



Conociendo Ansible

Pablo
Martínez
Schroder



¿Qué es Ansible?



¿Qué es Ansible?

Un sistema de gestión de configuraciones.

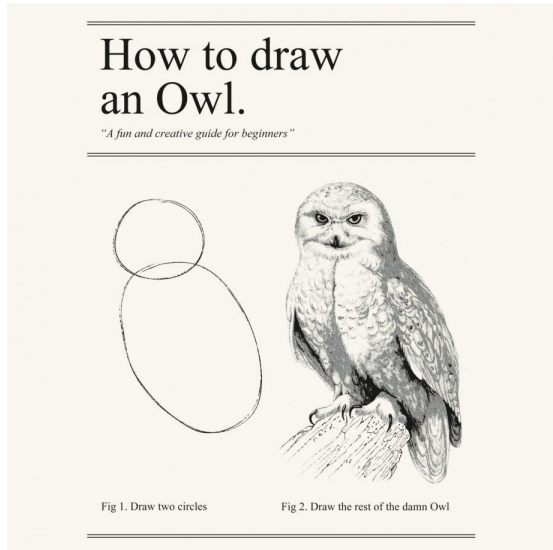
Más o menos.

¿Qué es Ansible?

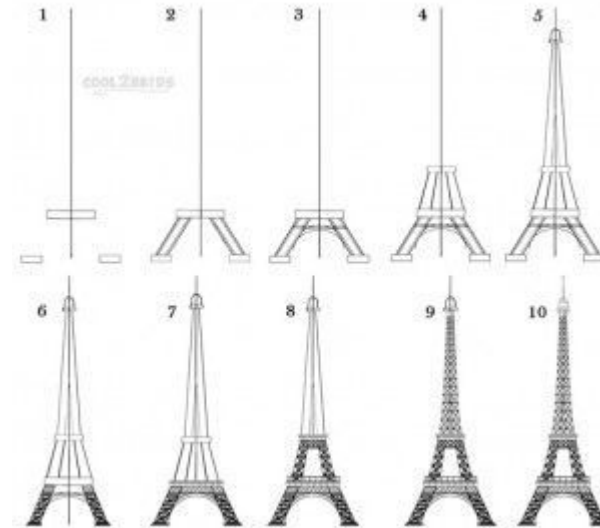
A diferencia de otros gestores de configuración en Ansible se dan una serie de pasos, no un estado deseado.

¿Qué es Ansible?

En otros C.M. describes el estado deseado y el sistema lo configura.



En Ansible se especifica cada paso a dar y así llegas al estado deseado.



Introducción de 5 minutos



Introducción de 5 minutos

Software libre.

No necesita ningún agente especial en el sistema remoto.

Usa SSH para controlar los sistemas.

Fue comprado por Red Hat.

<https://www.ansible.com/>

<https://github.com/ansible/ansible>

Introducción de 5 minutos

Acaba de publicar la versión 2.2.

Escrito en Python.

Importante comunidad de usuarios.

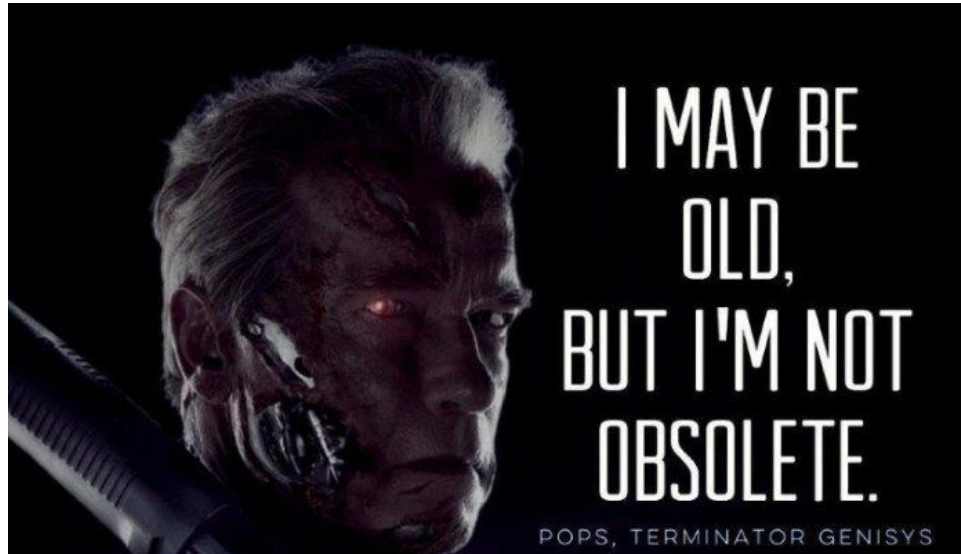
Multiplataforma.

