

Parcial #2 de Sistemas Operativos

Juliana Zúñiga, código 13207011

Diseño e implementación de pruebas para servicio web que realiza acciones sobre el sistema operativo (CentOS 6.8 server) mediante el uso de servidor de integración continua (Jenkins)

Se tomó como base el servicio web del examen 1 (https://github.com/julizuso/Parcial1_Juliana.git) y se instaló Jenkins siguiendo el procedimiento que se muestra en <https://github.com/d4n13lbc/testproject.git>.

Los pasos necesarios para la ejecución y prueba de la solución son:

1.

Crear un script .sh donde se active el ambiente creado para desarrollar las pruebas y ___. El script usado se muestra a continuación:

```
#!/usr/bin/env bash
set -e

. ~/.virtualenvs/parcial2/bin/activate

PYTHONPATH=. py.test --junitxml=python_tests.xml
```

2.

Crear un scrpt .py que realice pruebas a cada una de las funciones del servicio web implementado en python. Las funciones son: crear un archivo, listar los archivos creados, eliminar todos los archivos creados y listar los archivos creados en la ultima hora.

Para realizar las pruebas se utiliza la función **assert** proporcionada por **pytest**. Con esta función se verifica que las respuestas HTTP del servicio web sean las esperadas, es decir, que indiquen que la operación que realizan sobre el sistema operativo fue exitosa. En caso de que la operación no sea exitosa, se muestra un mensaje que indica el error ocurrido. Dicho error se especifica en la función **assert** utilizando la sintaxis:

```
assert "MENSAJE ESPERADO" == respuestaHTTP.status , "MENSAJE DE ERROR"
```

El script usado se muestra a continuación:

```

import pytest
import python
import json

@pytest.fixture
def client(request):
    client = python.app.test_client()
    return client

#PRUEBA DEL SERVICIO CREAR UN ARCHIVO
def create_file(client):
    return client.post('/files', data=json.dumps(dict(filename='test_file', content='It

def test_create_file(client):
    result= create_file(client)
    assert result.status == "201 CREATED", "Error creando archivo"

#PRUEBA DEL SERVICIO LISTAR TODOS LOS ARCHIVOS CREADOS
def get_files(client):
    return client.get('/files', follow_redirects=True)

def test_get_files(client):
    result = get_files(client)
    assert result.data != None, "Error obteniendo la lista de archivos"

#ESTA FUNCIÓN NO APLICA PARA EL SERVICIO PRESTADO
def put_file(client):
    return client.put('/files', follow_redirects=True)

def test_put_file(client):
    result= put_file(client)
    assert result.status == "404 NOT FOUND", "Error en el servicio"

#PRUEBA DEL SERVICIO ELIMINAR TODOS LOS ARCHIVOS CREADOS
def delete_files(client):
    return client.delete('/files', follow_redirects=True)

def test_delete_files(client):
    result= delete_files(client)
    assert result.status == "200 OK", "Error eliminando archivos"

#ESTA FUNCIÓN NO APLICA PARA EL SERVICIO PRESTADO
def post_recent(client):
    return client.post('/files/recently_created', follow_redirects=True)

def test_post_recent(client):
    result= post_recent(client)
    assert result.status == "404 NOT FOUND", "Error en el servicio"

#PRUEBA DEL SERVICIO LISTAR LOS ARCHIVOS RECIENTES
def get_recent_files(client):

```

```
return client.get('/files/recently_created',follow_redirects=True)

def test_get_recent_files(client):
    result = get_recent_files(client)
    assert result.data != None, "Error obteniendo la lista de archivos"

#ESTA FUNCIÓN NO APLICA PARA EL SERVICIO PRESTADO
def put_recent(client):
    return client.put('/files/recently_created',follow_redirects=True)

def test_put_recent(client):
    result= put_recent(client)
    assert result.status == "404 NOT FOUND", "Error en el servicio"

#ESTA FUNCIÓN NO APLICA PARA EL SERVICIO PRESTADO
def delete_recent(client):
    return client.put('/files/recently_created',follow_redirects=True)

def test_delete_recent(client):
    result= delete_recent(client)
    assert result.status == "404 NOT FOUND", "Error en el servicio"
```

3.

