

# Experiencias propias en la contribución a proyectos de software libre: Volatility

**Ricardo J. Rodríguez**

© **All wrongs reversed** – bajo licencia CC BY-NC-SA 4.0



**Universidad**  
Zaragoza

Dept. de Informática e Ingeniería de Sistemas  
Universidad de Zaragoza

13 de mayo, 2023

**esLibre**

Zaragoza, España





- **Profesor Titular de Universidad**

- **Líneas de investigación:**

- Análisis de software
- Forense digital
- Seguridad ofensiva
- Análisis de seguridad y supervivencia con modelos formales



## ■ Profesor Titular de Universidad

### ■ Líneas de investigación:

- Análisis de software
- Forense digital
- Seguridad ofensiva
- Análisis de seguridad y supervivencia con modelos formales

### ■ Equipo de investigación – *¡hacemos cosas chulas!* 😊

- <https://reversea.me>
- <https://twitter.com/reverseame/>
- <https://t.me/reverseame>



Dr. Javier Carrillo  
Post-doc



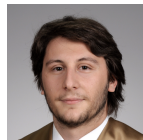
Daniel Uroz  
Estudiantes PhD



Razvan Raducu



Daniel Huici



Miguel Moniente  
Estudiantes TFM

# Agenda

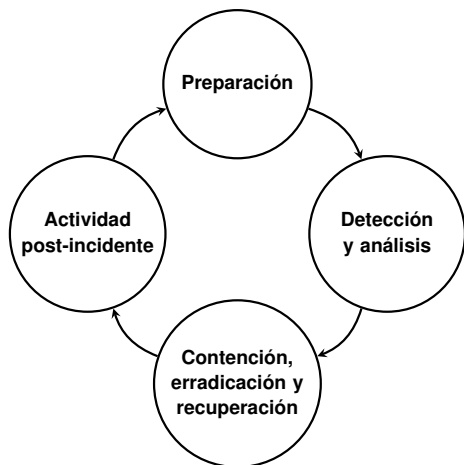
- 1 Introducción
- 2 Volatility
- 3 Nuestra primera experiencia con Volatility
- 4 Nuestra experiencia actual con Volatility
- 5 Conclusiones

# Agenda

- 1** Introducción
- 2 Volatility
- 3 Nuestra primera experiencia con Volatility
- 4 Nuestra experiencia actual con Volatility
- 5 Conclusiones

# Introducción

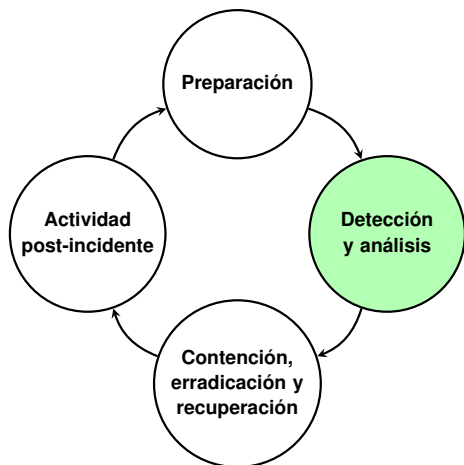
Un poco de recapitulación...



*Respuesta a incidentes definida por el NIST*

# Introducción

Un poco de recapitulación...

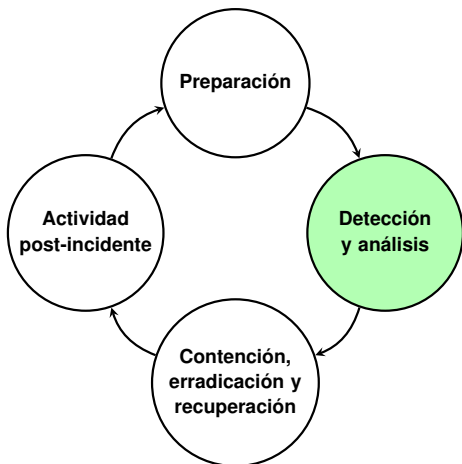


- **Análisis forense de red**
- **Análisis forense de ordenadores**
  - Disco + **memoria**

*Respuesta a incidentes definida por el NIST*

# Introducción

Un poco de recapitulación...