<https://www.ansible.com/>

<https://github.com/ansible/ansible>

Introducción de 5 minutos

¡NO USEIS ANSIBLE 1.x!



Introducción de 5 minutos

¡NO USEIS ANSIBLE 1.x!

YES, YOU ARE OBSOLETE!



Introducción de 5 minutos

¡NO USEIS ANSIBLE 1.x!

Es vieja y obsoleta.

¡USAD ANSIBLE 2.x!

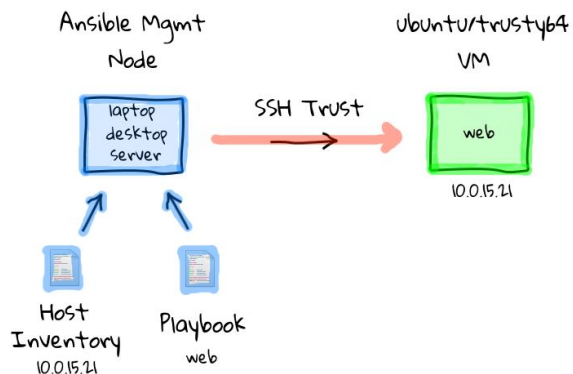
(evitad 2.1.x)

(2.2.0 es reciente, tiene unas semanas ¡intentad migrad a ella)

Introducción de 5 minutos

Un inventario es una lista de sistemas (*hosts*) agrupadas en distintos grupos.

Un playbook un conjunto de acciones (*tasks*) que aplicar a una serie de hosts.



Introducción de 5 minutos

Un inventario puede ser un fichero tipo .ini

```
$ cat inventory.txt  
mail.example.com
```

```
[webservers]  
foo.example.com  
bar.example.com
```

```
[dbservers]  
one.example.com  
two.example.com  
three.example.com
```

Introducción de 5 minutos

Un playbook describe una serie de acciones

```
$ cat install-server.yml
```

```
---
```

```
- hosts: webservers
```

```
vars:
```

```
  http_port: 80
```

```
  max_clients: 200
```

```
remote_user: root
```

```
tasks:
```

```
- name: ensure apache is at the latest version
```

```
  yum: name=httpd state=latest
```

```
- name: write the apache config file
```

```
  template: src=/srv/httpd.j2 dest=/etc/httpd.conf
```

```
  notify:
```

```
- restart apache
```

```
- name: ensure apache is running (and enable it at boot)
```

```
  service: name=httpd state=started enabled=yes
```

```
handlers:
```

```
- name: restart apache
```

```
  service: name=httpd state=restarted
```

Introducción de 5 minutos

Aplicando el playbook en el inventario, podemos adivinar qué pasará.

```
$ ansible-playbook -i inventory.txt install-server.yml
```

Introducción de 5 minutos

Aplicando el playbook en c

qué pasará.

```
$ ansible-playbook
```

```
tail-server.yml
```



Algo más dentro de Ansible



Algo más dentro de Ansible

- Inventario
- Funcionamiento
- Roles
- Módulos
 - Core
 - Extras
- Callbacks
- Facts
- Orquestación
- Recomendaciones (personales)

Algo más dentro de Ansible - Inventario

- Un inventario es un fichero tipo `ini`.
 - Existen inventarios dinámicos.
- Donde se especifican los nodos, que se pueden agrupar.
- Un nodo sólo debería estar en un grupo*
- Se pueden hacer grupos de grupos (con `:children`).
- También se pueden añadir variables en un inventario.
 - Pueden ser de nodo o de grupos.
- Es posible conectarse localmente o a docker en vez de ssh
- Inventarios dinámicos son binarios/scripts
 - Con `--list` deben mostrar un JSON con un diccionario de nodos
 - Con `--host` todas las variables de un JSON

Algo más dentro de Ansible - Funcionamiento

Ansible usa SSH (o docker!).

