



Test de infraestructura “Teuton”



Agradecimientos

AgileCanarias, comunidad Teuton, Software Libre.



Presentación

David Vargas (@dvarrui)

Ruby, OpenSUSE, software libre y Starwars

Concurso de programación - PROGRAMAME

[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](#)

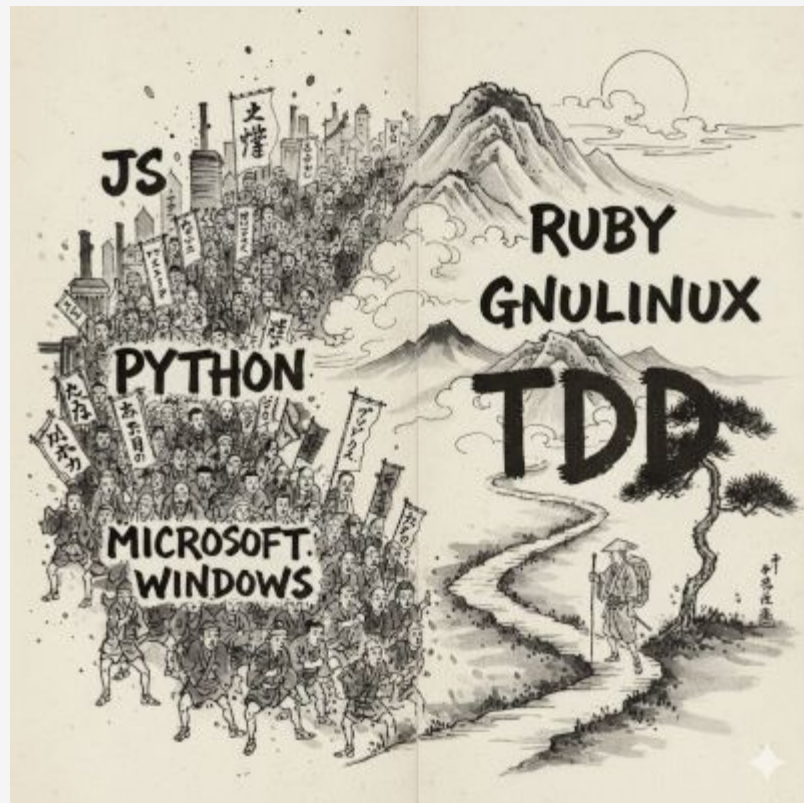
Visita nuestro [blog](#) con

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 limpio” (Carlos Ble)



Code testing - Example 1 - Recursive

```
def factorial(n)
  return 1 if n == 0 || n == 1
  n * factorial(n - 1)
end
```

```
require 'test/unit'
require_relative 'factorial_recursive'

class TestFactorial < Test::Unit::TestCase
  def test_base_case
    assert_equal(1, factorial(0))
    assert_equal(1, factorial(1))
  end

  def test_2_3_4
    assert_equal(2, factorial(2))
    assert_equal(6, factorial(3))
    assert_equal(24, factorial(4))
  end
end
```

Code testing - Example 2 - Object methods

```
def factorial(n) = (1..n).to_a.reduce(:*)
```

```
require 'test/unit'
require_relative 'factorial_recursive'

class TestFactorial < Test::Unit::TestCase
  def test_base_case
    # assert_equal(1, factorial(0))
    assert_equal(1, factorial(1))
  end

  def test_2_3_4
    assert_equal(2, factorial(2))
    assert_equal(6, factorial(3))
    assert_equal(24, factorial(4))
  end
end
```

Principios

- TDD (Test-Driven Development)
- DRY (Don't repeat yourself)
- DevOps (Automatización de procesos)

Problema

- Reducir los tiempos.
- Mejorar la calidad.
- Falta de automatización.
- Repetición (aburrido) VS Gamificación (divertido)

Teuton: Caso de uso



¿Qué es Teuton?

- 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.