- **Análisis forense de red**
- **Análisis forense de ordenadores**
  - Disco + **memoria**

## Disco vs. memoria

- Algunas veces, **el acceso a dispositivos físicos es difícil**
- **Límites actuales de capacidad de almacenamiento vs. capacidad de la memoria**
  - Terabytes versus gibibytes
  - **Facilita el triaje inicial**
- Algunos datos sólo residen en memoria

*Respuesta a incidentes definida por el NIST*



# Introducción

## Análisis forense de memoria

### Volcado de memoria

- **Lleno de datos** para analizar
- **Cada elemento que se puede analizar se llama artefacto de memoria**
  - Recuperados vía estructuras internas del SO o búsquedas de patrones
- “Foto” de los procesos en ejecución, usuarios activos, archivos abiertos, o conexiones de red abiertas – **todo lo que estaba en ejecución en el momento de captura**
- Puede contener también **recursos liberados recientemente**
  - Normalmente, la memoria no se “pone a cero” cuando se libera

# Agenda

- 1 Introducción
- 2 Volatility**
- 3 Nuestra primera experiencia con Volatility
- 4 Nuestra experiencia actual con Volatility
- 5 Conclusiones

# Volatility



- Herramienta **estándar de facto** de análisis de memoria forense
  - Versión 2 vs. versión 3 ⇒ Python2 vs. Python3
- **Útil para examinar y extraer información de la memoria volátil**
- Compatible con amplia gama de sistemas operativos
- **Perfiles**: información sobre estructura de la memoria del sistema operativo

<https://github.com/volatilityfoundation/volatility3>

# Agenda

- 1 Introducción
- 2 Volatility
- 3 Nuestra primera experiencia con Volatility**
- 4 Nuestra experiencia actual con Volatility
- 5 Conclusiones

# Nuestra primera experiencia con Volatility

- **Necesidad de investigación en 2016**, trabajando con Miguel Martín:
  - **Hashes criptográficos vs. hashes de similitud aproximada** (e.g, MD5 vs. ssdeep)
  - $m \in \{0, 1\}$  ( $m \in \mathbb{Z}$ ) vs.  $m \in [0, 1]$  ( $m \in \mathbb{R}$ )

# Nuestra primera experiencia con Volatility

- **Necesidad de investigación en 2016**, trabajando con Miguel Martín:
  - **Hashes criptográficos vs. hashes de similitud aproximada** (e.g, MD5 vs. ssdeep)
  - $m \in \{0, 1\}$  ( $m \in \mathbb{Z}$ ) vs.  $m \in [0, 1]$  ( $m \in \mathbb{R}$ )
  - **Aplicación a procesos extraídos de memoria**
  - Más información: doi: 10.1016/j.fsidi.2021.301120

# Nuestra primera experiencia con Volatility

- **Necesidad de investigación en 2016**, trabajando con Miguel Martín:
  - **Hashes criptográficos vs. hashes de similitud aproximada** (e.g, MD5 vs. ssdeep)
  - $m \in \{0, 1\}$  ( $m \in \mathbb{Z}$ ) vs.  $m \in [0, 1]$  ( $m \in \mathbb{R}$ )
  - **Aplicación a procesos extraídos de memoria**
  - Más información: doi: 10.1016/j.fsidi.2021.301120
- **Plugin ProcessFuzzyHash**, desarrollado por Iñaki Abadía Osta (TFG GII, curso 2016/2017; actualmente *Senior Engineer* en Arm)
  - Falta de documentación sobre cómo desarrollar plugins
  - <https://webdiis.unizar.es/~ricardo/files/TFGs/FuzzyHashingProcesos.pdf>
  - <https://github.com/reverseame/processfuzzyhash>

# Nuestra primera experiencia con Volatility

- Participación con ProcessFuzzyHash en *2017 Volatility Plugin Contest*
  - <https://volatility-labs.blogspot.com/2017/11/results-from-5th-annual-2017-volatility.html>



# Nuestra primera experiencia con Volatility

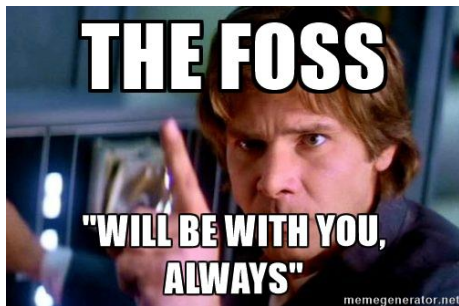
- Participación con ProcessFuzzyHash en *2017 Volatility Plugin Contest*