Acceder al portal web de Jenkins ingresando en el explorador la IP del servidor donde se ha habilitado dicho servicio, por el puerto 8080.

4.

Crear un proyecto de estilo libre, integrarlo con GitHub y poner la URL del repositorio donde estén guardados los scripts del servicio y los que realicen las pruebas del mismo.

General

Configurar el origen del código fuente

Disparadores de ejecuciones

Entorno de ejecución

Ejecutar

Acciones para ejecutar después.

Proyecto nombre

parcial2

Descripción

[Plain text]

Visualizar

☐ Desechar ejecuciones antiguas

☐ Esta ejecución debe parametrizarse

☒ GitHub project

Project url

https://github.com/julizuoso/Parcial2_SO_Juliana.git

Avanzado...

Guardar

Apply

Configurar el origen del código fuente

☐ Ninguno

☒ Git

Repositories

Repository URL

https://github.com/julizuoso/Parcial2_SO_Juliana.git

Credentials

- none -

Ad+

Avanzado...

Add Repository

5.

Configurar los disparadores de ejecuciones para decidir cada cuánto se ejecutan pruebas sobre el código fuente. En este caso, se ejecutan las pruebas cada 5 minutos.

Disparadores de ejecuciones

☐ Lanzar ejecuciones remotas (ejem: desde 'scripts')

☐ Build after other projects are built

☐ Build when a change is pushed to GitHub

☐ Consultar repositorio (SCM)

☒ Ejecutar periódicamente

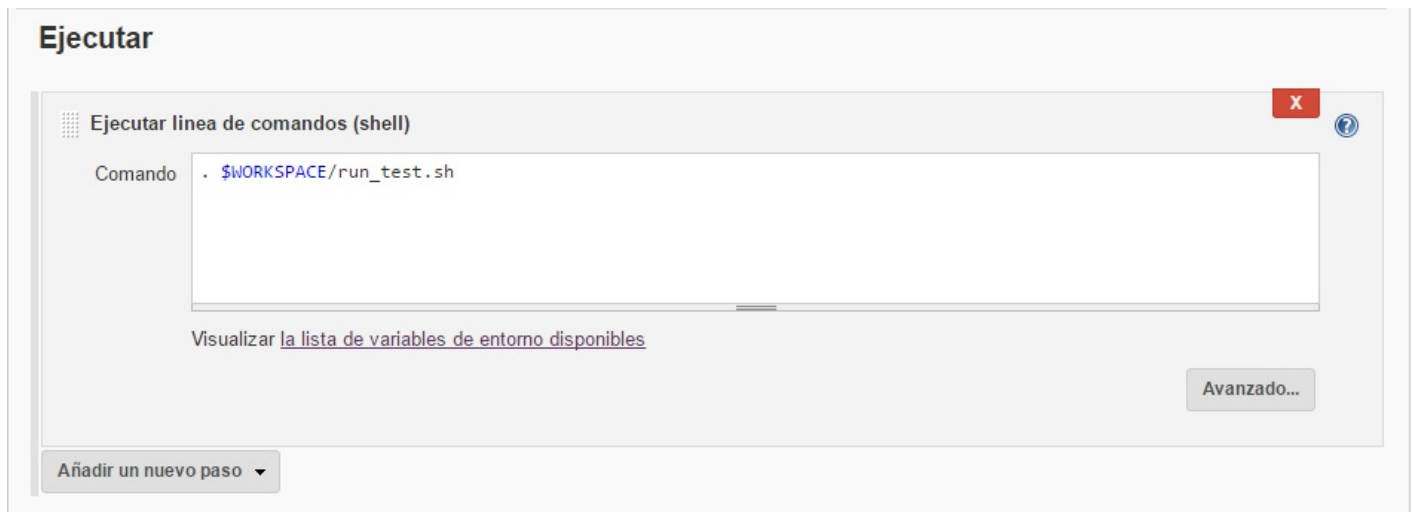
Programador

H/5 * * * *

Would last have run at lunes 7 de noviembre de 2016 11H04' COT; would next run at lunes 7 de noviembre de 2016 11H09' COT.

