

TEUTON: Test de infraestructura

Fecha : Viernes 24
Lugar : Sala Núñez
Hora : 10:30-10:55



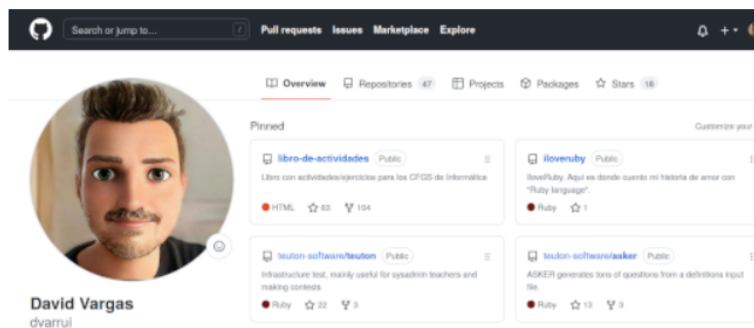
2. Agradecimientos

- Organización **EsLibre22**
- Equipo Teuton Software
- Comunidad Teuton de Telegram

Todo lo que vamos a ver está hecho con Software Libre

3. Presentación

- David Vargas
- GitHub: **@dvarrui**
- Ruby, software libre y StarWars.



4. Teuton

¿Qué es?

- Programa multiplataforma.
- Licencia Software libre.
- Test de infraestructura.

¿Qué resuelve?

1. Revisar nuestra infraestructura como si fuera código.
2. Automatizar las correcciones de las MV remotas de los alumnos.

5. TDD

- **Desarrollo guiado por pruebas de software.**
- Lograr código limpio que funcione.
- **Garantizar** que el software **cumple con los requisitos** que se han establecido.

=> "Clean Code" Robert C. Martin
=> "Código sostenible" Carlos Ble

6. Ejemplo: Factorizar

```
module MyMath
  def self.factorize(number)
    factors = []
    divisor = 2
    while(number > 1)
      while (number % divisor).zero?
        factors << divisor
        number /= divisor
      end
      divisor += 1
    end
    return factors
  end
end
```

```
module MyMath
  def self.factorize(number)
    factors = []
    divisor = 2
    while(number > 1)
      while (number % divisor).zero?
        factors << divisor
        number /= divisor
      end
      divisor += 1
    end
    return factors
  end
end
```

7. Ejemplo: test_factorize

```
class TestFactor < Test::Unit::TestCase
  def test_factorize_34
    assert_equal [2, 17], MyMath.factorize(34)
  end
end
```

```
require 'test/unit'
require_relative 'mymath.rb'

class TestFactor < Test::Unit::TestCase
  def test_factorize_34
    assert_equal [2, 17], MyMath.factorize(34)
  end
end
```

8. Concurso de programación

[ProgramaMe](#) [Información general](#) [Problemas](#) [Regionales](#) [Nacional](#) [Ediciones anteriores](#) [Contacto](#)

 ProgramaMe: Concurso de Programación para Ciclos Formativos

Estás aquí: / [Nacional 2022](#) / Inicio



Concurso Nacional

Organizado desde la [Facultad de Informática de la UCM](#)

28 de abril de 2022

[Inicio](#)
[Ayuda a Ucrania](#)
[Información para equipos](#)
[Información para centros](#)
[Inscripción](#)
[Reglamento](#)

ProgramaMe en tiempos de pandemia

Ponemos en marcha la final de ProgramaMe con dos elementos a nivel mundial que nos obligan a adaptarnos a las circunstancias:

- *Pandemia de la COVID-19*: la realización de actividades presenciales multitudinarias sigue sin estar completamente despejada, por lo que, como [el curso pasado](#), esta edición tendrá el modelo de *final online multisitio*.
- *Invasión rusa a Ucrania*: el 24 de febrero fuerzas terrestres rusas entraron en Ucrania y desde entonces se vive una crisis humanitaria. Desde ProgramaMe queremos ayudar [Acnur](#), la agencia de la ONU para los refugiados, a [recaudar fondos](#).



[Prueba tus soluciones](#)

Puedes probar tus soluciones en [¡Acepta el reto!](#), el juez on-line con [problemas de ediciones anteriores](#).

[Programame 2.0](#)

Visita nuestra [blog](#) con

9. QA

Aplicar **tests para mantener la calidad** de...

- Mi código.
- ¿Mi infraestructura?
- ¿Mis scripts?

https://rubygems.org/gems/teuton

Buscar gemas...

RELEASESBLOGGEMSGUÍASINGRESARREG

teuton 2.3.6

Intrastructure test, useful for: (1) Sysadmin teachers to evaluate students remote machines. (2) Sysadmin apprentices to evaluate their learning process as a game. (3) Professional sysadmin to monitor remote machines. Allow us: (a) Write test units for real or virtual machines using simple DSL. (b) Check compliance with requirements on remote machines.

