

# Universidad Autónoma de Madrid

## Escuela Politécnica Superior



Doble Grado en Ingeniería Informática y Matemáticas

## Trabajo de Fin de Grado

Interfaz web para la gestión de sondas de red de altas prestaciones

**Juan Sidrach de Cardona Mora**  
*juan.sidrach@gmail.com*

Junio 2014



# Interfaz web para la gestión de sondas de red de altas prestaciones

**Autor:** Juan Sidrach de Cardona Mora

**Tutor:** Dr. Sergio López-Buedo

High Performance Computing and Networking Research Group  
Dpto. de Tecnología Electrónica y de las Comunicaciones  
Escuela Politécnica Superior  
Universidad Autónoma de Madrid  
Junio 2014



# Agradecimientos

TODO: Agradecimientos.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus laoreet dolor at sodales porta. Morbi facilisis hendrerit lacus vel sollicitudin. Aenean eleifend urna metus, eget vestibulum libero dictum tincidunt. Curabitur quis ultrices lorem. Duis ultricies, eros eget condimentum pharetra, tellus eros lobortis nulla, vel mattis nibh dui et felis. Interdum et malesuada fames ac ante ipsum primis in faucibus. Nam non lorem et ligula condimentum molestie. Fusce quis dolor non metus suscipit commodo. Praesent vel pulvinar lectus. Nullam ac dui eget magna accumsan volutpat. Aliquam sed purus quis lorem dictum rutrum auctor eu enim. Pellentesque a urna ac ligula cursus lacinia. Aenean sodales justo massa, vel imperdiet justo imperdiet ut. Nulla euismod pulvinar arcu eu convallis. Vivamus a tempus nunc, et vulputate nulla.

Sed dapibus aliquam imperdiet. Vivamus est quam, fermentum vitae augue id, ultricies tincidunt massa. Praesent tincidunt ex sem, ut aliquet nulla imperdiet eu. Duis ac ultricies lorem. Aenean consequat ipsum nec arcu aliquam, sit amet interdum quam tempus. In justo odio, bibendum vel nulla nec, aliquet tristique justo. In vel metus ut libero suscipit ultricies.

Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos. Proin urna elit, iaculis id quam at, pretium laoreet ipsum. Phasellus ultricies faucibus ex et eleifend. Quisque facilisis erat dolor, ac rhoncus erat convallis et. Aliquam semper eleifend imperdiet. Sed eros ipsum, sagittis in pellentesque vel, vestibulum a augue. Duis sapien mauris, fringilla a tortor ut, sollicitudin volutpat nunc. Pellentesque vestibulum vel arcu in molestie. Nullam fermentum dolor luctus metus efficitur pulvinar. Pellentesque risus enim, tempus id ullamcorper in, maximus id nisl. Cras rhoncus consequat augue eu gravida. Ut efficitur mauris vitae orci dignissim sagittis. Suspendisse vitae massa eget nunc bibendum interdum.

Vestibulum quis turpis sed diam facilisis convallis. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos. Vivamus congue tellus nec lobortis feugiat. Nam hendrerit ullamcorper tempus. Proin maximus, lacus at tempor pellentesque, sem nisi facilisis lorem, sagittis tristique mauris dui at est. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos. Mauris pellentesque lobortis leo, ac dictum urna tempus id. Curabitur sed ante leo. Proin laoreet nisi nec dictum auctor. Mauris lacinia erat ut massa viverra, nec tempus metus elementum. Cras ut blandit justo, in pretium massa. In hac habitasse platea dictumst. Donec malesuada viverra quam, in ultricies libero. Phasellus finibus velit in sem tempus mattis at tristique ligula.

*“TODO: Cita relevante”* TODO: Autor de la cita



# Abstract

**Abstract** — TODO: Resumen en inglés, 250-500 palabras.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aliquam malesuada libero auctor sapien volutpat, sed fringilla enim tristique. Aliquam varius lorem in risus tempus egestas. Aenean accumsan elementum diam vel commodo. Nulla lectus sapien, finibus ac mauris non, efficitur venenatis felis. Donec at rutrum dolor, a lobortis arcu. In fermentum hendrerit bibendum. Phasellus eget arcu quam. Maecenas vulputate sapien eu dictum pulvinar. Suspendisse sit amet neque a turpis efficitur dapibus ut et turpis.

Vestibulum commodo faucibus tellus vitae consequat. Donec purus enim, hendrerit vitae feugiat sed, sagittis in tortor. Duis sed ex non ligula cursus dapibus. Etiam pellentesque suscipit dolor, vel facilisis est ornare sed. Nullam eleifend tellus non elementum efficitur. Donec semper felis ac porttitor ultricies. Vestibulum sodales justo nisl, in egestas lacus egestas nec. Fusce faucibus felis lacus, sit amet placerat justo porta vitae. Nullam volutpat viverra lorem quis euismod. Duis felis erat, dictum et sem vitae, fringilla ultrices dui. Morbi mattis arcu at orci accumsan facilisis. Aenean tortor velit, hendrerit id vulputate ac, sagittis nec libero. Donec elementum dolor orci, a mattis augue lobortis nec. Suspendisse vulputate, diam vel accumsan pellentesque, ex purus volutpat ipsum, vel luctus urna sem non turpis. Donec vitae molestie odio.

Donec lobortis, eros non sodales dapibus, ex eros sollicitudin tortor, ut vulputate massa nibh sit amet ipsum. Sed a lectus eu diam pretium vestibulum. Pellentesque finibus, felis ac finibus vulputate, libero mauris placerat nulla, ut vestibulum ante metus ut neque. Aliquam tempus tortor ac mauris pulvinar iaculis. Vivamus pretium id libero sed tempus. Donec tincidunt turpis tempor vehicula egestas. Vestibulum elementum, urna non tincidunt tempus, risus ipsum posuere felis, ac suscipit diam nunc et neque. Vestibulum faucibus leo vel nibh tempor tincidunt. Nullam nunc augue, aliquet in congue nec, gravida at risus. Proin semper iaculis nisi vitae imperdiet. Suspendisse sed risus feugiat, dapibus sapien quis, pulvinar turpis.

Maecenas convallis aliquet euismod. Donec sollicitudin ligula nec lorem dignissim, sit amet finibus felis mollis. Fusce eget sapien eu sapien blandit congue quis a odio. Fusce accumsan condimentum dapibus. Aliquam eu ante porttitor nulla pellentesque feugiat pharetra nec mauris. Ut tincidunt urna vitae ligula mattis malesuada. Interdum et malesuada fames ac ante ipsum primis in faucibus. Integer pretium tincidunt nisi, in pulvinar velit dapibus et.

**Key words** — TODO: Palabras clave en inglés, separadas por coma.





# Resumen

**Resumen** — TODO: Resumen en español, 250-500 palabras.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aliquam malesuada libero auctor sapien volutpat, sed fringilla enim tristique. Aliquam varius lorem in risus tempus egestas. Aenean accumsan elementum diam vel commodo. Nulla lectus sapien, finibus ac mauris non, efficitur venenatis felis. Donec at rutrum dolor, a lobortis arcu. In fermentum hendrerit bibendum. Phasellus eget arcu quam. Maecenas vulputate sapien eu dictum pulvinar. Suspendisse sit amet neque a turpis efficitur dapibus ut et turpis.

Vestibulum commodo faucibus tellus vitae consequat. Donec purus enim, hendrerit vitae feugiat sed, sagittis in tortor. Duis sed ex non ligula cursus dapibus. Etiam pellentesque suscipit dolor, vel facilisis est ornare sed. Nullam eleifend tellus non elementum efficitur. Donec semper felis ac porttitor ultricies. Vestibulum sodales justo nisl, in egestas lacus egestas nec. Fusce faucibus felis lacus, sit amet placerat justo porta vitae. Nullam volutpat viverra lorem quis euismod. Duis felis erat, dictum et sem vitae, fringilla ultrices dui. Morbi mattis arcu at orci accumsan facilisis. Aenean tortor velit, hendrerit id vulputate ac, sagittis nec libero. Donec elementum dolor orci, a mattis augue lobortis nec. Suspendisse vulputate, diam vel accumsan pellentesque, ex purus volutpat ipsum, vel luctus urna sem non turpis. Donec vitae molestie odio.

Donec lobortis, eros non sodales dapibus, ex eros sollicitudin tortor, ut vulputate massa nibh sit amet ipsum. Sed a lectus eu diam pretium vestibulum. Pellentesque finibus, felis ac finibus vulputate, libero mauris placerat nulla, ut vestibulum ante metus ut neque. Aliquam tempus tortor ac mauris pulvinar iaculis. Vivamus pretium id libero sed tempus. Donec tincidunt turpis tempor vehicula egestas. Vestibulum elementum, urna non tincidunt tempus, risus ipsum posuere felis, ac suscipit diam nunc et neque. Vestibulum faucibus leo vel nibh tempor tincidunt. Nullam nunc augue, aliquet in congue nec, gravida at risus. Proin semper iaculis nisi vitae imperdiet. Suspendisse sed risus feugiat, dapibus sapien quis, pulvinar turpis.

Maecenas convallis aliquet euismod. Donec sollicitudin ligula nec lorem dignissim, sit amet finibus felis mollis. Fusce eget sapien eu sapien blandit congue quis a odio. Fusce accumsan condimentum dapibus. Aliquam eu ante porttitor nulla pellentesque feugiat pharetra nec mauris. Ut tincidunt urna vitae ligula mattis malesuada. Interdum et malesuada fames ac ante ipsum primis in faucibus. Integer pretium tincidunt nisi, in pulvinar velit dapibus et.