6.

En las opciones de ejecución, añadir un nuevo paso que ejecute en la línea de comandos de nuestro servidor el script .sh que fue creado en el paso 1 y guardar la configuración.




7.

A continuación se muestran las ejecuciones de las pruebas realizadas. Al hacer click en ellas y luego ver la salida de consola de cada una, obtenemos información sobre el resultado de las pruebas (si tuvieron éxito o no, y por qué fallaron). En la ejecución **#166** la prueba se realizó sobre el código fuente modificado para que respondiera algo inesperado al crear un nuevo archivo, por eso el resultado de la prueba no es exitoso como el de la ejecución **#165**.

 Volver al Panel de Control

 **Estado Actual**

 Cambios

 Zona de Trabajo

 Construir ahora

 Borrar Proyecto

 Configurar

 Move

 GitHub


Historia de tareas
Tendencia 

x

	#166	07-nov-2016 16:19
	#165	07-nov-2016 16:19
	#164	07-nov-2016 16:14
	#163	07-nov-2016 16:14
	#162	07-nov-2016 16:14

Salida de consola de la ejecución **#165**:

```
Ejecutado por el programador
Ejecutando en el espacio de trabajo /var/lib/jenkins/workspace/parcial2
> git rev-parse --is-inside-work-tree # timeout=10
Fetching changes from the remote Git repository
> git config remote.origin.url https://github.com/julizuso/Parcial2_SO_Juliana.git #
Fetching upstream changes from https://github.com/julizuso/Parcial2_SO_Juliana.git
> git --version # timeout=10
> git fetch --tags --progress https://github.com/julizuso/Parcial2_SO_Juliana.git +r
> git rev-parse refs/remotes/origin/master^{commit} # timeout=10
> git rev-parse refs/remotes/origin/origin/master^{commit} # timeout=10
Checking out Revision 313b8167d0c57cc03fec12f1d1002b669562456b (refs/remotes/origin/ma
> git config core.sparsecheckout # timeout=10
> git checkout -f 313b8167d0c57cc03fec12f1d1002b669562456b
> git rev-list 0e8d8bf4c497d96f098cd9a546adb0d9d61aa18e # timeout=10
[parcial2] $ /bin/sh -xe /tmp/hudson8850538158026302707.sh
+ . /var/lib/jenkins/workspace/parcial2/run_test.sh
++ set -e
++ . /var/lib/jenkins/.virtualenvs/parcial2/bin/activate
```

```

+++ deactivate nondestructive
+++ unset -f pydoc
+++ '[' -z '' ']'
+++ '[' -z '' ']'
+++ '[' -n /bin/sh ']'
+++ hash -r
+++ '[' -z '' ']'
+++ unset VIRTUAL_ENV
+++ '[' '!' nondestructive = nondestructive ']'
+++ VIRTUAL_ENV=/var/lib/jenkins/.virtualenvs/parcial2
+++ export VIRTUAL_ENV
+++ _OLD_VIRTUAL_PATH=/sbin:/usr/sbin:/bin:/usr/bin
+++ PATH=/var/lib/jenkins/.virtualenvs/parcial2/bin:/sbin:/usr/sbin:/bin:/usr/bin
+++ export PATH
+++ '[' -z '' ']'
+++ '[' -z '' ']'
+++ _OLD_VIRTUAL_PS1=
+++ '[' x '!=' x ']'
+++++ basename /var/lib/jenkins/.virtualenvs/parcial2
+++ PS1='(parcial2) '
+++ export PS1
+++ alias pydoc
+++ '[' -n /bin/sh ']'
+++ hash -r
++ PYTHONPATH=.
++ py.test --junitxml=python_tests.xml
===== test session starts =====
platform linux2 -- Python 2.6.6, pytest-3.0.3, py-1.4.31, pluggy-0.4.0
rootdir: /var/lib/jenkins/workspace/parcial2, inifile:
plugins: cov-2.4.0
collected 8 items

test_project.py .....

--- generated xml file: /var/lib/jenkins/workspace/parcial2/python_tests.xml ---
===== 8 passed in 0.35 seconds =====
Finished: SUCCESS