- <https://volatility-labs.blogspot.com/2017/11/results-from-5th-annual-2017-volatility.html>
- No ganamos 😞
- *Big kudos to Xabier!*

- **Resultado:** plugin añadido a repositorio oficial

- <https://github.com/volatilityfoundation/community/tree/master/ProcessFuzzyHash>

# Nuestra primera experiencia con Volatility



- Manera diferente de participar en comunidad FOSS: concurso anual
  - Extensión independiente de las capacidades de la herramienta “madre”
  - Incorporación al repositorio. **Contras:** mantenimiento
  - Detección de errores y posibles soluciones a la herramienta (durante el uso)

# Agenda

- 1 Introducción
- 2 Volatility
- 3 Nuestra primera experiencia con Volatility
- 4 Nuestra experiencia actual con Volatility**
- 5 Conclusiones

# Nuestra experiencia actual con Volatility

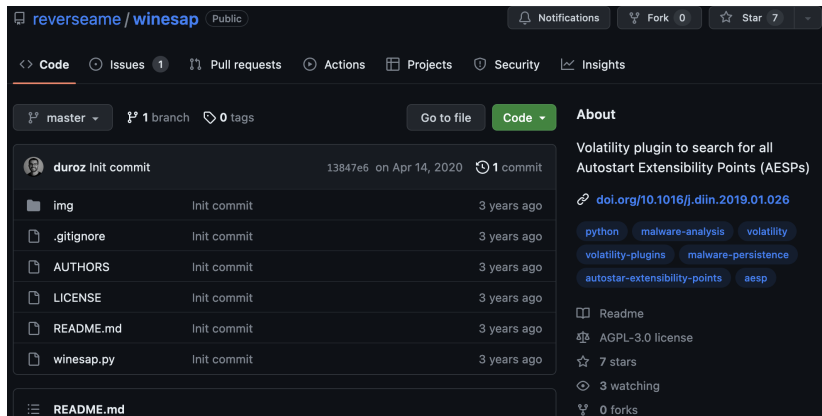
- Actualmente, ya no participamos en los concursos... (mucha burocracia)
- Pero **seguimos liberando todo bajo GNU/GPLv3**
  - *Parte de nuestro compromiso ético con la ciencia abierta y con proyectos FOSS*



Créditos: <https://www.flickr.com/photos/opensourceway/4812651268>

# Nuestra experiencia actual con Volatility

## Herramienta winesap



reverseame / winesap Public

Notifications Fork 0 Star 7

Code Issues 1 Pull requests Actions Projects Security Insights

master 1 branch 0 tags Go to file Code

**About**

Volatility plugin to search for all Autostart Extensibility Points (AESPs)

[doi.org/10.1016/j.diin.2019.01.026](https://doi.org/10.1016/j.diin.2019.01.026)

python malware-analysis volatility volatility-plugins malware-persistence autostar-extensibility-points aesp

Readme AGPL-3.0 license 7 stars 3 watching 0 forks

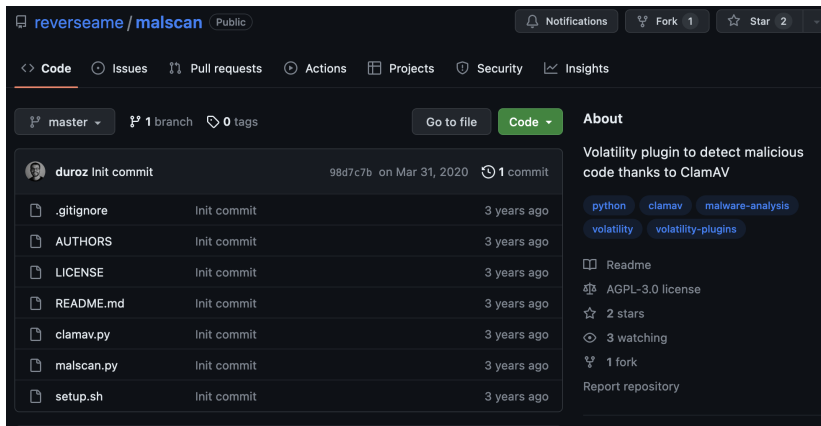
duroz	Init commit	13847e6 on Apr 14, 2020	1 commit
img	Init commit		3 years ago
.gitignore	Init commit		3 years ago
AUTHORS	Init commit		3 years ago
LICENSE	Init commit		3 years ago
README.md	Init commit		3 years ago
winesap.py	Init commit		3 years ago