**Palabras clave** — TODO: Palabras clave en español, separadas por coma.



# Glosario

- framework** Entorno software que proporciona una funcionalidad base para facilitar la organización y desarrollo de aplicaciones. 29, 30
- proxy** Servidor que sirve de intermediario entre las peticiones de recursos que realiza un cliente a otro servidor. 30
- Servicio Web** Conjunto de métodos remotos accesibles a través de la red. 21, 22, 25–28, 31, 32
- simple** Formato de traza que acepta la FPGA utilizada. 8, 22, 57–59, 63, 64
- traza** Archivo que contiene paquetes de red capturados. 4, 8, 9, 22, 27, 33–46, 52–54, 56–66



# Acrónimos

- API** Application Programming Interface (métodos públicos de una aplicación). 27, 31, 32
- FPGA** Field-Programmable Gate Array. 3, 7–9, 21, 22, 25–28, 31–45, 47, 49, 51–54
- HTTP** Hypertext Transfer Protocol. 31
- IFG** InterFrame Gap (pausa temporal entre paquetes). 8, 39, 41, 53
- JSON** JavaScript Object Notation. 31
- pcap** Packet capture (formato de traza, utilizado por programas como *Wireshark* y *tcpdump*). 8, 22, 57, 59–61, 63–65
- PHP** PHP Hypertext Pre-processor. 30
- RAID** Redundant Array of Independent Disks. 9, 27, 48, 49, 55
- URL** Uniform Resource Locator. 31



# Índice general

<b>1. Introducción</b>	<b>1</b>
1.1. Ámbito . . . . .	1
1.2. Motivación . . . . .	1
1.3. Objetivos . . . . .	1
1.4. Estructura del documento . . . . .	2
<b>2. Estado del arte</b>	<b>3</b>
2.1. Sistemas de captura y/o reproducción de tráfico de red . . . . .	3
2.2. FPGA HPCN . . . . .	3
2.3. Sistemas de gestión y monitorización . . . . .	4
2.4. Conclusiones . . . . .	4
<b>3. Definición del proyecto</b>	<b>5</b>
3.1. Objetivos . . . . .	5
3.2. Alcance . . . . .	5
3.3. Metodología . . . . .	5
3.4. Herramientas . . . . .	6
<b>4. Requisitos</b>	<b>7</b>
4.1. Requisitos Funcionales . . . . .	7
4.2. Requisitos No Funcionales . . . . .	9
<b>5. Diseño</b>	<b>11</b>
5.1. Arquitectura . . . . .	11
5.2. Módulos . . . . .	11
5.2.1. Gestión . . . . .	11
5.2.2. Capturas . . . . .	12
5.2.3. Estado/Estadísticas . . . . .	12
5.3. Servicio Web FPGA . . . . .	12
5.4. Interfaz Gráfica . . . . .	12
<b>6. Implementación</b>	<b>13</b>
6.1. Back-End . . . . .	13
6.2. Front-End . . . . .	13
6.2.1. Localización . . . . .	13
6.2.2. Temas . . . . .	14

<b>7. Pruebas</b>	<b>15</b>
7.1. Pruebas de verificación . . . . .	15
7.2. Pruebas de validación . . . . .	15
<b>8. Mantenimiento</b>	<b>17</b>
<b>9. Conclusiones</b>	<b>19</b>
<b>10. Líneas de trabajo futuras</b>	<b>21</b>
<b>Bibliografía</b>	<b>23</b>
<b>Apéndices</b>	<b>25</b>
<b>A. Manual de Usuario</b>	<b>27</b>
A.1. Instalación del Servicio Web FPGA . . . . .	27
A.2. Configuración del Servicio Web FPGA . . . . .	28
A.3. Instalación de la interfaz web . . . . .	29
A.4. Configuración de la interfaz web . . . . .	30
A.5. Uso de la aplicación . . . . .	30
A.6. Solución de problemas . . . . .	30
<b>B. Framework MVC propio</b>	<b>31</b>
B.1. Modelo Vista Controlador . . . . .	31
B.2. Manejo de rutas . . . . .	31
B.3. Redireccionamiento de peticiones a servicio web . . . . .	32
B.4. Manejo de dependencias . . . . .	32
B.5. Configuración . . . . .	32
B.6. Registro de eventos . . . . .	32
B.7. Conclusiones . . . . .	32
<b>C. API del Servicio Web FPGA</b>	<b>33</b>
C.1. Métodos de Gestión . . . . .	34
C.1.1. POST /system/reboot . . . . .	34
C.1.2. POST /player/init . . . . .	35
C.1.3. POST /recorder/init . . . . .	37
C.1.4. POST /player/install . . . . .	38
C.1.5. POST /recorder/install . . . . .	40
C.1.6. POST /player/start/:capturename/:mask/:ifg . . . . .	41
C.1.7. POST /player/start/loop/:capturename/:mask/:ifg . . . . .	43
C.1.8. POST /recorder/start/:capturename/:port/:bytes . . . . .	45
C.1.9. POST /player/stop . . . . .	47
C.1.10. POST /recorder/stop . . . . .	48
C.1.11. DELETE /storage/raid . . . . .	50
C.2. Métodos de Estadísticas . . . . .	51
C.2.1. GET /info/ping . . . . .	51
C.2.2. GET /info/delay . . . . .	52



C.2.3.	GET /info/status . . . . .	53
C.2.4.	GET /storage/stats . . . . .	57
C.3.	Métodos de Trazas . . . . .	58
C.3.1.	GET /captures/all . . . . .	58
C.3.2.	GET /captures/simple . . . . .	60
C.3.3.	GET /captures/pcap . . . . .	61
C.3.4.	GET /captures/path . . . . .	62
C.3.5.	PUT /captures/rename/:oldname/:newname . . . . .	63
C.3.6.	PUT /captures/simple/pcap/:name/:convertedname . . . . .	65
C.3.7.	PUT /captures/pcap/simple/:name/:convertedname . . . . .	66
C.3.8.	DELETE /captures/delete/:name . . . . .	68



# Índice de tablas

A.1. Variables de configuración del Servicio Web Field-Programmable Gate Array (FPGA) . . . . .	29
C.1. Parámetros de <i>/system/reboot</i> . . . . .	34
C.2. Salida de <i>/system/reboot</i> asociada al código 200 . . . . .	35
C.3. Salida de <i>/system/reboot</i> asociada al código 412 . . . . .	35
C.4. Parámetros de <i>/player/init</i> . . . . .	36
C.5. Salida de <i>/player/init</i> asociada al código 200 . . . . .	36
C.6. Salida de <i>/player/init</i> asociada al código 412 . . . . .	36
C.7. Parámetros de <i>/recorder/init</i> . . . . .	37
C.8. Salida de <i>/recorder/init</i> asociada al código 200 . . . . .	37
C.9. Salida de <i>/recorder/init</i> asociada al código 412 . . . . .	38
C.10. Parámetros de <i>/player/install</i> . . . . .	38
C.11. Salida de <i>/player/install</i> asociada al código 200 . . . . .	39
C.12. Salida de <i>/player/install</i> asociada al código 412 . . . . .	39
C.13. Parámetros de <i>/recorder/install</i> . . . . .	40
C.14. Salida de <i>/recorder/install</i> asociada al código 200 . . . . .	40
C.15. Salida de <i>/recorder/install</i> asociada al código 412 . . . . .	41
C.16. Parámetros de <i>/player/start</i> . . . . .	41
C.17. Salida de <i>/player/start</i> asociada al código 200 . . . . .	42
C.18. Salida de <i>/player/start</i> asociada al código 400 . . . . .	42
C.19. Salida de <i>/player/start</i> asociada al código 412 . . . . .	43
C.20. Parámetros de <i>/player/start/loop</i> . . . . .	43
C.21. Salida de <i>/player/start/loop</i> asociada al código 200 . . . . .	44
C.22. Salida de <i>/player/start/loop</i> asociada al código 400 . . . . .	44
C.23. Salida de <i>/player/start/loop</i> asociada al código 412 . . . . .	45
C.24. Parámetros de <i>/recorder/start</i> . . . . .	45
C.25. Salida de <i>/recorder/start</i> asociada al código 200 . . . . .	46
C.26. Salida de <i>/recorder/start</i> asociada al código 400 . . . . .	46
C.27. Salida de <i>/recorder/start</i> asociada al código 412 . . . . .	47
C.28. Parámetros de <i>/player/stop</i> . . . . .	47
C.29. Salida de <i>/player/stop</i> asociada al código 200 . . . . .	47
C.30. Salida de <i>/player/stop</i> asociada al código 412 . . . . .	48
C.31. Parámetros de <i>/recorder/stop</i> . . . . .	49
C.32. Salida de <i>/recorder/stop</i> asociada al código 200 . . . . .	49
C.33. Salida de <i>/recorder/stop</i> asociada al código 412 . . . . .	49
C.34. Parámetros de <i>/storage/raid</i> . . . . .	50

C.35.Salida de <i>/storage/raid</i> asociada al código 200 . . . . .	50
C.36.Salida de <i>/storage/raid</i> asociada al código 412 . . . . .	51
C.37.Salida de <i>/info/ping</i> asociada al código 200 . . . . .	52
C.38.Parámetros de <i>/info/delay</i> . . . . .	52
C.39.Salida de <i>/info/delay</i> asociada al código 200 . . . . .	52
C.40.Salida de <i>/info/status</i> asociada al código 200 - <i>hugepages_off</i> . . . . .	53
C.41.Salida de <i>/info/status</i> asociada al código 200 - <i>init_off</i> . . . . .	54
C.42.Salida de <i>/info/status</i> asociada al código 200 - <i>mount_off</i> . . . . .	54
C.43.Salida de <i>/info/status</i> asociada al código 200 - <i>player_ready</i> . . . . .	54
C.44.Salida de <i>/info/status</i> asociada al código 200 - <i>recorder_ready</i> . . . . .	55
C.45.Salida de <i>/info/status</i> asociada al código 200 - <i>playing</i> . . . . .	55
C.46.Salida de <i>/info/status</i> asociada al código 200 - <i>recording</i> . . . . .	56
C.47.Salida de <i>/storage/stats</i> asociada al código 200 . . . . .	57
C.48.Salida de <i>/captures/all</i> asociada al código 200 . . . . .	59
C.49.Salida de <i>/captures/simple</i> asociada al código 200 . . . . .	61
C.50.Salida de <i>/captures/pcap</i> asociada al código 200 . . . . .	62
C.51.Salida de <i>/captures/path</i> asociada al código 200 . . . . .	63
C.52.Parámetros de <i>/captures/rename</i> . . . . .	63
C.53.Salida de <i>/captures/rename</i> asociada al código 200 . . . . .	64
C.54.Salida de <i>/captures/rename</i> asociada al código 400 . . . . .	64
C.55.Parámetros de <i>/captures/simple/pcap</i> . . . . .	65
C.56.Salida de <i>/captures/simple/pcap</i> asociada al código 200 . . . . .	65
C.57.Salida de <i>/captures/simple/pcap</i> asociada al código 400 . . . . .	66
C.58.Parámetros de <i>/captures/pcap/simple</i> . . . . .	66
C.59.Salida de <i>/captures/pcap/simple</i> asociada al código 200 . . . . .	67
C.60.Salida de <i>/captures/pcap/simple</i> asociada al código 400 . . . . .	67
C.61.Parámetros de <i>/captures/delete</i> . . . . .	68
C.62.Salida de <i>/captures/delete</i> asociada al código 200 . . . . .	68
C.63.Salida de <i>/captures/delete</i> asociada al código 400 . . . . .	69