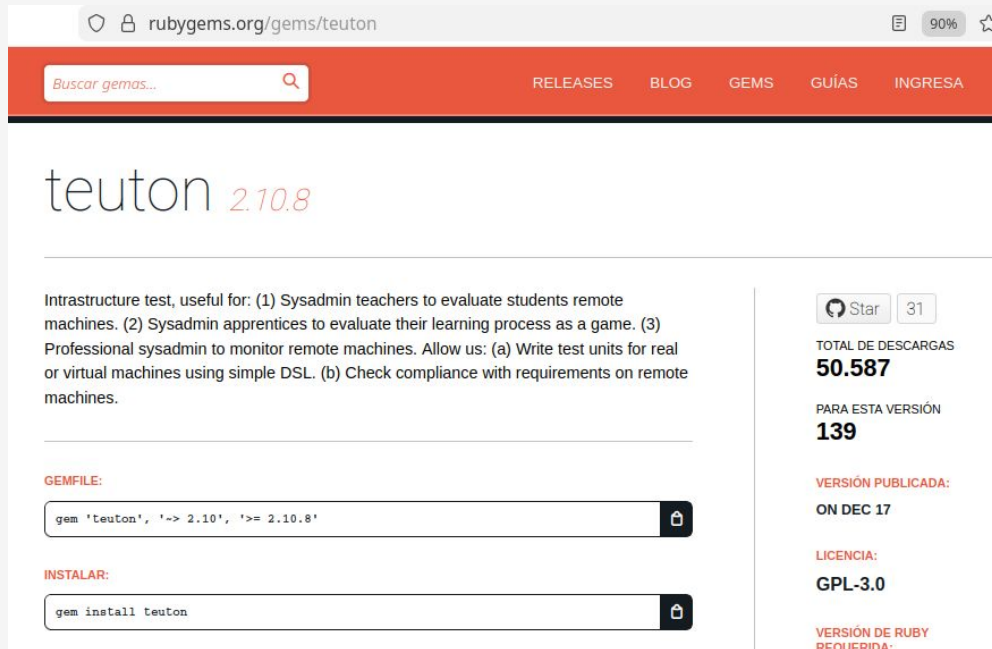


RubyGems - “gem install teuton”

Aplicar tests para mantener la calidad de...

- El código.
- ¿La infraestructura?
- ¿Los scripts?
- ¿Configuraciones?
- ¿Entornos?

==> ¡Cualquier entorno CLI!



The screenshot shows the RubyGems.org page for the 'teuton' gem. The browser address bar displays 'rubygems.org/gems/teuton'. The page has a red header with a search bar and navigation links: RELEASES, BLOG, GEMS, GUÍAS, and INGRESA. The main content area features the gem name 'teuton' with the version '2.10.8' in red. Below this is a description of the gem as an infrastructure test tool. On the right side, there is a 'Star' button with a count of 31, and statistics for downloads (50,587 total, 139 for this version) and release date (December 17). At the bottom, there are sections for 'GEMFILE:' and 'INSTALAR:' with corresponding code snippets and copy buttons.

rubygems.org/gems/teuton

Buscar gemas...

RELEASES BLOG GEMS GUÍAS INGRESA

teuton 2.10.8

Infrastructure 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.

Star 31

TOTAL DE DESCARGAS
50.587

PARA ESTA VERSIÓN
139

VERSIÓN PUBLICADA:
ON DEC 17

LICENCIA:
GPL-3.0

VERSIÓN DE RUBY
REQUERIDA:

GEMFILE:

```
gem 'teuton', '~> 2.10', '>= 2.10.8'
```

INSTALAR:

```
gem install teuton
```

Instalación - CLI

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

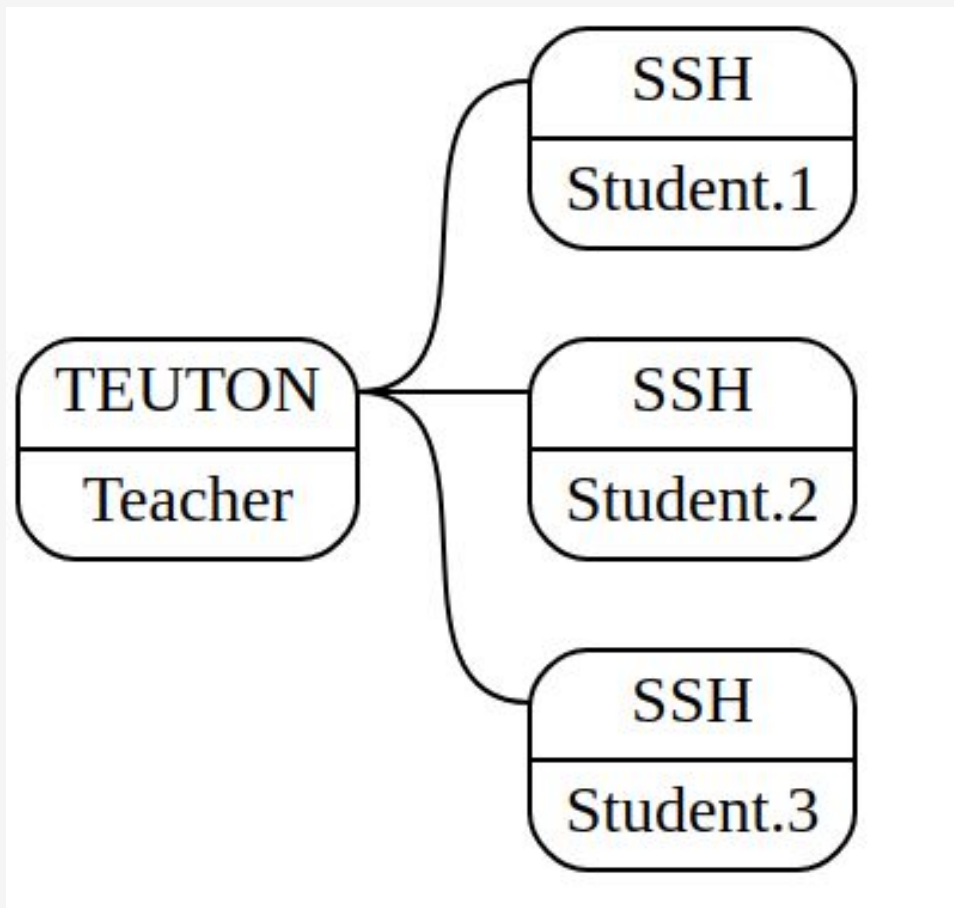
```
$ teuton help
```

Commands:

```
teuton [run] [OPTIONS] DIRECTORY # Run test from directory
teuton check [OPTIONS] DIRECTORY # Check test and config file content
teuton config [OPTIONS] DIRECTORY # Suggest configuration.
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
```

Entorno de trabajo

- **config.yaml**: Fichero de configuración.
- **start.rb**: Test (target, run ,expect, etc.)



DSL (Domain Specific Language)

- **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 qué debes "mirar" en la salida del comando...

¡Ya tienes el test!

```
group "Learn about targets" do
  target "Create user obiwan"
  run "id obiwan"
  expect ["uid=", "(obiwan)", "gid="]

  target "Delete user vader"
  run "id vader"
  expect_fail
end
```

Configuration + Test

```
# File: config.yaml
# Desc: configuracion file
---
global:
  host1_username: root
cases:
- tt_members: student_1
  host1_ip: 192.168.1.201
  host1_password: secret_1
- tt_members: student_2
  host1_ip: 192.168.1.202
  host1_password: secret_2
- tt_members: student_3
  host1_ip: 127.0.0.1
  host1_password: secret_3
```

```
# File: start.rb
# Desc: test definitions
group "Remote host" do
  target "Create user root"
  run "id root", on: :host1
  expect ["uid=", "(root)", "gid="]

  target "Delete user vader"
  run "id vader", on: :host1
  expect_fail
end

play do
  show
  export
end
```

DEMO!

Roadmap

github.com/teuton-software/teuton

github.com/teuton-software/teuton-get

1. Documentación
2. Difusión
3. Feedback
4. teuton-server
5. Orientar a monitorización



¡Muchas gracias!

github: @dvarrui
email: dvarrui@proton.me
repo: github.com/teuton-software/teuton