Leerá el fichero de configuración `/etc/ansible/ansible.cfg`, que puede modificarse con el contenido de `~/.ansible.cfg`

También puede especificarse con la variable de entorno `ANSIBLE_CONFIG`

Algo más dentro de Ansible - Funcionamiento

Añadir nodos

Si no tenemos una ssh key, tenemos que agregarla. Para ello Ansible debe conocer la host key. Pero la primera vez no lo sabe, así que la agregamos.

```
ANSIBLE_HOST_KEY_CHECKING=False ANSIBLE_CONFIG=ansible.cfg ansible  
-k web -m raw -a 'hostname' -i inventory.txt
```

Con -k forzamos a que nos pida la password.

Algo más dentro de Ansible - Funcionamiento

Añadir nodos

Se puede especificar la contraseña y el usuario como variables en el inventario.

O utilizar claves SSH.

El sistema debe ser capaz de conectarse por SSH.

Algo más dentro de Ansible - Funcionamiento

Estrategias (*strategies*)

Por defecto ansible se conectará con la estrategia ***linear***: se espera a que todos los nodos ejecuten una tarea antes de continuar con la siguiente.

Otra estrategia interesante es ***batch***, que permite correr en lotes. Los lotes pueden especificarse de distintas formas.

También se incluye una estrategia ***free*** que hace que cada nodo corra las tareas tan rápido como pueda, de forma independiente

Y por último, podemos correr con la estrategia ***debug*** que abrirá un debugger cuando una tarea falle.

Algo más dentro de Ansible - Funcionamiento

Estrategias (*strategies*)

La estrategia se especifica en el playbook:

```
- hosts: all
  strategy: free
  tasks:
```

Se pueden hacer batches con serial:

```
- hosts: webservers
  serial: 3
```


Algo más dentro de Ansible - Funcionamiento

Serial

Con la estrategia serial es posible especificar porcentaje o grupos de batches:

```
serial: "30%"
```

```
serial:
```

- 1
- 5
- 10

Tolerancia a errores

Por defecto Ansible seguirá ejecutando las tareas en todos los nodos, pero es posible especificar que se detenga el play si se supera un porcentaje:

```
- hosts: webservers
  max_fail_percentage: 30
  serial: 10
```

Forks

En el fichero de configuración puedes especificar cuántos procesos paralelos. Por defecto es 5.

Algo más dentro de Ansible - Roles

Los roles son conjuntos de tareas que se organizan de forma que pueden reutilizarse.

Permiten tener más organización con los playbooks, y pueden ser compartidos.

Algo más dentro de Ansible - Roles

Ansible Galaxy ofrece miles de nodos que puedes utilizar. Puedes instalar un nodo de un repositorio sencillamente con:

```
ansible-galaxy install username.rolename
```

Aunque la verdad es que no se suelen usar ciegamente. Puedes navegar y ver los existentes en:

<https://galaxy.ansible.com/>

Algo más dentro de Ansible - Roles

Es muy fácil crear un role

```
cd proyecto/roles
ansible-galaxy init miservicio
```

La recomendación es que vayas agrupando en roles, de esa forma podrás tener el código más separado. En el playbook los puedes invocar fácilmente:

```
hosts: webservers
roles:
  - common
  - { role: miservicio, port: 8000 }
```

Algo más dentro de Ansible - Roles

Ademas de usar Ansible Galaxy podrías tener un directorio con roles comunes.

Puedes especificar su localización con la directiva `roles_path` en el fichero de configuración.

```
[defaults]  
roles_path = /opt/ansible/common_roles/
```

Algo más dentro de Ansible - Módulos

Los módulos ofrecen las funciones que puedes ejecutar con los tasks.

Existen módulos incluidos con el sistema (*core*), los adicionales (*extras*) y es posible desarrollar módulos propios.

La lista completa de módulos está en

https://docs.ansible.com/ansible/list_of_all_modules.html

Algo más dentro de Ansible - Módulos core

Los módulos *core* incluyen las funciones más habituales como gestión de paquetes, de usuarios, de repositorios.