VERSIONES:

2.3.6 - June 22, 2022 (71 KB)

2.3.5 - June 22, 2022 (71 KB)

2.3.4 - June 21, 2022 (71 KB)

2.3.3 - June 21, 2022 (71 KB)

2.3.2 - June 21, 2022 (71,5 KB)

Mostrar todas las versiones (22 total) →

DEPENDENCIAS DE RUNTIME (9):

colorize ~> 0.8.1

json_pure ~> 2.2

net-sftp ~> 2.1

net-ssh ~> 5.0

net-telnet ~> 0.1

os ~> 1.1

rainbow ~> 3.0

Star22

TOTAL DE DESCARGAS

14.802

PARA ESTA VERSIÓN

96

GEMFILE:

gem 'teuton', '~> 2.3.6'

INSTALAR:

gem install teuton

help? contents? - copyright not found -

slide 9/19

10. Instalar Teuton

- Instalar Ruby
- `gem install teuton`, instalar la gema

```
> teuton
Commands:
  teuton [run] [OPTIONS] DIRECTORY # Run test from directory
  teuton check [OPTIONS] DIRECTORY # Check test and config file content
  teuton help [COMMAND]           # Describe available commands or one specific command
  teuton new DIRECTORY            # Create skeleton for a new project
  teuton readme DIRECTORY         # Show README extracted from test contents
  teuton version                  # Show the program version
```

11. Test local



```
# File: localtest/start.rb
group "Local test" do
  target "Existe el usuario Obiwan"
  run "id Obiwan"
  expect "Obiwan"
end
```

Para ejecutar el test hacemos:

```
teuton localtest
```

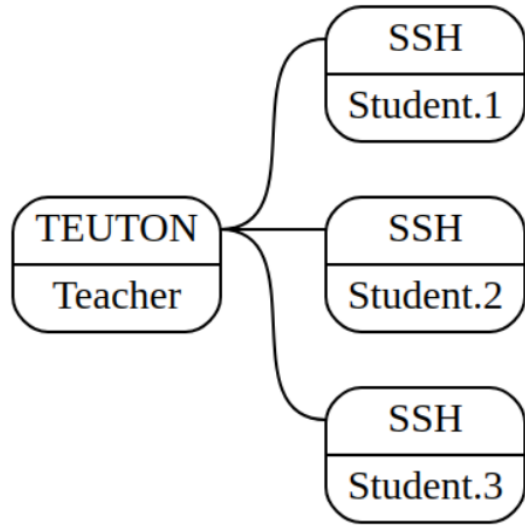
12. DSL sencillo

- **target:** Descripción del objetivo
- **run:** Comando que ejecutamos para comprobar.
- **expect:** Lo que esperamos “ver” en la salida del comando anterior.

```
1) Si sabes qué comando usar para comprobar el target...  
2) Y sabes que debes "mirar" en la salida del comando...  
¡Ya tienes el test!
```

<https://github.com/dvarrui/teuton-tests>

13. Equipos remotos



- Un host con TEUTON instalado.
- El resto con SSH para poder acceder.

14. Test remoto

¿Existe el usuario “Obiwan” en los equipos remotos?



```
# File: reomotetest/start.rb
group "Remote test" do
  target "Existe el usuario Obiwan"
  run "id Obiwan", on: :host1
  expect "Obiwan"
end
```

Para ejecutar el test hacemos:

```
teuton remotetest
```

15. Configuración de los hosts

```
# File: remotetest/config.yaml
---
global:
  host1_username: root
cases:
- tt_members: Student-1
  host1_ip: 192.168.1.101
  host1_password: secret1
- tt_members: Student-2
  host1_ip: 192.168.1.102
  host1_password: secret2
- tt_members: Student-3
  host1_ip: 192.168.1.103
  host1_password: secret3
```

```
# File: remotetest/config.yaml
---
global:
  host1_username: root
cases:
- tt_members: Student-1
  host1_ip: 192.168.1.101
  host1_password: secret1
- tt_members: Student-2
  host1_ip: 192.168.1.102
  host1_password: secret2
- tt_members: Student-3
  host1_ip: 192.168.1.103
  host1_password: secret3
```

16. Generar informes

```
# File: remotetest/start.rb
group "Remote test" do
  target "Existe el usuario Obiwan"
  run "id Obiwan", on: :host1
  expect "Obiwan"
end

play do
  show
  export format: :html
  send copy_to: :host1
end
```

```
play do
  show
  export format: :html
  send copy_to: :host1
end
```


17. Expect

```
expect 'obiwan', expect line/s with "obiwan".
expect ['obiwan', 'kenobi', expect line/s with "obiwan" and kenobi".
expect_one 'obiwan', expect only one line with "obiwan".
expect_one ['obiwan','kenobi'], expect only one line with "obiwan" and "kenobi".
expect_none 'obiwan', expect no line with "obiwan".
expect_none ['obiwan', 'kenobi'], expect no line with "obiwan" and "kenobi".
expect /Obiwan|obi-wan/, Expect line/s with Obiwan or obi-wan. This example uses regular expressions.
```

```
target 'Exist user vader'
run     'cat /etc/passwd'
```

```
expect result.find("vader").count.eq(1), expect there exists only 1 line with "vader" text.
expect result.find(/Darth|darth/).find(/Vader|vader/).count.gt(2), expect there exists more than 2 lines with "Darth" or "darth" text.
expect result.not_find('#').find('vader').count.lt(3), expects there exists less than 3 lines with text "vader".
```

18. URLs

- GitHub y docs: <https://github.com/teuton-software/teuton>
- Rubygems: <https://rubygems.org/gems/teuton>
- Ejemplos: <https://github.com/dvarrui/teuton-tests>

19. Muchas Gracias

- **Email:** teuton.software@protonmail.com
- **Telegram:** “TEUTON Software”
- **GitHub:** @teuton-software, @dvarrui