README.md

<https://github.com/reverseame/winesap>

# Nuestra experiencia actual con Volatility

## Herramienta malscan



reverseame / malscan Public

Notifications Fork 1 Star 2

Code Issues Pull requests Actions Projects Security Insights









master 1 branch 0 tags Go to file Code

**About**

Volatility plugin to detect malicious code thanks to ClamAV

python clamav malware-analysis  
volatility volatility-plugins

Readme  
AGPL-3.0 license  
2 stars  
3 watching  
1 fork  
Report repository

 duroz	Init commit	98d7c7b on Mar 31, 2020	1 commit
 .gitignore	Init commit		3 years ago
 AUTHORS	Init commit		3 years ago
 LICENSE	Init commit		3 years ago
 README.md	Init commit		3 years ago
 clamav.py	Init commit		3 years ago
 malscan.py	Init commit		3 years ago
 setup.sh	Init commit		3 years ago

<https://github.com/reverseame/malscan>

# Nuestra experiencia actual con Volatility

## Herramienta similarity-unrelocated-module

The screenshot shows the GitHub repository page for 'reversease/similarity-unrelocated-module'. The repository is public and has 2 stars and 0 forks. The main content area displays a list of files and their commit history:

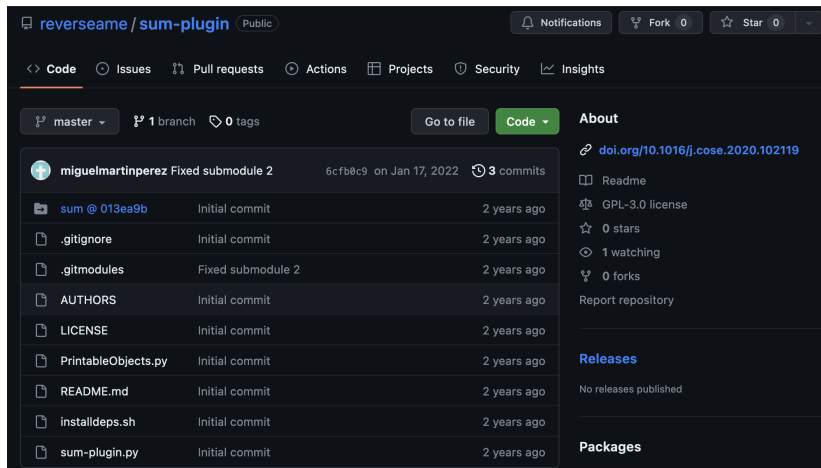
File	Commit Message	Time
marked_pefile @ 7aa...	Upgraded README and module dependency ...	2 years ago
.gitignore	SUM initial	3 years ago
.gitmodules	Installation fixed	2 years ago
AUTHORS	fix typos	3 years ago
LICENSE	Added AUTHORS, LICENSE, README	3 years ago
README.md	Upgraded README and module dependency ...	2 years ago
__init__.py	SUM initial	3 years ago
derelocation.py	Bug fixed: infinite loop when the last two byt...	2 years ago
hashengine.py	Installation fixed	2 years ago
installdeps.sh	Fixed submodule clone 3	2 years ago
pe_section.py	SUM initial	3 years ago
peobject.py	SUM initial	3 years ago
print_object.py	SUM initial	3 years ago
sum.py	Bug fixes: 'valid_pages' not defined and tish...	2 years ago

The right sidebar contains the 'About' section, which describes the repository as a 'Volatility plugin to yield and compare similarity digest of modules on execution'. It includes a DOI link (doi.org/10.1016/j.cose.2020.102119) and a list of related tags: python, sum, volatility, memory-forensics, volatility-plugins, approximate-matching, fuzzy-hash, and similarity-digest. Below the tags, there are statistics: 2 stars, 3 watching, 0 forks, and a GPL-3.0 license. The 'Releases' section indicates that no releases have been published.