# Índice de figuras

C.1. Documentación web de la Application Programming Interface (métodos públicos de una aplicación) (API) del Servicio Web FPGA. . . . .	34
--	----



# 1

## Introducción

TODO: Introducción del trabajo

### 1.1. Ámbito

TODO: Ámbito del trabajo

### 1.2. Motivación

TODO: Motivación del trabajo

### 1.3. Objetivos

TODO: Objetivos del trabajo

## 1.4. Estructura del documento

TODO: Descripción de la estructura del documento



# 2

## Estado del arte

TODO: [Introducción]

### 2.1. Sistemas de captura y/o reproducción de tráfico de red

TODO: Sistemas de captura y/o reproducción de tráfico de red

### 2.2. FPGA HPCN

TODO: Cambiar organización Los posibles estados de la FPGA son:

- No programada
- Programada para reproducir

- Programada para capturar
- Montada para reproducir
- Montada para capturar
- Reproduciendo una traza
- Capturando tráfico

### 2.3. Sistemas de gestión y monitorización

TODO: Sistemas de gestión y monitorización

### 2.4. Conclusiones

TODO: Conclusiones

# 3

## Definición del proyecto

TODO: [Introducción]

### 3.1. Objetivos

TODO: Objetivos del proyecto

### 3.2. Alcance

TODO: Alcance del proyecto

### 3.3. Metodología

TODO: Metodología del proyecto Sprints

## 3.4. Herramientas

TODO: Metodologia del proyecto. Dividir en Back-End y Front-End Lista de herramientas, una subsección por cada una

# 4

## Requisitos

En este capítulo se enumeran los requisitos de la aplicación a desarrollar. Para la elaboración de esta lista de requisitos se ha realizado un análisis sobre el problema planteado: diseñar un servicio que permita gestionar y monitorizar una FPGA que captura y reproduce tráfico de red. Este análisis se ha realizado principalmente mediante la consulta directa con los potenciales usuarios de la aplicación y la evaluación del estado del arte.

Se han agrupado los requisitos en dos clases: funcionales y no funcionales. Los primeros describen el comportamiento que tendrá la aplicación, y los segundos los atributos de calidad y restricciones de la misma.

### 4.1. Requisitos Funcionales

Los requisitos funcionales que deberá cumplir la aplicación desarrollada son los siguientes:

- RF. 1** *Se podrá conocer el estado actual de la FPGA entre los posibles estados descritos en 2.2.*
- RF. 2** *Se podrá configurar la FPGA para captura de tráfico de red.*
- RF. 3** *Una vez configurada la FPGA para captura de tráfico de red, se le podrá ordenar que capture tráfico de red desde un puerto específico de la FPGA. Este tráfico se irá guardando en una traza en formato simple, hasta llegar a un tamaño decidido por el usuario.*
- RF. 4** *Si existe una captura en curso, el se podrá parar dicha captura, borrándose la traza asociada a la captura.*
- RF. 5** *Si existe una captura en curso, se podrán conocer los parámetros con los que se inició dicha captura, así como el tiempo que ha transcurrido desde el inicio y cuántos bytes se ha capturado hasta el momento.*
- RF. 6** *Se podrá configurar la FPGA para la reproducción de una traza.*
- RF. 7** *Una vez configurada la FPGA para la reproducción de una traza, se le podrá ordenar que reproduzca una traza concreta en formato simple. La reproducción se realizará con una una serie de parámetros dados por el usuario: máscara de puertos a los que dirigir la reproducción, InterFrame Gap (pausa temporal entre paquetes) (IFG) asociado y reproducir en bucle o solo una vez.*
- RF. 8** *Si existe una reproducción de traza en curso, se podrá parar dicha reproducción.*
- RF. 9** *Si existe una reproducción de traza en curso, se podrán conocer los parámetros con los que se inició dicha reproducción, así como el tiempo que ha transcurrido desde el inicio y cuántos paquetes se han reproducido hasta el momento.*
- RF. 10** *Se podrá configurar y consultar en qué directorio se almacenan las trazas.*
- RF. 11** *Se podrá conocer la lista de trazas existentes, así como su tamaño, fecha y tipo (simple o Packet capture (formato de traza, utilizado por programas como Wireshark y tcpdump) (pcap)).*
- RF. 12** *Una traza en formato simple podrá ser convertida a formato pcap.*
- RF. 13** *Una traza en formato pcap podrá ser convertida a formato simple.*
- RF. 14** *Una traza podrá ser renombrada.*

- RF. 15** *Una traza podrá ser borrada.*
- RF. 16** *Se podrán conocer el espacio total y el espacio ocupado del sistema de archivos que contiene las trazas.*
- RF. 17** *Si el sistema de archivos que contiene las trazas es un Redundant Array of Independent Disks (RAID), se podrá conocer la velocidad de escritura global del RAID, así como la de cada disco que lo compone.*
- RF. 18** *Si el sistema de archivos que contiene las trazas es un RAID, se podrá formatear y recrear el RAID.*

## 4.2. Requisitos No Funcionales

Los requisitos no funcionales que deberá cumplir la aplicación desarrollada son los siguientes:

- RNF. 1** *La funcionalidad descrita en 4.1 será accesible al usuario mediante una interfaz gráfica.*
- RNF. 2** *Se podrán seleccionar dos idiomas para la interfaz gráfica: inglés y español.*
- RNF. 3** *Se podrán seleccionar distintos temas (aspectos visuales) para la interfaz gráfica.*
- RNF. 4** *La interfaz gráfica será una web adaptativa, de forma que se pueda visualizar en distintas resoluciones de pantalla, como las de ordenadores y móviles.*
- RNF. 5** *La interfaz gráfica estará disponible aun cuando haya algún fallo en el servidor que aloja la FPGA, e informará del error.*





# 5

## Diseño

TODO: [Introducción]

### 5.1. Arquitectura

TODO: Arquitectura Front-End/Back-End

### 5.2. Módulos

TODO: [Introducción]

#### 5.2.1. Gestión

TODO: Gestión Capturador,Reproductor

### 5.2.2. Capturas

TODO: Capturas Detección, Conversión, Renombrado, Borrado

### 5.2.3. Estado/Estadísticas

TODO: Estado/Estadísticas Velocidad, Estado, RAID

## 5.3. Servicio Web FPGA

TODO: REST-like, asíncrono

## 5.4. Interfaz Gráfica

TODO: Interfaz Gráfica Diseño adaptativo, localización Maquetas

# 6

## Implementación

TODO: [Introducción]

### 6.1. Back-End

TODO: Back-End [Introducción] io.js,express,supervisor,apiDoc,npm API REST,mensajes json Servicio

### 6.2. Front-End

TODO: Front-End [Introducción] - MVC propio,Bootstrap,jQuery,gettext,phpDocumentor

#### 6.2.1. Localización

TODO: Localización

### 6.2.2. Temas

TODO: Temas

# 7

## Pruebas

TODO: [Introducción]

### 7.1. Pruebas de verificación

TODO: Pruebas de verificación

### 7.2. Pruebas de validación

TODO: Pruebas de validación



# 8

## Mantenimiento

TODO: [Introducción] Open Source/GitHub Issues/Pull Requests





# 9

## Conclusiones

TODO: Conclusiones sobre el trabajo realizado



# 10

## Líneas de trabajo futuras

En el contexto de este Trabajo de Fin de Grado, se ha desarrollado una interfaz web para el manejo de sondas red de altas prestaciones. Gracias al trabajo realizado, se han identificado áreas de interés que podrían ser consideradas con el objetivo de mejorar y ampliar la aplicación en el futuro, y que no han podido ser abordadas en el mismo por la limitación del tiempo disponible. Se describen a continuación algunas de estas posibles mejoras.

### **Estandarización del Servicio Web**

La aplicación implementada gestiona un dispositivo concreto de captura y reproducción de tráfico de red. Aunque algunos componentes son específicos para la FPGA utilizada, también se han desarrollado componentes más genéricos como los de gestión de capturas o almacenamiento. Es por ello que una posible área de mejora sería estandarizar el Servicio Web, documentando los métodos mínimos necesarios para el funcionamiento del servicio de forma genérica. Esto facilitaría la tarea de añadir una interfaz gráfica a otros dispositivos de reproducción y captura de tráfico de red.

### Soporte de subtipos de trazas pcap adicionales

El sistema de gestión de trazas actual soporta los formatos simple y pcap. Las trazas en formato pcap tienen sin embargo subtipos, cada uno con características distintas que en el sistema actual se descartan. En línea con la estandarización del Servicio Web, poder distinguir entre los distintos subtipos de trazas pcap facilitaría obtener información adicional propia de cada subformato, permitiendo además clasificar y convertir entre cada uno de los subtipos.

### Registro de estadísticas adicionales

El sistema actual consta de un módulo que proporciona estadísticas en tiempo real sobre el estado de la FPGA y de los distintos componentes que intervienen en el proceso de captura y reproducción. Estos datos no se almacenan de forma persistente una vez obtenidos. Una opción sería guardar en una base de datos estas estadísticas y parámetros de utilización de la FPGA. Esto permitiría un análisis posterior de estas estadísticas almacenadas para sacar conclusiones sobre distintos parámetros como el rendimiento o las operaciones más frecuentes.

### Localización en otros idiomas

El trabajo base para dar soporte a diferentes idiomas en la interfaz gráfica ya ha sido realizado, y actualmente la aplicación está disponible en español e inglés. Por tanto, es posible añadir idiomas adicionales a la interfaz traduciendo las distintas cadenas de texto a otros idiomas, sin ser necesario esfuerzo adicional a nivel de diseño e implementación.

### Módulo de autenticación

Dado que la interfaz web está pensada para ser utilizada en redes internas, sin acceso desde el exterior, no se ha planteado implementar un módulo de autenticación que impida a usuarios no autorizados el acceso a la aplicación. Desarrollar este módulo de autenticación haría posible instalar el servidor en una dirección pública, sin ceder por ello el control del sistema a una persona ajena. Esto permitiría que un usuario autorizado pudiera utilizar la interfaz desde cualquier punto con conexión a internet.