Algo más dentro de Ansible - Módulos extra

<https://github.com/ansible/ansible-modules-extras>

cloud	py3 - ported ec2_vpc_route_table iterkeys to dict.keys() (#3503)	3 days ago
clustering	consul: Pass through service_id if specified on a check (#3295)	14 days ago
commands	Make expect module compile on python 3	a month ago
crypto	Use native YAML in the examples (#3441)	14 days ago
database	Update version_added, update docs (#3542)	18 hours ago
files	blockinfile: Fixes #1926 by comparing a marker to a whole line i...	8 days ago
identity	lpa bugfixes (#3421)	14 days ago
infrastructure	Fix broken entitles reference in Katello module (#3136)	a month ago
messaging	rabbitmq_queue - Update requirements to show minimum versl...	3 months ago
monitoring	corrected version added	18 days ago
network	add a10_server_axapl3 module (#3239)	18 hours ago
notification	Fix bare variable references in docs	15 days ago
packaging	Making yum_repository module compatible with Python 3 (#3487)	7 days ago
remote_management	renamed bmc to remote_management to clarify	3 months ago
source_control	Respect include.* directives by default.	a month ago
storage	Fix 'netapp_e_lun_mapping ... documentation error' (#2966)	2 months ago
system	replace iteritems with items to ensure python3 compatibility	8 days ago
test/utils/shippable	Simplify compileall checks. Remove py3 skip list.	22 days ago
univention	Use addresses and names reserved for documentation (#2894)	2 months ago
web_infrastructure	letsencrypt: Locale-independent date parsing (#3314)	8 days ago
windows	Detect and correct PowerShell mishandling nssm Unicode outp...	14 days ago

Algo más dentro de Ansible - Callbacks

Los Callback plugin son un tipo de plugin que gestionan la salida y los eventos que se producen.

Si quieres usar alguno debes habilitarlos con la siguiente configuración:

```
callback_whitelist = timer, mail, json  
callback_plugins = /path/a/tus/callback/
```

Luego puedes especificar qué plugin utilizar por defecto para gestionar la salida.

```
stdout_callback = json
```

Algo más dentro de Ansible - Facts

Cada vez que corres un plugin, por defecto Ansible ejecutará un módulo llamado `setup` que recoge algunos datos, llamados facts.

Esos datos están disponibles en los playbooks como `{{ ansible_facts }}`

Si tienes instalado en el nodo remoto el program `facter` u `ohai` entonces también usará los datos que proveen esas utilidades y los almacenarán.

Por defecto sólo se almacenan durante la ejecución pero es posible almacenarlos en redis o en ficheros json.

Algo más dentro de Ansible - Facts

Por ejemplo:

```
gathering = smart # Puede ser implicit, explicit o smart
fact_caching = jsonfile
fact_caching_connection = facts/
```

Se puede deshabilitar en un Playbook con `gather_facts`:

```
---
- hosts: all
  gather_facts: False
  tasks:
    - name: Hello
      shell: "echo hello"
```

Algo más dentro de Ansible - Orquestación

Ansible, por su naturaleza más imperativa, es bastante útil para gestionar la orquestación. Incluye unos módulos que permiten gestionar ejecución en distintos nodos.

`delegate_to` (o `local_action`), `wait_for` o `run_once` son herramientas que pueden hacernos más fácil la orquestación.

Mención también para `async`, aunque no entraremos en detalle.

https://docs.ansible.com/ansible/playbooks_async.html

Algo más dentro de Ansible - Orquestación

delegate_to

```
hosts: webservers
```

```
  serial: 5
```

```
tasks:
```

- name: take out of load balancer pool
 command: /usr/bin/take_out_of_pool {{ inventory_hostname }}
 delegate_to: 127.0.0.1

- name: actual steps would go here
 yum: name=acme-web-stack state=latest

- name: add back to load balancer pool
 command: /usr/bin/add_back_to_pool {{ inventory_hostname }}
 delegate_to: 127.0.0.1

Algo más dentro de Ansible - Orquestación

wait_for

```
# Esperar a que el puerto 8000 esté libre
- wait_for: port=8000 delay=10 # Esperamos a que esté respondiendo antes de seguir
- wait_for: port=8000 state=stopped # Después arrancamos nuestro servidor
- wait_for: host=0.0.0.0 port=8000 delay=10 state=drained # Todas las conexiones cerradas
- wait_for: host=0.0.0.0 port=8000 state=drained exclude_hosts=10.2.1.2,10.2.1.3

- wait_for: path=/var/run/server.pid
- wait_for: path=/var/log/server.log search_regex="Initialization completed"
- wait_for: path=/var/lock/server.lock state=absent
- wait_for: path=/proc/3466/status state=absent

# En lugar de delegate_to podemos usar local_action para esperar a que el nodo esté arriba
- local_action: wait_for port=22 host="{{ ansible_ssh_host | default(inventory_hostname) }}"
  search_regex=OpenSSH delay=10
```

