

# EJECUCIÓN REMOTA Y CONFIGURACIÓN AUTOMÁTICA

## TAREA 1

---

Autor: Diego Marco Beisty, 755232

Fecha: 15/03/2020

Asignatura: Administración de Sistemas 2

## FUNCIONALIDAD

Tras la realización de la parte 1 del trabajo, u.rb soporta los siguientes comandos:

p	ping TCP al puerto 22
s "command"	ejecución comando remoto por ssh

## ASPECTOS DE DISEÑO

### Error checking:

Se ha decidido hacer un sencillo control de errores a los argumentos del programa. Se abortaría la ejecución del mismo si el comando introducido tras ./u.rb no está disponible o directamente no se ha introducido ningún argumento.

### Obtención de direcciones IP o dominios. del fichero de configuración:

Se va a leer línea por línea el fichero de configuración por defecto ~/.u/hosts, a razón de una máquina por línea.

En el caso de que no exista el fichero ~/.u/hosts, directamente no se leerá de él y la salida de cualquier comando introducido al programa será vacía. Como si se hubiese creado el fichero ~/.u/hosts pero no se hubiese especificado ninguna máquina.

### Comando ./u.rb p:

Tal y como se especifica en el enunciado, si el servicio ssh escuchando en una máquina especificada en el fichero de configuración responde al ping TCP, se imprimirá por stdout "máquina\_<nº máquina>: FUNCIONA".

En caso contrario se mostrará "máquina\_<nº máquina>:falla".

### Comando ./u.rb s <comando remoto>:

Se ha tenido en cuenta el caso en el que la máquina objetivo no está activa en la red y no responde. Ante este caso, para que no se quede colgado el programa hasta que caduque el timeout máximo que por defecto son 300s, se establecerá un timeout por defecto de 7 segundos, tras el cual se mostrará por stdout el mensaje "máquina<nº máquina>: UNREACHABLE".

Si se consigue ejecutar satisfactoriamente el comando en la máquina remota, se imprimirá por stdout "máquina<nº máquina>: éxito" seguido de la salida del comando remoto.

## PROBLEMAS ENCONTRADOS

Al leer las máquinas especificadas del fichero de configuración, no tuve en cuenta los caracteres especiales como los saltos de línea, implícitos en el fichero.

Esto ocasionaba que ninguna máquina respondiera al estar mal el dominio o IP obtenido de ~/.u/hosts.

Este problema se solucionó eliminando los saltos de línea con la opción strip.

```
File.foreach(f) { |host| maquinas << host.strip }
```

## CÓDIGO FUENTE

```
#!/usr/bin/ruby -w
#####
# File:    u.rb
# Author:  Diego Marco Beisty 755232
# Date:    15/03/2020
# Coms:    Herramienta de ejecución remota, despliegue y configuracion
#           automática
#####
require 'net/ping/tcp'
require 'net/ssh'

# <maquinas> := {"<domain>"|"<ip>"}
# Ejecuta ping TCP al puerto 22 con timeout de 0.1s a las máquinas
# especificadas
# en el array <maquinas>
def ping_tcp(maquinas)
  num_host = 1
  maquinas.each do |host|
    t = Net::Ping::TCP.new(host, 22, 0.1)
    if t.ping?
      puts "máquina_" + num_host.to_s + ": FUNCIONA"
    else
      puts "máquina_" + num_host.to_s + ": falla"
    end
    num_host = num_host + 1
  end
end

# Ejecuta comando remoto <remote_comand> mediante ssh a todas las
# máquinas
# del array <maquinas>
def ssh_command(maquinas, remote_command)
  num_host = 1
  maquinas.each do |host|
    begin
      ssh = Net::SSH.start(host, "a755232",:timeout=> 7)
      res = ssh.exec!(remote_command)
      puts "máquina" + num_host.to_s + ": exito"
      puts res + "\n\n"
    rescue
      puts "máquina" + num_host.to_s + ": UNREACHABLE\n\n"
    end
  end
end
```

```

        num_host = num_host + 1
    end
end

# Lee una maquina (ip o dominio) por linea del fichero <f_config>
# y las almacena en el array <maquinas>
def obtener(maquinas, f_config)
    f = File.expand_path(f_config) # Expandir ~
    if File.file?(f)
        File.foreach(f) { |host| maquinas << host.strip }
    end
end

options = ["p", "s"] #Comandos disponibles
command = ARGV.first #Comando introducido

#Error checking: Se aborta ejecución si command no está implementado o
es vacío
abort "Uso: u [p] [s \"comando\"]\n" +
    "  p          ping al puerto 22\n" +
    "  s \"command\"  ejecucion comando remoto\n" \
    unless options.include?(command)

maquinas=[] #Máquinas target
f_config = "~/u/hosts" #Fichero de configuración por defecto
obtener(maquinas, f_config)#Obtenemos las máquinas objetivo del fichero
de configuración

if command == "p"
    ping_tcp(maquinas)
elsif command == "s"
    ssh_command(maquinas, ARGV[1])
end

```