# Bibliografía

- [1] Debian distribution. [Online]. Available: <http://www.debian.org>
- [2] The Apache HTTP Server Project. [Online]. Available: <http://httpd.apache.org/>
- [3] HTTP method definitions. World Wide Web Consortium (W3C). [Online]. Available: <http://www.w3.org/Protocols/rfc2616/rfc2616-sec9.html>
- [4] HTTP code definitions. World Wide Web Consortium (W3C). [Online]. Available: <http://www.w3.org/Protocols/rfc2616/rfc2616-sec10.html>
- [5] JSON format. [Online]. Available: <http://json.org/>



# Apéndices







# Manual de Usuario

Este manual pretende ser una guía en la instalación, configuración y uso de la interfaz web para la gestión de sondas de red de altas prestaciones.

## A.1. Instalación del Servicio Web FPGA

### Requisitos

El servidor que aloje el Servicio Web FPGA debe cumplir los siguientes requisitos:

- La FPGA para capturar/reproducir tráfico debe estar conectada.
- El sistema operativo debe estar basado en una distribución *Debian* [1] y tener una arquitectura de 64 bits.
- La opción por defecto en el gestor de arranque debe ser iniciar con la opción *HugePages* activa.

- El usuario *root* debe existir.

### Instalación

Para instalar el Servicio Web FPGA, compruebe que se cumplan todos los requisitos y siga los siguientes pasos:

1. Descargue el código fuente del repositorio del proyecto ([github.com/JSidrach/NetWatcher](https://github.com/JSidrach/NetWatcher)). La instalación se realiza de forma remota, así que no es necesario descargárselo en el propio servidor, aunque sí en un entorno con terminal.
2. Descomprima el archivo *.zip*.
3. Edite el archivo `./fpga-api/scripts/update_server.sh`, estableciendo los parámetros `SERVER_IP` y `SERVER_PATH` como la dirección del servidor remoto que alojará el Servicio Web FPGA y la ruta donde guardar el código, respectivamente.
4. Sitúese en la carpeta `./fpga-api/`.
5. Despliegue el servidor ejecutando el siguiente comando:  

```
./scripts/update_server.sh.
```
6. Inicie sesión en el servidor remoto.
7. Ponga en marcha el Servicio Web ejecutando el siguiente comando:  

```
sudo service fpga-api start
```
8. Compruebe que el servidor está activo con el siguiente comando:  

```
sudo service fpga-api status
```

### A.2. Configuración del Servicio Web FPGA

Para configurar el Servicio Web FPGA, inicie sesión en el servidor en el que se instaló este servicio. Los distintos parámetros de configuración vienen recogidos en el archivo `config.js`, dentro de la carpeta raíz del servicio (el contenido de la variable `SERVER_PATH`, establecida en la instalación). Edite las distintas variables de este archivo (explicadas en la Tabla A.1) para configurar el servicio. No es necesario reiniciar el servicio para que los cambios en el archivo `config.js` se reflejen en el servidor.

Variable	Tipo	Descripción
BASE_PREFIX	Cadena de texto	Prefijo base de la API
PORT	Número entero	Puerto del servicio
MAX_DELAY	Número entero	Retraso máximo entre el <i>timestamp</i> de las peticiones y el <i>timestamp</i> del servidor. Si es menor o igual que 0, no se descartará ninguna petición basándose en el <i>timestamp</i>
IMPACT_BIN	Cadena de texto	Ruta al ejecutable <b>impact</b> de <i>Xilinx</i>
CAPTURES_DIR	Cadena de texto	Directorio donde se guardarán las trazas (debe acabar en /)
RAID	Booleano	Bandera que indica si el RAID está activo o no. Establezca esta variable como <i>true</i> sólo si CAPTURES_DIR está sobre un RAID y las variables RAID_DEV y RAID_DISKS están asignadas.
RAID_DEV	Cadena de texto	Ruta al RAID
RAID_DISKS	Array de cadenas	Discos físicos del RAID (por ejemplo: /dev/sdc, /dev/sdd, etc.)

Tabla A.1: Variables de configuración del Servicio Web FPGA

### A.3. Instalación de la interfaz web

#### Requisitos

El servidor que aloje la interfaz web debe cumplir los siguientes requisitos:

- *Apache httpd* [2] debe estar instalado.
- La dirección del Servicio Web FPGA debe ser accesible desde este servidor.

#### Instalación

Para instalar la interfaz web, compruebe que se cumplan todos los requisitos y siga los siguientes pasos:

1. Descargue el código fuente del repositorio del proyecto ([github.com/JSidrach/NetWatcher](https://github.com/JSidrach/NetWatcher)).
2. Descomprima el *.zip* y mueva la carpeta base *NetWatcher* al directorio público de *Apache* (normalmente */var/www/html/*).

3. Sitúese en la carpeta base *NetWatcher*.
4. Instale los paquetes y librerías necesarios ejecutando el siguiente comando:

```
sudo ./scripts/build.sh -install
```

### A.4. Configuración de la interfaz web

TODO: Configuración - IP, tema, idioma

### A.5. Uso de la aplicación

Una vez instalado y configurado tanto el Servicio Web FPGA como la interfaz web.  
TODO: [Introducción]. Rutas arriba, rutas abajo, documentaciones

### A.6. Solución de problemas

Si tras seguir las instrucciones paso a paso algo impide el correcto funcionamiento de la aplicación, puedes consultar la página de solución de problemas (en inglés), disponible dentro del repositorio del proyecto en:

[github.com/JSidrach/NetWatcher/blob/master/docs/wiki/Troubleshooting.md](https://github.com/JSidrach/NetWatcher/blob/master/docs/wiki/Troubleshooting.md)

# B

## Framework MVC propio

En este apéndice se explican los distintos componentes del framework propio desarrollado, del que hace uso la interfaz web. Lenguajes (php)

TODO: [Introducción], Imagen, Visión global

### B.1. Modelo Vista Controlador

TODO: Modelo Vista Controlador, interacción entre componentes, división en módulos, clases abstractas

### B.2. Manejo de rutas

TODO: Router, redireccionamiento, urls rest, parámetros, proxy, htaccess

## B.3. Redireccionamiento de peticiones a servicio web

Proxy

## B.4. Manejo de dependencias

TODO: Manejo de dependencias. Composer, bower

## B.5. Configuración

TODO: Config: Carpetas, defines

## B.6. Registro de eventos

TODO: Logger: Eventos, errores/warnings capturados, proxy

## B.7. Conclusiones

Se ha desarrollado un framework que sirve de base para la interfaz web, proporcionando un conjunto mínimo de funcionalidad necesaria para el problema planteado. Aunque desarrollarlo ha supuesto un coste temporal adicional para el proyecto, ha repercutido positivamente en fases posteriores de la implementación.

Conocer al detalle el framework sobre el que se basa la aplicación y tener un control total sobre el mismo ha permitido agilizar el proceso de desarrollo. Además, se ha adquirido experiencia en distintos conceptos útiles: orientación a objetos en PHP Hypertext Pre-processor (PHP), patrón de diseño Modelo-Vista-Controlador, manejo automático de dependencias y librerías externas, localización de interfaces web y codificación de un servidor proxy.



## API del Servicio Web FPGA

En este apéndice se detallan los métodos de la API del Servicio Web FPGA. Esta documentación puede también consultarse de manera interactiva (Figura C.1) en la propia página de la aplicación (en inglés), accediendo desde el navegador a la ruta relativa *NetWatcher/docs-back-end/*. Al ser un Servicio Web, cada método se invoca mediante una llamada Hypertext Transfer Protocol (HTTP) [3].

Se han agrupado todos los métodos disponibles en tres categorías, coincidiendo con los módulos implementados: gestión, trazas y estadísticas. Para cada método, se especifica su ruta (relativa a la ruta del Servicio Web), los parámetros necesarios para su invocación y las posibles salidas, con un ejemplo cada una. Los parámetros se pasan por la propia Uniform Resource Locator (URL) o en la cabecera en el caso del *timestamp*. Cada método devuelve siempre un código de estado HTTP [4] y una salida en formato JavaScript Object Notation (JSON) [5].

CAPTURES

[/captures/all](#)  
[/captures/delete/](#)  
[/captures/path](#)  
[/captures/pcap](#)  
[/captures/pcap/simple/](#)  
[/captures/rename/](#)  
[/captures/simple](#)  
[/captures/simple/pcap/](#)

MANAGER

[/player/init](#)  
[/player/install](#)  
[/player/start/](#)  
[/player/start/loop/](#)  
[/player/stop](#)  
[/recorder/init](#)  
[/recorder/install](#)  
[/recorder/start/](#)  
[/recorder/stop](#)  
[/storage/raid](#)  
[/system/reboot](#)

STATISTICS

[/info/delay](#)  
[/info/ping](#)  
[/info/status](#)  
[/storage/stats](#)

## FPGA Web Service API

Web Service to manage a traffic player/recorder FPGA

### Captures

#### Captures - /captures/all

Gets all the captures (simple/pcap format) in the CAPTURES\_DIR

GET

proxy.php?curl=/captures/all

Success 200

Field	Type	Description
code	String	Return code ('success')
type	String	Return type ('data')
captures	Object	Array of information for each capture
name	String	Name of the capture
type	String	Type of the capture (simple/pcap)
size	Number	Size of the capture (in bytes)
date	String	Date of the capture (yyyy-mm-dd hh:mm:ss)

Example data on success

{  
  "code": "success",  
  "type": "data",

Figura C.1: Documentación web de la API del Servicio Web FPGA.

## C.1. Métodos de Gestión

### C.1.1. POST /system/reboot

Reinicia el servidor remoto. Los parámetros necesarios para la invocación de este método se detallan en la Tabla C.1.

Parámetro	Clase	Tipo	Descripción
timestamp	Cabecera	Número	Tiempo transcurrido (ms) desde el 1 de Enero de 1970 00:00:00 UTC hasta ahora (salida de <i>Date.now()</i> )

Tabla C.1: Parámetros de */system/reboot*

A continuación se enumeran los distintos códigos de retorno asociados a este método:

- **200 OK:** se ha ordenado con éxito el reinicio del servidor remoto. Este código de éxito es retornado junto con los parámetros indicados en la Tabla C.2.



Nombre	Tipo	Descripción
code	Cadena de texto	Código de retorno ('success')
type	Cadena de texto	Código de tipo ('notification')
description	Cadena de texto	Descripción del código de retorno

Tabla C.2: Salida de `/system/reboot` asociada al código 200

Ejemplo de datos asociados al código 200:

```
{
  "code": "success",
  "type": "notification",
  "description": "The host is rebooting now."
}
```

- **412 Error:** el servidor remoto no puede ser reiniciado ya que la FPGA está siendo usada. Este código de error es retornado junto a los parámetros indicados en la Tabla C.3.