Algo más dentro de Ansible - Orquestación

run_once

```
# Sólo correrlo en uno de los servidores
- command: /opt/server/scripts/upgrade_db.py
  run_once: true
```

Esto viene a ser igual que

```
- command: /opt/server/scripts/upgrade_db.py
  delegate_to: web1

- command: /opt/server/scripts/upgrade_db.py
  when: inventory_hostname == 'web[0]'
```

Otra opción interesante de command es “creates”:

```
- command: yum update -y kernel creates=/tmp/kernel_upgraded
```

Algo más dentro de Ansible - Recomendaciones

Unas recomendaciones personales. Muy discutibles, por supuesto.

- Define tus servidores en “proyectos”.
- Usa un fichero de configuración por proyecto.
- Ten un directorio de roles común para todos tus proyectos.
- Utiliza un `known_hosts` por proyecto.
- Si es posible, usa inventarios dinámicos.

Algo más dentro de Ansible - Recomendaciones

Se reflejaría en algo como

```
-- proyecto/  
+- ansible.cfg  
+- playbook-1.yml  
+- .known_hosts  
+- inventory/  
+- facts/  
+- group_vars/  
|   +- all  
+- roles/  
    +-{vars,handlers,defaults,tasks,meta,templates}
```

Algo más dentro de Ansible - Recomendaciones

ansible.cfg

```
[defaults]
remote_tmp = /tmp/tmp
roles_path = /path/a/los/roles/compartidos
callback_plugin = /path/a/los/roles/compartidos
library = /parth/a/los/modulos/compartidos
gathering = smart
fact_caching = jsonfile
fact_caching_connection = facts
inventory = inventory/default.txt # or inventory si es dinámico

[ssh_connection]
scp_if_ssh = True
ssh_args = -o ConnectTimeout=60 -o UseRoaming=no -o UserKnownHostsFile=.known_hosts
```

Ecosistema de Ansible



Ecosistema de Ansible

Ansible Galaxy

Versión de software libre del software que gestiona Ansible Galaxy. Puede servir para que un grupo comparta de forma limitada los roles.

<https://github.com/ansible/galaxy>

Ecosistema de Ansible

Ansible Tower

Herramienta comercial para gestionar Ansible.

<https://www.ansible.com/tower>

Ecosistema de Ansible

Ansible ARA

Herramienta open source que almacena todas las ejecuciones y sus salidas en una base de datos, de forma que puede ser accedido por una GUI

<https://github.com/dmsimard/ara>

Ecosistema de Ansible

Ansible ARA

The screenshot displays the Ansible ARA web interface for a specific playbook run. The interface includes a sidebar with navigation options (Home, All playbooks, Last playbooks) and a main content area with the following sections:

- Playbook:** /var/www/ara/tests/integration/hosts.yml
- Run Time:** 2016-07-04 15:36:20 » 2016-07-04 15:36:24
- Summary:** 4 Hosts, 2 Plays, 5 Tasks, 1 Files, 0:00:04 Duration.
- Statistics:** A table showing the status of tasks for each host.
- Plays:** A table showing the execution of individual plays.
- Tasks:** A detailed table of tasks, including their offset from the playbook, duration, name, action, file, and line number.

Host	OK	CHANGED	FAILED	SKIPPED	UNREACHABLE
host1	3	0	0	1	0
host2	3	0	0	1	0
host3	2	0	0	2	0
localhost	3	1	0	2	0

Name	Start	End	Duration	
Create fake hosts for host tests	2016-07-04 15:36:20	2016-07-04 15:36:20	0:00:00	details
ARA Hosts test play	2016-07-04 15:36:20	2016-07-04 15:36:24	0:00:03	details

Offset from Playbook	Duration	Name	Action	File	Line	
0:00:00	0:00:00		add_host	...tests/integration/hosts.yml	21	details
0:00:00	0:00:01		setup			details
0:00:02	0:00:00	Ping	ping	...tests/integration/hosts.yml	34	details
0:00:03	0:00:00	Do something only on host1	set_fact	...tests/integration/hosts.yml	37	details
0:00:03	0:00:00	Fail something only on host2	command	...tests/integration/hosts.yml	42	details

Ecosistema de Ansible

Ansistrano

Ansistrano es un port de Capistrano para Ansible. Está disponible en Ansible Galaxy y consiste en 2 nodos en los que mediante variables configuras la instalación, actualización y todo lo relacionado con el despliegue de software.