<https://github.com/reversease/similarity-unrelocated-module>

# Nuestra experiencia actual con Volatility

## Herramienta sum-plugin



The screenshot shows the GitHub repository page for `reversease/sum-plugin`. The repository is public and has 0 notifications, 0 forks, and 0 stars. The main navigation bar includes links for Code, Issues, Pull requests, Actions, Projects, Security, and Insights. The repository is currently on the `master` branch, with 1 branch and 0 tags. A file list shows the following files and their commit history:

File	Commit	Time
<code>sum @ 013ea9b</code>	Initial commit	2 years ago
<code>.gitignore</code>	Initial commit	2 years ago
<code>.gitmodules</code>	Fixed submodule 2	2 years ago
<code>AUTHORS</code>	Initial commit	2 years ago
<code>LICENSE</code>	Initial commit	2 years ago
<code>PrintableObjects.py</code>	Initial commit	2 years ago
<code>README.md</code>	Initial commit	2 years ago
<code>installdeps.sh</code>	Initial commit	2 years ago
<code>sum-plugin.py</code>	Initial commit	2 years ago

The right sidebar contains the following information:

- About**: [doi.org/10.1016/j.cose.2020.102119](https://doi.org/10.1016/j.cose.2020.102119)
- Readme
- GPL-3.0 license
- 0 stars
- 1 watching
- 0 forks
- Report repository
- Releases**: No releases published
- Packages**

<https://github.com/reversease/sum-plugin>



# Nuestra experiencia actual con Volatility

## Herramienta sigcheck

reverseame / sigcheck Public

Notifications Fork 4 Star 15

Code Issues 1 Pull requests Actions Projects Security Insights

master 1 branch 0 tags Go to file Code

About

Volatility plugin to validate Authenticode-signed processes, either with embedded signature or catalog-signed

[doi.org/10.1016/j.fsdi.2020.300917](https://doi.org/10.1016/j.fsdi.2020.300917)

python openssl authenticode volatility sigcheck volatility-plugins

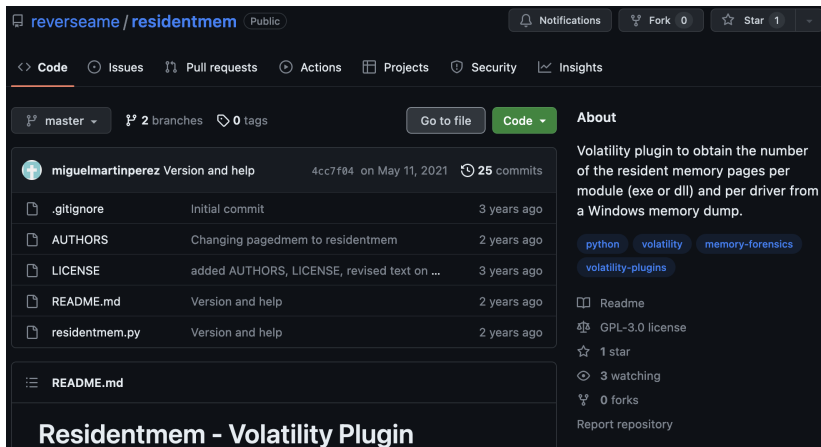
Readme GPL-3.0 license 15 stars 5 watching 4 forks

File	Commit Message	Time
.gitignore	Init commit	3 years ago
AUTHORS	Authors added	3 years ago
LICENSE	Authors added	3 years ago
README.md	catroot test included in README.md and del...	3 years ago
addresses.json	Init commit	3 years ago
setup.sh	Init commit	3 years ago
sigcheck.py	Fixes #2. Now, sigcheck relies in the user op...	3 years ago
sigvalidator.py	catroot test included in README.md and del...	3 years ago

<https://github.com/reverseame/sigcheck>

# Nuestra experiencia actual con Volatility

## Herramienta residentmem



The screenshot shows the GitHub repository page for 'residentmem' by 'reverseame'. The repository is public and has 0 forks and 1 star. The main branch is 'master'. The repository description is: 'Volatility plugin to obtain the number of the resident memory pages per module (exe or dll) and per driver from a Windows memory dump.' The repository includes tags for 'python', 'volatility', 'memory-forensics', and 'volatility-plugins'. The license is GPL-3.0. There are 1 star, 3 watching, and 0 forks. The repository is reported as a repository.