Nombre	Tipo	Descripción
code	Cadena de texto	Código de retorno ('error')
type	Cadena de texto	Código de tipo ('notification')
description	Cadena de texto	Descripción del error

Tabla C.3: Salida de `/system/reboot` asociada al código 412

Ejemplo de datos asociados al código 412:

```
{
  "code": "error",
  "type": "notification",
  "description": "The host can not be rebooted. The FPGA
    ↪ is being used."
}
```

### C.1.2. POST `/player/init`

Programa la FPGA para reproducir trazas y reinicia el servidor remoto. Los parámetros necesarios para la invocación de este método se detallan en la Tabla C.4.

Parámetro	Clase	Tipo	Descripción
timestamp	Cabecera	Número	Tiempo transcurrido (ms) desde el 1 de Enero de 1970 00:00:00 UTC hasta ahora (salida de <i>Date.now()</i> )

Tabla C.4: Parámetros de */player/init*

A continuación se enumeran los distintos códigos de retorno asociados a este método:

- **200 OK:** se ha programado con éxito la FPGA para reproducir trazas y se va a reiniciar el servidor remoto. Este código de éxito es retornado junto con los parámetros indicados en la Tabla C.5.

Nombre	Tipo	Descripción
code	Cadena de texto	Código de retorno ('success')
type	Cadena de texto	Código de tipo ('notification')
description	Cadena de texto	Descripción del código de retorno

Tabla C.5: Salida de */player/init* asociada al código 200

Ejemplo de datos asociados al código 200:

```
{
  "code": "success",
  "type": "notification",
  "description": "The FPGA has been initialized. The host
    ↪ will reboot now."
}
```

- **412 Error:** la FPGA no puede ser programada ya que está siendo usada. Este código de error es retornado junto a los parámetros indicados en la Tabla C.6.

Nombre	Tipo	Descripción
code	Cadena de texto	Código de retorno ('error')
type	Cadena de texto	Código de tipo ('fpga_invalid_state')
description	Cadena de texto	Descripción del error

Tabla C.6: Salida de */player/init* asociada al código 412

Ejemplo de datos asociados al código 412:

```
{
  "code": "error",
  "type": "fpga_invalid_state",
  "description": "Invalid State. The FPGA is already
    ↪ running, stop it to init the FPGA."
}
```

### C.1.3. POST /recorder/init

Programa la FPGA para capturar trazas y reinicia el servidor remoto. Los parámetros necesarios para la invocación de este método se detallan en la Tabla C.7.

Parámetro	Clase	Tipo	Descripción
timestamp	Cabecera	Número	Tiempo transcurrido (ms) desde el 1 de Enero de 1970 00:00:00 UTC hasta ahora (salida de <i>Date.now()</i> )

Tabla C.7: Parámetros de */recorder/init*

A continuación se enumeran los distintos códigos de retorno asociados a este método:

- **200 OK:** se ha programado con éxito la FPGA para reproducir trazas y se va a reiniciar el servidor remoto. Este código de éxito es retornado junto con los parámetros indicados en la Tabla C.8.

Nombre	Tipo	Descripción
code	Cadena de texto	Código de retorno ('success')
type	Cadena de texto	Código de tipo ('notification')
description	Cadena de texto	Descripción del código de retorno

Tabla C.8: Salida de */recorder/init* asociada al código 200

Ejemplo de datos asociados al código 200:

```
{
  "code": "success",
  "type": "notification",
  "description": "The FPGA has been initialized. The host
    ↪ will reboot now."
}
```

- **412 Error:** la FPGA no puede ser programada ya que está siendo usada. Este código de error es retornado junto a los parámetros indicados en la Tabla C.9.

Nombre	Tipo	Descripción
code	Cadena de texto	Código de retorno ('error')
type	Cadena de texto	Código de tipo ('fpga_invalid_state')
description	Cadena de texto	Descripción del error

Tabla C.9: Salida de */recorder/init* asociada al código 412

Ejemplo de datos asociados al código 412:

```
{
  "code": "error",
  "type": "fpga_invalid_state",
  "description": "Invalid State. The FPGA is already
    ↪ running, stop it to init the FPGA."
}
```

#### C.1.4. POST */player/install*

Instala y monta la FPGA para reproducir trazas. Los parámetros necesarios para la invocación de este método se detallan en la Tabla C.10.

Parámetro	Clase	Tipo	Descripción
timestamp	Cabecera	Número	Tiempo transcurrido (ms) desde el 1 de Enero de 1970 00:00:00 UTC hasta ahora (salida de <i>Date.now()</i> )

Tabla C.10: Parámetros de */player/install*

A continuación se enumeran los distintos códigos de retorno asociados a este método:

- **200 OK:** la FPGA se ha instalado y montado con éxito, y está lista para reproducir trazas. Este código de éxito es retornado junto con los parámetros indicados en la Tabla C.11.

Nombre	Tipo	Descripción
code	Cadena de texto	Código de retorno ('success')
type	Cadena de texto	Código de tipo ('notification')
description	Cadena de texto	Descripción del código de retorno

Tabla C.11: Salida de */player/install* asociada al código 200

Ejemplo de datos asociados al código 200:

```
{
  "code": "success",
  "type": "notification",
  "description": "The FPGA has been mounted and is ready
    ↪ to be used."
}
```

- **412 Error:** la FPGA no puede ser instalada y montada ya que no ha sido programada. Este código de error es retornado junto a los parámetros indicados en la Tabla C.12.

Nombre	Tipo	Descripción
code	Cadena de texto	Código de retorno ('error')
type	Cadena de texto	Código de tipo ('fpga_invalid_state')
description	Cadena de texto	Descripción del error

Tabla C.12: Salida de */player/install* asociada al código 412

Ejemplo de datos asociados al código 412:

```
{
  "code": "error",
  "type": "fpga_invalid_state",
  "description": "Invalid State. The FPGA must be
    ↪ programmed before mounted."
}
```

### C.1.5. POST /recorder/install

Instala y monta la FPGA para capturar trazas. Los parámetros necesarios para la invocación de este método se detallan en la Tabla C.13.

Parámetro	Clase	Tipo	Descripción
timestamp	Cabecera	Número	Tiempo transcurrido (ms) desde el 1 de Enero de 1970 00:00:00 UTC hasta ahora (salida de <i>Date.now()</i> )

Tabla C.13: Parámetros de */recorder/install*

A continuación se enumeran los distintos códigos de retorno asociados a este método:

- **200 OK:** la FPGA se ha instalado y montado con éxito, y está lista para capturar trazas. Este código de éxito es retornado junto con los parámetros indicados en la Tabla C.14.

Nombre	Tipo	Descripción
code	Cadena de texto	Código de retorno ('success')
type	Cadena de texto	Código de tipo ('notification')
description	Cadena de texto	Descripción del código de retorno

Tabla C.14: Salida de */recorder/install* asociada al código 200

Ejemplo de datos asociados al código 200:

```
{
  "code": "success",
  "type": "notification",
  "description": "The FPGA has been mounted and is ready
    ↪ to be used."
}
```

- **412 Error:** la FPGA no puede ser instalada y montada ya que no ha sido programada. Este código de error es retornado junto a los parámetros indicados en la Tabla C.15.

Nombre	Tipo	Descripción
code	Cadena de texto	Código de retorno ('error')
type	Cadena de texto	Código de tipo ('fpga_invalid_state')
description	Cadena de texto	Descripción del error

Tabla C.15: Salida de `/recorder/install` asociada al código 412

Ejemplo de datos asociados al código 412:

```
{
  "code": "error",
  "type": "fpga_invalid_state",
  "description": "Invalid State. The FPGA must be
    ↪ programmed before mounted."
}
```

### C.1.6. POST `/player/start/:capturename/:mask/:ifg`

Reproduce una traza con los parámetros indicados. Los parámetros necesarios para la invocación de este método se detallan en la Tabla C.16.

Parámetro	Clase	Tipo	Descripción
timestamp	Cabecera	Número	Tiempo transcurrido (ms) desde el 1 de Enero de 1970 00:00:00 UTC hasta ahora (salida de <code>Date.now()</code> )
capturename	Parámetro URL	Cadena de texto	Nombre de la traza a reproducir
mask	Parámetro URL	Número	Máscara de reproducción (0-1-2-3)
ifg	Parámetro URL	Número	IFG (0 para tasa original)

Tabla C.16: Parámetros de `/player/start`

A continuación se enumeran los distintos códigos de retorno asociados a este método:

- **200 OK:** la FPGA ha empezado a reproducir la traza seleccionada con los parámetros indicados. Este código de éxito es retornado junto con los parámetros indicados en la Tabla C.17.

Nombre	Tipo	Descripción
code	Cadena de texto	Código de retorno ('success')
type	Cadena de texto	Código de tipo ('notification')
description	Cadena de texto	Descripción del código de retorno

Tabla C.17: Salida de */player/start* asociada al código 200

Ejemplo de datos asociados al código 200:

```
{
  "code": "success",
  "type": "notification",
  "description": "The FPGA has started playing a capture
    ↪ ."
}
```

- **400 Error:** la FPGA no puede reproducir la traza ya que los parámetros pasados son inválidos. Este código de error es retornado junto a los parámetros indicados en la Tabla C.18.

Nombre	Tipo	Descripción
code	Cadena de texto	Código de retorno ('error')
type	Cadena de texto	Código de tipo ('fpga_invalid_state')
description	Cadena de texto	Descripción del error

Tabla C.18: Salida de */player/start* asociada al código 400

Ejemplo de datos asociados al código 400:

```
{
  "code": "error",
  "type": "notification",
  "description": "Invalid parameters. The FPGA could not
    ↪ start playing a capture."
}
```

- **412 Error:** la FPGA no puede reproducir la traza ya que no ha sido instalada y montada para reproducir trazas. Este código de error es retornado junto a los parámetros indicados en la Tabla C.19.



Nombre	Tipo	Descripción
code	Cadena de texto	Código de retorno ('error')
type	Cadena de texto	Código de tipo ('fpga_invalid_state')
description	Cadena de texto	Descripción del error

 Tabla C.19: Salida de `/player/start` asociada al código 412

Ejemplo de datos asociados al código 412:

```
{
  "code": "error",
  "type": "fpga_invalid_state",
  "description": "Invalid State. The FPGA is not
    ↪ programmed and mounted as a player."
}
```

### C.1.7. POST `/player/start/loop/:capturename/:mask/:ifg`

Reproduce en bucle una traza con los parámetros indicados. Los parámetros necesarios para la invocación de este método se detallan en la Tabla C.20.