<http://ansistrano.com/>

Ecosistema de Ansible

Gestores de tareas

Rundeck y Gunnery son herramientas de software libre para gestionar la gestión de tareas. Pueden utilizarse como alternativas a Ansible Tower.

<http://rundeck.org/>

<http://gunnery.io/>

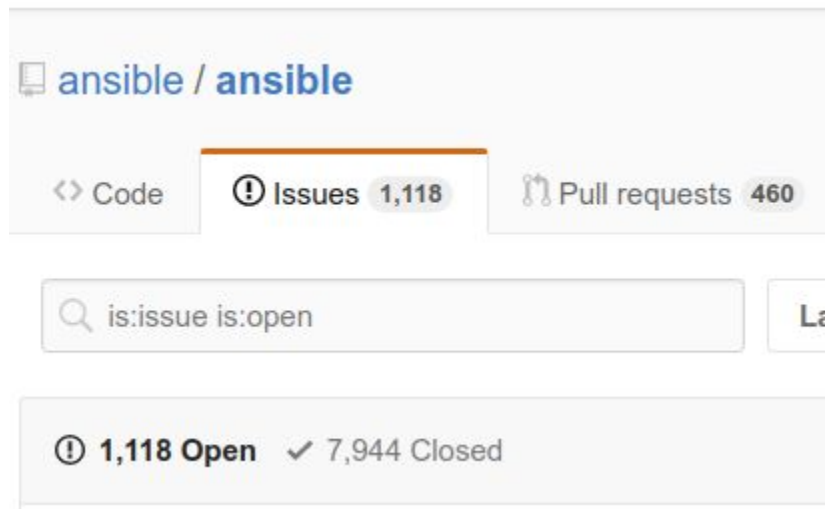
No todo es oro



Bart Simpson después de gestionar unos playbooks complejos con una nueva versión de Ansible.

No todo es oro

Más de 1.100 issues abiertas.



The screenshot shows the GitHub repository page for 'ansible / ansible'. The repository name is displayed at the top. Below it, there are three navigation tabs: 'Code', 'Issues 1,118', and 'Pull requests 460'. The 'Issues' tab is currently selected and highlighted with an orange underline. Below the navigation tabs, there is a search bar containing the text 'is:issue is:open' and a button labeled 'La'. At the bottom of the screenshot, there is a summary bar showing '1,118 Open' with a warning icon and '7,944 Closed' with a checkmark icon.

ansible / ansible

<> Code ⓘ Issues 1,118 🔄 Pull requests 460

🔍 is:issue is:open La

ⓘ 1,118 Open ✓ 7,944 Closed

No todo es oro

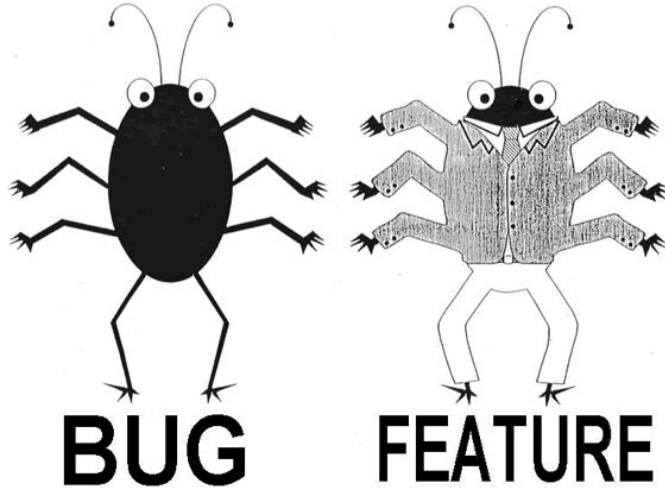
La documentación es muy mejorable.

Hay mucha documentación, pero es algo desordenada e inconsistente.

Los módulos tiene documentación inconsistente y no siempre actualizada.

No todo es oro

Algunos comportamientos son “poco prácticos”.



¿Preguntas?



¡Gracias!

Y recordad: donar sangre salva vidas: <http://www.donantesmalaga.org/>