde *https://www.um.es/innova/OCW/informatica-para-universitarios/ipu_docs/la_shell/awk.pdf*

[7] Elastic. Kibana - Guía de introducción. Último acceso: 1 de mayo de 2023, desde *https://www.elastic.co/guide/en/kibana/current/introduction.html*

[8] Elastic. Kibana Docker image. Último acceso: 1 de mayo de 2023, desde *https://www.elastic.co/guide/en/kibana/current/docker.html*

[9] HashiCorp. Vagrant by HashiCorp. Último acceso: 1 de mayo de 2023, desde *https://developer.hashicorp.com/vagrant/intro*

[10] HashiCorp. Vagrantfile - Vagrant by HashiCorp. Último acceso: 1 de mayo de 2023, desde *https://developer.hashicorp.com/vagrant/docs/vagrantfile*

[11] VirtualBox. Downloads - Oracle VM VirtualBox. Último acceso: 1 de mayo de 2023, desde *https://www.virtualbox.org/wiki/Downloads*

[12] Docker. (s.f.). Compose file. Último acceso: 1 de mayo de 2023, desde *https://docs.docker.com/compose/compose-file/*

- AWS: Servicio de nube de Amazon.
- AWK: Lenguaje de programación para procesamiento de texto y datos.
- Contiki-ng: Sistema operativo de código abierto para la internet de las cosas (IoT).
- csv: Formato de archivo de texto para datos tabulares.
- deb: Formato de archivo para paquetes de software utilizados en sistemas operativos basados en Debian.
- ELK: Siglas de Elasticsearch, Logstash y Kibana, un conjunto de herramientas para la visualización y análisis de datos.
- Ethernet: Tecnología de red utilizada para conectividad de computadoras.
- IoT: Siglas de Internet de las cosas.
- Linux: Sistema operativo de código abierto basado en Unix.
- README: Archivo de texto que contiene información sobre un proyecto de software.
- Ubuntu: Distribución de Linux basada en Debian.
- VirtualBox: Software de virtualización que permite ejecutar sistemas operativos adicionales en una máquina anfitriona.
- VMWare: Compañía que produce software de virtualización, incluyendo VMWare Workstation y VMWare ESXi.
- XFCE: Entorno de escritorio de código abierto para sistemas operativos tipo Unix.
- XML: Lenguaje de marcado utilizado para describir datos estructurados.