Parámetro	Clase	Tipo	Descripción
timestamp	Cabecera	Número	Tiempo transcurrido (ms) desde el 1 de Enero de 1970 00:00:00 UTC hasta ahora (salida de <code>Date.now()</code> )
capturename	Parámetro URL	Cadena de texto	Nombre de la traza a reproducir
mask	Parámetro URL	Número	Máscara de reproducción (0-1-2-3)
ifg	Parámetro URL	Número	IFG (0 para tasa original)

 Tabla C.20: Parámetros de `/player/start/loop`

A continuación se enumeran los distintos códigos de retorno asociados a este método:

- **200 OK:** la FPGA ha empezado a reproducir en bucle la traza seleccionada con los parámetros indicados. Este código de éxito es retornado junto con los parámetros indicados en la Tabla C.21.

Nombre	Tipo	Descripción
code	Cadena de texto	Código de retorno ('success')
type	Cadena de texto	Código de tipo ('notification')
description	Cadena de texto	Descripción del código de retorno

Tabla C.21: Salida de */player/start/loop* asociada al código 200

Ejemplo de datos asociados al código 200:

```
{
  "code": "success",
  "type": "notification",
  "description": "The FPGA has started playing a capture
    ↪ ."
}
```

- **400 Error:** la FPGA no puede reproducir la traza ya que los parámetros pasados son inválidos. Este código de error es retornado junto a los parámetros indicados en la Tabla C.22.

Nombre	Tipo	Descripción
code	Cadena de texto	Código de retorno ('error')
type	Cadena de texto	Código de tipo ('fpga_invalid_state')
description	Cadena de texto	Descripción del error

Tabla C.22: Salida de */player/start/loop* asociada al código 400

Ejemplo de datos asociados al código 400:

```
{
  "code": "error",
  "type": "notification",
  "description": "Invalid parameters. The FPGA could not
    ↪ start playing a capture."
}
```

- **412 Error:** la FPGA no puede reproducir la traza ya que no ha sido instalada y montada para reproducir trazas. Este código de error es retornado junto a los parámetros indicados en la Tabla C.23.

Nombre	Tipo	Descripción
code	Cadena de texto	Código de retorno ('error')
type	Cadena de texto	Código de tipo ('fpga_invalid_state')
description	Cadena de texto	Descripción del error

Tabla C.23: Salida de `/player/start/loop` asociada al código 412

Ejemplo de datos asociados al código 412:

```
{
  "code": "error",
  "type": "fpga_invalid_state",
  "description": "Invalid State. The FPGA is not
    ↪ programmed and mounted as a player."
}
```

### C.1.8. POST `/recorder/start/:capturename/:port/:bytes`

Captura una traza con los parámetros indicados. Los parámetros necesarios para la invocación de este método se detallan en la Tabla C.24.

Parámetro	Clase	Tipo	Descripción
timestamp	Cabecera	Número	Tiempo transcurrido (ms) desde el 1 de Enero de 1970 00:00:00 UTC hasta ahora (salida de <i>Date.now()</i> )
capturename	Parámetro URL	Cadena de texto	Nombre de la traza a capturar
port	Parámetro URL	Número	Puerto a capturar (0-1-2-3)
bytes	Parámetro URL	Número	Bytes a capturar

Tabla C.24: Parámetros de `/recorder/start`

A continuación se enumeran los distintos códigos de retorno asociados a este método:

- **200 OK:** la FPGA ha empezado a capturar una traza con los parámetros indicados. Este código de éxito es retornado junto con los parámetros indicados en la Tabla C.25.

Nombre	Tipo	Descripción
code	Cadena de texto	Código de retorno ('success')
type	Cadena de texto	Código de tipo ('notification')
description	Cadena de texto	Descripción del código de retorno

Tabla C.25: Salida de */recorder/start* asociada al código 200

Ejemplo de datos asociados al código 200:

```
{
  "code": "success",
  "type": "notification",
  "description": "The FPGA has started recording data."
}
```

- **400 Error:** la FPGA no puede empezar a capturar ya que los parámetros pasados son inválidos. Este código de error es retornado junto a los parámetros indicados en la Tabla C.26.

Nombre	Tipo	Descripción
code	Cadena de texto	Código de retorno ('error')
type	Cadena de texto	Código de tipo ('fpga_invalid_state')
description	Cadena de texto	Descripción del error

Tabla C.26: Salida de */recorder/start* asociada al código 400

Ejemplo de datos asociados al código 400:

```
{
  "code": "error",
  "type": "notification",
  "description": "Invalid capture name (must not exist)."
}
```

- **412 Error:** la FPGA no puede empezar a capturar ya que no ha sido instalada y montada para capturar trazas. Este código de error es retornado junto a los parámetros indicados en la Tabla C.27.

Nombre	Tipo	Descripción
code	Cadena de texto	Código de retorno ('error')
type	Cadena de texto	Código de tipo ('fpga_invalid_state')
description	Cadena de texto	Descripción del error

Tabla C.27: Salida de */recorder/start* asociada al código 412

Ejemplo de datos asociados al código 412:

```
{
  "code": "error",
  "type": "fpga_invalid_state",
  "description": "Invalid State. The FPGA is not
    ↪ programmed and mounted as a recorder."
}
```

### C.1.9. POST */player/stop*

Detiene la reproducción de traza en curso. Los parámetros necesarios para la invocación de este método se detallan en la Tabla C.28.

Parámetro	Clase	Tipo	Descripción
timestamp	Cabecera	Número	Tiempo transcurrido (ms) desde el 1 de Enero de 1970 00:00:00 UTC hasta ahora (salida de <i>Date.now()</i> )

Tabla C.28: Parámetros de */player/stop*

A continuación se enumeran los distintos códigos de retorno asociados a este método:

- **200 OK:** la FPGA ha detenido la reproducción en curso. Este código de éxito es retornado junto con los parámetros indicados en la Tabla C.29.

Nombre	Tipo	Descripción
code	Cadena de texto	Código de retorno ('success')
type	Cadena de texto	Código de tipo ('notification')
description	Cadena de texto	Descripción del código de retorno

Tabla C.29: Salida de */player/stop* asociada al código 200

Ejemplo de datos asociados al código 200:

```
{
  "code": "success",
  "type": "notification",
  "description": "The FPGA has stopped playing a capture
    ↪ ."
}
```

- **412 Error:** no se ha podido detener la reproducción ya que no existe ninguna en curso. Este código de error es retornado junto a los parámetros indicados en la Tabla C.30.

Nombre	Tipo	Descripción
code	Cadena de texto	Código de retorno ('error')
type	Cadena de texto	Código de tipo ('fpga_invalid_state')
description	Cadena de texto	Descripción del error

Tabla C.30: Salida de */player/stop* asociada al código 412

Ejemplo de datos asociados al código 412:

```
{
  "code": "error",
  "type": "fpga_invalid_state",
  "description": "Invalid State. The FPGA is not playing
    ↪ a capture."
}
```

### C.1.10. POST */recorder/stop*

Detiene la captura de traza en curso. Los parámetros necesarios para la invocación de este método se detallan en la Tabla C.31.

Parámetro	Clase	Tipo	Descripción
timestamp	Cabecera	Número	Tiempo transcurrido (ms) desde el 1 de Enero de 1970 00:00:00 UTC hasta ahora (salida de <i>Date.now()</i> )

 Tabla C.31: Parámetros de */recorder/stop*

A continuación se enumeran los distintos códigos de retorno asociados a este método:

- **200 OK:** la FPGA ha detenido la captura en curso. Este código de éxito es retornado junto con los parámetros indicados en la Tabla C.32.

Nombre	Tipo	Descripción
code	Cadena de texto	Código de retorno ('success')
type	Cadena de texto	Código de tipo ('notification')
description	Cadena de texto	Descripción del código de retorno

 Tabla C.32: Salida de */recorder/stop* asociada al código 200

Ejemplo de datos asociados al código 200:

```
{
  "code": "success",
  "type": "notification",
  "description": "The FPGA has stopped recording data."
}
```

- **412 Error:** no se ha podido detener la captura ya que no existe ninguna en curso. Este código de error es retornado junto a los parámetros indicados en la Tabla C.33.

Nombre	Tipo	Descripción
code	Cadena de texto	Código de retorno ('error')
type	Cadena de texto	Código de tipo ('fpga_invalid_state')
description	Cadena de texto	Descripción del error

 Tabla C.33: Salida de */recorder/stop* asociada al código 412

Ejemplo de datos asociados al código 412:

```
{
  "code": "error",
  "type": "fpga_invalid_state",
  "description": "Invalid State. The FPGA is not
    ↪ recording data."
}
```

### C.1.11. DELETE /storage/raid

Borra (formatea y vuelve a crear) el RAID de almacenamiento. Los parámetros necesarios para la invocación de este método se detallan en la Tabla C.34.

Parámetro	Clase	Tipo	Descripción
timestamp	Cabecera	Número	Tiempo transcurrido (ms) desde el 1 de Enero de 1970 00:00:00 UTC hasta ahora (salida de <i>Date.now()</i> )

Tabla C.34: Parámetros de */storage/raid*

A continuación se enumeran los distintos códigos de retorno asociados a este método:

- **200 OK:** se ha formateado y vuelto a montar el RAID de almacenamiento. Este código de éxito es retornado junto con los parámetros indicados en la Tabla C.35.

Nombre	Tipo	Descripción
code	Cadena de texto	Código de retorno ('success')
type	Cadena de texto	Código de tipo ('notification')
description	Cadena de texto	Descripción del código de retorno

Tabla C.35: Salida de */storage/raid* asociada al código 200



Ejemplo de datos asociados al código 200:

```
{
  "code": "success",
  "type": "notification",
  "description": "The RAID has been formatted and mounted
    ↪ properly."
}
```

- **412 Error:** no se ha formateado el RAID ya que o está activado o la FPGA está en uso. Este código de error es retornado junto a los parámetros indicados en la Tabla C.36.