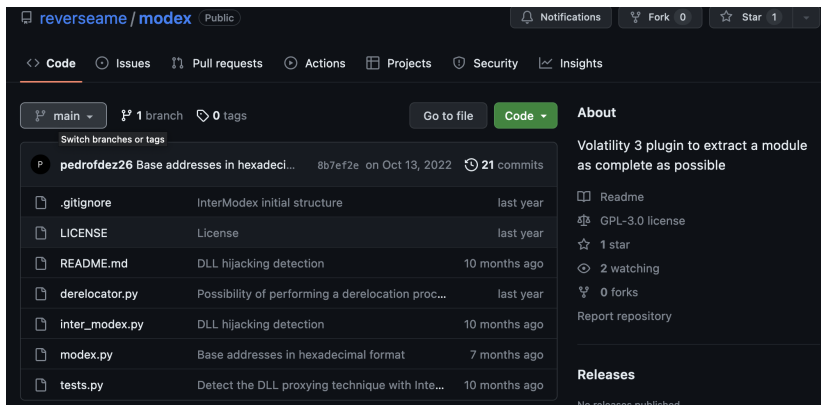
File	Commit Message	Time Ago
.gitignore	Initial commit	3 years ago
AUTHORS	Changing pagedmem to residentmem	2 years ago
LICENSE	added AUTHORS, LICENSE, revised text on ...	3 years ago
README.md	Version and help	2 years ago
residentmem.py	Version and help	2 years ago

**Residentmem - Volatility Plugin**

<https://github.com/reverseame/residentmem>

# Nuestra experiencia actual con Volatility

## Herramienta modex



The screenshot shows the GitHub repository page for `reverseseam/modex`. The repository is public and contains the following files:

File Name	Description	Last Commit
<code>.gitignore</code>	InterModex initial structure	last year
<code>LICENSE</code>	License	last year
<code>README.md</code>	DLL hijacking detection	10 months ago
<code>derelocator.py</code>	Possibility of performing a derelocation proc...	last year
<code>inter_modex.py</code>	DLL hijacking detection	10 months ago
<code>modex.py</code>	Base addresses in hexadecimal format	7 months ago
<code>tests.py</code>	Detect the DLL proxying technique with Inte...	10 months ago

The repository also has 1 star and 0 forks. The description of the repository is: "Volatility 3 plugin to extract a module as complete as possible".

<https://github.com/reverseseam/modex>

# Agenda

- 1 Introducción
- 2 Volatility
- 3 Nuestra primera experiencia con Volatility
- 4 Nuestra experiencia actual con Volatility
- 5 Conclusiones**

# Conclusiones

## ■ +6 plugins desarrollados para Volatility

- Buena acogida de la comunidad (científica, sobre todo)
- Contacto con Volexity (<https://www.volexity.com/>)

# Conclusiones

## ■ +6 plugins desarrollados para Volatility

- Buena acogida de la comunidad (científica, sobre todo)
- Contacto con Volexity (<https://www.volexity.com/>)

## ■ Otras soluciones FOSS desarrolladas:

- rop3: <https://github.com/reverseame/rop3>
- Chitón: <https://github.com/reverseame/chiton>
- Windows Memory Extractor:  
<https://github.com/reverseame/windows-memory-extractor>
- IM Artifact Finder:  
<https://github.com/reverseame/instant-messaging-artifact-finder>
- EvalMe: <https://github.com/reverseame/EvalMe>
- Secure\_Socket: [https://github.com/reverseame/Secure\\_Socket](https://github.com/reverseame/Secure_Socket)
- pinVMShield: <https://github.com/reverseame/pinVMShield>

## ■ Licencias GNU/GPL3.0 y GNU/AGPL3.0

# Conclusiones

## Desventajas

- Modelo de “colaboración” con FOSS actual **poco reactivo**
- **Poca interacción de/desde la comunidad** (más allá de *stars/watchings*)
- Muchas herramientas derivadas de trabajos de investigación
- **Dificultad de gestionar issues** (al final, no somos una empresa desarrolladora)

## Ventajas

- **Permite posicionarnos** (*conocen qué sabemos hacer*)
- **Posibilidad de amplia difusión y mejora** (por la comunidad)
- **Modelo de negocio basado en adaptaciones para empresas**

# Experiencias propias en la contribución a proyectos de software libre: Volatility

**Ricardo J. Rodríguez**

© All wrongs reversed – bajo licencia CC BY-NC-SA 4.0



**Universidad**  
Zaragoza

Dept. de Informática e Ingeniería de Sistemas  
Universidad de Zaragoza

13 de mayo, 2023

**esLibre**

Zaragoza, España