Nombre	Tipo	Descripción
code	Cadena de texto	Código de retorno ('error')
type	Cadena de texto	Código de tipo ('fpga_invalid_state')
description	Cadena de texto	Descripción del error

Tabla C.36: Salida de */storage/raid* asociada al código 412

Ejemplo de datos asociados al código 412:

```
{
  "code": "error",
  "type": "notification",
  "description": "The RAID could not be formatted, RAID
    ↪ configuration option is not set or the FPGA is
    ↪ running."
}
```

## C.2. Métodos de Estadísticas

### C.2.1. GET /info/ping

Método para comprobar si el servidor está operativo. Este método no requiere parámetros.

A continuación se enumeran los distintos códigos de retorno asociados a este método:

- **200 OK:** servidor operativo. Este código de éxito es retornado junto con los parámetros indicados en la Tabla C.37.

Nombre	Tipo	Descripción
code	Cadena de texto	Código de retorno ('success')

Tabla C.37: Salida de */info/ping* asociada al código 200

Ejemplo de datos asociados al código 200:

```
{
  "code": "success"
}
```

### C.2.2. GET */info/delay*

Solicita la diferencia de tiempos existente entre los relojes del cliente y del servidor (en segundos). Los parámetros necesarios para la invocación de este método se detallan en la Tabla C.38.

Parámetro	Clase	Tipo	Descripción
timestamp	Cabecera	Número	Tiempo transcurrido (ms) desde el 1 de Enero de 1970 00:00:00 UTC hasta ahora (salida de <i>Date.now()</i> )

Tabla C.38: Parámetros de */info/delay*

A continuación se enumeran los distintos códigos de retorno asociados a este método:

- **200 OK:** solicitud con éxito. Este código de éxito es retornado junto con los parámetros indicados en la Tabla C.39.

Nombre	Tipo	Descripción
code	Cadena de texto	Código de retorno ('success')
type	Cadena de texto	Código de tipo ('data')
delay	Número	Diferencia entre relojes (en segundos)
maxDelay	Número	Máxima diferencia permitida (en segundos)

Tabla C.39: Salida de */info/delay* asociada al código 200

Ejemplo de datos asociados al código 200:

```
{
  "code": "success",
  "type": "data",
  "delay": 1,
  "maxDelay": 30
}
```

### C.2.3. GET /info/status

Solicita el estado actual de la FPGA. Este método no requiere parámetros.

A continuación se enumeran los distintos códigos de retorno asociados a este método:

- **200 OK - hugepages\_off:** opción *HugePages* del sistema operativo no activa. Este código de éxito es retornado junto con los parámetros indicados en la Tabla C.40.

Nombre	Tipo	Descripción
code	Cadena de texto	Código de estado ('hugepages_off')
description	Cadena de texto	Descripción del código de estado

Tabla C.40: Salida de */info/status* asociada al código 200 - hugepages\_off

Ejemplo de datos asociados al código 200 - hugepages\_off:

```
{
  "status": "hugepages_off",
  "description": "HugePages is not enabled. To fix this,
    ↪ the host should be rebooted with this option
    ↪ selected on the GRUB menu."
}
```

- **200 OK - init\_off:** la FPGA aún no ha sido configurada para captura o reproducción. Este código de éxito es retornado junto con los parámetros indicados en la Tabla C.41.

Nombre	Tipo	Descripción
code	Cadena de texto	Código de estado ('init_off')
description	Cadena de texto	Descripción del código de estado

Tabla C.41: Salida de */info/status* asociada al código 200 - init\_off

Ejemplo de datos asociados al código 200 - init\_off:

```
{
  "status": "init_off",
  "description": "The FPGA is not configured either as
    ↪ player or as recorder yet."
}
```

- **200 OK - mount\_off:** la FPGA está inicializada pero no ha sido montada aún. Este código de éxito es retornado junto con los parámetros indicados en la Tabla C.42.

Nombre	Tipo	Descripción
code	Cadena de texto	Código de estado ('mount_off')
description	Cadena de texto	Descripción del código de estado

Tabla C.42: Salida de */info/status* asociada al código 200 - mount\_off

Ejemplo de datos asociados al código 200 - mount\_off:

```
{
  "status": "mount_off",
  "description": "The FPGA is initialized but not mounted
    ↪ ."
}
```

- **200 OK - player\_ready:** la FPGA está lista para reproducir una traza. Este código de éxito es retornado junto con los parámetros indicados en la Tabla C.43.

Nombre	Tipo	Descripción
code	Cadena de texto	Código de estado ('player_ready')
description	Cadena de texto	Descripción del código de estado

Tabla C.43: Salida de */info/status* asociada al código 200 - player\_ready

Ejemplo de datos asociados al código 200 - player\_ready:

```
{
  "status": "player_ready",
  "description": "The FPGA is ready to reproduce a
    ↪ capture."
}
```

- **200 OK - recorder\_ready:** la FPGA está lista para capturar una traza. Este código de éxito es retornado junto con los parámetros indicados en la Tabla C.44.

Nombre	Tipo	Descripción
code	Cadena de texto	Código de estado ('recorder_ready')
description	Cadena de texto	Descripción del código de estado

Tabla C.44: Salida de */info/status* asociada al código 200 - recorder\_ready

Ejemplo de datos asociados al código 200 - recorder\_ready:

```
{
  "status": "player_ready",
  "description": "The FPGA is ready to record a capture."
}
```

- **200 OK - playing:** la FPGA está reproduciendo una traza. Este código de éxito es retornado junto con los parámetros indicados en la Tabla C.45.

Nombre	Tipo	Descripción
code	Cadena de texto	Código de estado ('playing')
description	Cadena de texto	Descripción del código de estado
capture	Cadena de texto	Nombre de la traza en reproducción
size	Número	Tamaño de la traza en reproducción
date	Cadena de texto	Fecha de la traza en reproducción
elapsed_time	Número	Tiempo transcurrido desde el inicio (segundos)
packets_sent	Número	Paquetes enviados
loop	Booleano	<i>true</i> si se está reproduciendo en bucle
interframe_gap	Número	IFG (0 si es la tasa original)
mask	Número	Máscara de reproducción (0-1-2-3)

Tabla C.45: Salida de */info/status* asociada al código 200 - playing

Ejemplo de datos asociados al código 200 - playing:

```
{
  "status": "playing",
  "description": "The FPGA is reproducing a capture.",
  "capture": "my_capture.simple",
  "size": 714131923845,
  "date": "2014-09-29 15:40:34",
  "elapsed_time": 548,
  "packets_sent": 394578123,
  "loop": true,
  "interframe_gap": 0,
  "mask": 3
}
```

- **200 OK - recording:** la FPGA está capturando una traza. Este código de éxito es retornado junto con los parámetros indicados en la Tabla C.46.

Nombre	Tipo	Descripción
code	Cadena de texto	Código de estado ('recording')
description	Cadena de texto	Descripción del código de estado
capture	Cadena de texto	Nombre de la traza a capturar
elapsed_time	Número	Tiempo transcurrido desde el inicio (segundos)
bytes_captured	Número	Bytes capturados
bytes_total	Número	Total de bytes a capturar
port	Número	Puerto que está siendo capturado (0-1-2-3)

Tabla C.46: Salida de */info/status* asociada al código 200 - recording

Ejemplo de datos asociados al código 200 - recording:

```
{
  "status": "recording",
  "description": "The FPGA is recording a capture.",
  "capture": "flows_test",
  "elapsed_time": 447,
  "bytes_captured": 5984234711238,
  "bytes_total": 234856352341724128,
  "port": 2
}
```

#### C.2.4. GET /storage/stats

Solicita estadísticas del almacenamiento. Este método no requiere parámetros.

A continuación se enumeran los distintos códigos de retorno asociados a este método:

- **200 OK:** estadísticas obtenidas. Este código de éxito es retornado junto con los parámetros indicados en la Tabla C.47.

Nombre	Tipo	Descripción
code	Cadena de texto	Código de retorno ('success')
type	Cadena de texto	Código de tipo ('data')
total_space	Número	Espacio total (en bytes)
used_space	Número	Espacio utilizado (en bytes)
raid_stats	Objeto	Estadísticas del RAID
raid_stats.raid_active	Booleano	<i>true</i> si el RAID está activo
raid_stats.write_speed	Número	Velocidad de escritura del RAID (B/s)
raid_stats.disks	Vector	Vector con estadísticas de cada disco
raid_stats.disks.name	Cadena de texto	Nombre del disco
raid_stats.disks.write_speed	Número	Velocidad de escritura del disco (B/s)

Tabla C.47: Salida de /storage/stats asociada al código 200

Ejemplo de datos asociados al código 200:

```
{
  "code": "success",
  "type": "data",
  "total_space": 240972104,
  "used_space": 70828412,
  "raid_stats": {
    "raid_active": true,
    "raid_name": "/dev/md0",
    "write_speed": 4051114978890,
    "disks": [
      {
        "name": "/dev/sdc",
        "write_speed": 15435231341
      },
      {
        "name": "/dev/sdd",
        "write_speed": 32112351239
      },
      {
        "name": "/dev/sde",
        "write_speed": 19123843109
      }
    ]
  }
}
```

## C.3. Métodos de Trazas

### C.3.1. GET /captures/all

Solicita información sobre todas las trazas disponibles. Este método no requiere parámetros.



A continuación se enumeran los distintos códigos de retorno asociados a este método:

- **200 OK:** información de las trazas obtenida. Este código de éxito es retornado junto con los parámetros indicados en la Tabla C.48.

Nombre	Tipo	Descripción
code	Cadena de texto	Código de retorno ('success')
type	Cadena de texto	Código de tipo ('data')
captures	Vector	Vector con información de las trazas
captures.name	Cadena de texto	Nombre de la traza
captures.type	Cadena de texto	Tipo de traza (simple o pcap)
captures.size	Número	Tamaño de la traza (en bytes)
captures.date	Cadena de texto	Fecha de la traza

Tabla C.48: Salida de */captures/all* asociada al código 200

Ejemplo de datos asociados al código 200:

```
{
  "code": "success",
  "type": "data",
  "captures": [
    {
      "name": "flows_crc",
      "type": "simple",
      "size": 956092345897,
      "date": "2015-03-05 13:42:15"
    },
    {
      "name": "my_capture.simple",
      "type": "pcap",
      "size": 4981234712,
      "date": "2015-02-16 11:08:18"
    },
    {
      "name": "capture_labs.pcap",
      "type": "pcap",
      "size": 30563653141,
      "date": "2015-01-11 17:32:19"
    }
  ]
}
```

### C.3.2. GET /captures/simple

Solicita información sobre todas las trazas en formato simple disponibles. Este método no requiere parámetros.

A continuación se enumeran los distintos códigos de retorno asociados a este método:

- **200 OK:** información de las trazas en formato simple obtenida. Este código de éxito

es retornado junto con los parámetros indicados en la Tabla C.49.

Nombre	Tipo	Descripción
code	Cadena de texto	Código de retorno ('success')
type	Cadena de texto	Código de tipo ('data')
captures	Vector	Vector con información de las trazas
captures.name	Cadena de texto	Nombre de la traza
captures.type	Cadena de texto	Tipo de traza (simple)
captures.size	Número	Tamaño de la traza (en bytes)
captures.date	Cadena de texto	Fecha de la traza

Tabla C.49: Salida de */captures/simple* asociada al código 200

Ejemplo de datos asociados al código 200:

```
{
  "code": "success",
  "type": "data",
  "captures": [
    {
      "name": "flows_crc",
      "type": "simple",
      "size": 956092345897,
      "date": "2015-03-05 13:42:15"
    }
  ]
}
```

### C.3.3. GET */captures/pcap*

Solicita información sobre todas las trazas en formato pcap disponibles. Este método no requiere parámetros.

A continuación se enumeran los distintos códigos de retorno asociados a este método:

- **200 OK:** información de las trazas en formato pcap obtenida. Este código de éxito es retornado junto con los parámetros indicados en la Tabla C.50.

Nombre	Tipo	Descripción
code	Cadena de texto	Código de retorno ('success')
type	Cadena de texto	Código de tipo ('data')
captures	Vector	Vector con información de las trazas
captures.name	Cadena de texto	Nombre de la traza
captures.type	Cadena de texto	Tipo de traza (pcap)
captures.size	Número	Tamaño de la traza (en bytes)
captures.date	Cadena de texto	Fecha de la traza

Tabla C.50: Salida de `/captures/pcap` asociada al código 200

Ejemplo de datos asociados al código 200:

```
{
  "code": "success",
  "type": "data",
  "captures": [
    {
      "name": "my_capture.simple",
      "type": "pcap",
      "size": 4981234712,
      "date": "2015-02-16 11:08:18"
    },
    {
      "name": "capture_labs.pcap",
      "type": "pcap",
      "size": 30563653141,
      "date": "2015-01-11 17:32:19"
    }
  ]
}
```

### C.3.4. GET `/captures/path`

Solicita información sobre dónde se almacenan las trazas. Este método no requiere parámetros.

A continuación se enumeran los distintos códigos de retorno asociados a este método:

- **200 OK:** información de las trazas en formato pcap obtenida. Este código de éxito es retornado junto con los parámetros indicados en la Tabla C.51.

Nombre	Tipo	Descripción
code	Cadena de texto	Código de retorno ('success')
type	Cadena de texto	Código de tipo ('data')
path	Cadena de texto	Carpeta donde se almacenan las trazas

Tabla C.51: Salida de `/captures/path` asociada al código 200

Ejemplo de datos asociados al código 200:

```
{
  "code": "success",
  "type": "data",
  "path": "/dev/raid/captures/"
}
```

### C.3.5. PUT `/captures/rename/:oldname/:newname`

Renombra una traza. Los parámetros necesarios para la invocación de este método se detallan en la Tabla C.52.

Parámetro	Clase	Tipo	Descripción
timestamp	Cabecera	Número	Tiempo transcurrido (ms) desde el 1 de Enero de 1970 00:00:00 UTC hasta ahora (salida de <i>Date.now()</i> )
oldname	Parámetro URL	Cadena de texto	Nombre de la traza a renombrar
newname	Parámetro URL	Cadena de texto	Nuevo nombre de la traza

Tabla C.52: Parámetros de `/captures/rename`

A continuación se enumeran los distintos códigos de retorno asociados a este método:

- **200 OK:** la traza se ha renombrado con éxito. Este código de éxito es retornado junto con los parámetros indicados en la Tabla C.53.

Nombre	Tipo	Descripción
code	Cadena de texto	Código de retorno ('success')
type	Cadena de texto	Código de tipo ('notification')
description	Cadena de texto	Descripción del código de retorno

Tabla C.53: Salida de */captures/rename* asociada al código 200

Ejemplo de datos asociados al código 200:

```
{
  "code": "success",
  "type": "notification",
  "description": "The capture has been successfully
    ↪ renamed."
}
```

- **400 Error:** no se ha podido renombrar la traza (está en uso o el nuevo nombre no es válido). Este código de error es retornado junto a los parámetros indicados en la Tabla C.54.

Nombre	Tipo	Descripción
code	Cadena de texto	Código de retorno ('error')
type	Cadena de texto	Código de tipo ('notification')
description	Cadena de texto	Descripción del error

Tabla C.54: Salida de */captures/rename* asociada al código 400

Ejemplo de datos asociados al código 400:

```
{
  "code": "error",
  "type": "notification",
  "description": "The capture could not be renamed. The
    ↪ new name is already in use or the capture is
    ↪ being used."
}
```

### C.3.6. PUT /captures/simple/pcap/:name/:convertedname

Convierte una traza de formato simple a formato pcap. Los parámetros necesarios para la invocación de este método se detallan en la Tabla C.55.

Parámetro	Clase	Tipo	Descripción
timestamp	Cabecera	Número	Tiempo transcurrido (ms) desde el 1 de Enero de 1970 00:00:00 UTC hasta ahora (salida de <i>Date.now()</i> )
name	Parámetro URL	Cadena de texto	Nombre de la traza a convertir
convertedname	Parámetro URL	Cadena de texto	Nombre de la traza convertida

Tabla C.55: Parámetros de /captures/simple/pcap

A continuación se enumeran los distintos códigos de retorno asociados a este método:

- **200 OK:** la traza se ha convertido con éxito. Este código de éxito es retornado junto con los parámetros indicados en la Tabla C.56.

Nombre	Tipo	Descripción
code	Cadena de texto	Código de retorno ('success')
type	Cadena de texto	Código de tipo ('notification')
description	Cadena de texto	Descripción del código de retorno

Tabla C.56: Salida de /captures/simple/pcap asociada al código 200

Ejemplo de datos asociados al código 200:

```
{
  "code": "success",
  "type": "notification",
  "description": "The capture has been successfully
    ↪ converted."
}
```

- **400 Error:** no se ha podido convertir la traza (el nuevo nombre no es válido o la traza original no está en formato simple). Este código de error es retornado junto a los parámetros indicados en la Tabla C.57.

Nombre	Tipo	Descripción
code	Cadena de texto	Código de retorno ('error')
type	Cadena de texto	Código de tipo ('notification')
description	Cadena de texto	Descripción del error

Tabla C.57: Salida de `/captures/simple/pcap` asociada al código 400

Ejemplo de datos asociados al código 400:

```
{
  "code": "error",
  "type": "notification",
  "description": "The capture could not be converted. The
    ↪ capture has not a valid format or name, or the
    ↪ new name is already in use."
}
```

### C.3.7. PUT `/captures/pcap/simple/:name/:convertedname`

Convierte una traza de formato pcap a formato simple. Los parámetros necesarios para la invocación de este método se detallan en la Tabla C.58.

Parámetro	Clase	Tipo	Descripción
timestamp	Cabecera	Número	Tiempo transcurrido (ms) desde el 1 de Enero de 1970 00:00:00 UTC hasta ahora (salida de <i>Date.now()</i> )
name	Parámetro URL	Cadena de texto	Nombre de la traza a convertir
convertedname	Parámetro URL	Cadena de texto	Nombre de la traza convertida

Tabla C.58: Parámetros de `/captures/pcap/simple`

A continuación se enumeran los distintos códigos de retorno asociados a este método:

- **200 OK:** la traza se ha convertido con éxito. Este código de éxito es retornado junto con los parámetros indicados en la Tabla C.59.



Nombre	Tipo	Descripción
code	Cadena de texto	Código de retorno ('success')
type	Cadena de texto	Código de tipo ('notification')
description	Cadena de texto	Descripción del código de retorno

Tabla C.59: Salida de */captures/pcap/simple* asociada al código 200

Ejemplo de datos asociados al código 200:

```
{
  "code": "success",
  "type": "notification",
  "description": "The capture has been successfully
    ↪ converted."
}
```

- **400 Error:** no se ha podido convertir la traza (el nuevo nombre no es válido o la traza original no está en formato pcap). Este código de error es retornado junto a los parámetros indicados en la Tabla C.60.

Nombre	Tipo	Descripción
code	Cadena de texto	Código de retorno ('error')
type	Cadena de texto	Código de tipo ('notification')
description	Cadena de texto	Descripción del error

Tabla C.60: Salida de */captures/pcap/simple* asociada al código 400

Ejemplo de datos asociados al código 400:

```
{
  "code": "error",
  "type": "notification",
  "description": "The capture could not be converted. The
    ↪ capture has not a valid format or name, or the
    ↪ new name is already in use."
}
```

### C.3.8. DELETE /captures/delete/:name

Borra una traza. Los parámetros necesarios para la invocación de este método se detallan en la Tabla C.61.

Parámetro	Clase	Tipo	Descripción
timestamp	Cabecera	Número	Tiempo transcurrido (ms) desde el 1 de Enero de 1970 00:00:00 UTC hasta ahora (salida de <i>Date.now()</i> )
name	Parámetro URL	Cadena de texto	Nombre de la traza a borrar

Tabla C.61: Parámetros de /captures/delete

A continuación se enumeran los distintos códigos de retorno asociados a este método:

- **200 OK:** la traza se ha borrado con éxito. Este código de éxito es retornado junto con los parámetros indicados en la Tabla C.62.

Nombre	Tipo	Descripción
code	Cadena de texto	Código de retorno ('success')
type	Cadena de texto	Código de tipo ('notification')
description	Cadena de texto	Descripción del código de retorno

Tabla C.62: Salida de /captures/delete asociada al código 200

Ejemplo de datos asociados al código 200:

```
{
  "code": "success",
  "type": "notification",
  "description": "The capture has been successfully
    ↪ deleted."
}
```

- **400 Error:** no se ha podido borrar la traza (está en uso o no existe). Este código de error es retornado junto a los parámetros indicados en la Tabla C.63.

Nombre	Tipo	Descripción
code	Cadena de texto	Código de retorno ('error')
type	Cadena de texto	Código de tipo ('notification')
description	Cadena de texto	Descripción del error

Tabla C.63: Salida de */captures/delete* asociada al código 400

Ejemplo de datos asociados al código 400:

```
{
  "code": "error",
  "type": "notification",
  "description": "The capture could not be deleted (it is
    ↪ in use or it does not exist)."
```