```

Salida de consola de la ejecución #166:

```

Lanzada por el usuario Juliana
Ejecutando en el espacio de trabajo /var/lib/jenkins/workspace/parcial2
> git rev-parse --is-inside-work-tree # timeout=10
Fetching changes from the remote Git repository
> git config remote.origin.url https://github.com/julizuso/Parcial2\_SO\_Juliana.git #
Fetching upstream changes from https://github.com/julizuso/Parcial2\_SO\_Juliana.git
> git --version # timeout=10
> git fetch --tags --progress https://github.com/julizuso/Parcial2\_SO\_Juliana.git +r
> git rev-parse refs/remotes/origin/master^{commit} # timeout=10
> git rev-parse refs/remotes/origin/origin/master^{commit} # timeout=10

```

```

Checking out Revision d232b1a17854c028d494d41e8f693694ce81d064 (refs/remotes/origin/ma
> git config core.sparsecheckout # timeout=10
> git checkout -f d232b1a17854c028d494d41e8f693694ce81d064
> git rev-list 313b8167d0c57cc03fec12f1d1002b669562456b # timeout=10
[parcial2] $ /bin/sh -xe /tmp/hudson5930937829762580892.sh
+ . /var/lib/jenkins/workspace/parcial2/run_test.sh
++ set -e
++ . /var/lib/jenkins/.virtualenvs/parcial2/bin/activate
+++ deactivate nondestructive
+++ unset -f pydoc
+++ '[' -z '' ']'
+++ '[' -z '' ']'
+++ '[' -n /bin/sh ']'
+++ hash -r
+++ '[' -z '' ']'
+++ unset VIRTUAL_ENV
+++ '[' '!' nondestructive = nondestructive ']'
+++ VIRTUAL_ENV=/var/lib/jenkins/.virtualenvs/parcial2
+++ export VIRTUAL_ENV
+++ _OLD_VIRTUAL_PATH=/sbin:/usr/sbin:/bin:/usr/bin
+++ PATH=/var/lib/jenkins/.virtualenvs/parcial2/bin:/sbin:/usr/sbin:/bin:/usr/bin
+++ export PATH
+++ '[' -z '' ']'
+++ '[' -z '' ']'
+++ _OLD_VIRTUAL_PS1=
+++ '[' x '!=' x ']'
++++ basename /var/lib/jenkins/.virtualenvs/parcial2
+++ PS1='(parcial2) '
+++ export PS1
+++ alias pydoc
+++ '[' -n /bin/sh ']'
+++ hash -r
++ PYTHONPATH=.
++ py.test --junitxml=python_tests.xml
===== test session starts =====
platform linux2 -- Python 2.6.6, pytest-3.0.3, py-1.4.31, pluggy-0.4.0
rootdir: /var/lib/jenkins/workspace/parcial2, inifile:
plugins: cov-2.4.0
collected 8 items

test_project.py F.....

--- generated xml file: /var/lib/jenkins/workspace/parcial2/python_tests.xml ---
===== FAILURES =====
_____ test_create_file _____

client = <FlaskClient <Flask 'python'>>

    def test_create_file(client):
        result= create_file(client)
>         assert result.status == "201 CREATED", "Error creando archivo"
E         AssertionError: Error creando archivo

```



```
E    assert '200 OK' == '201 CREATED'
E      - 200 OK
E      + 201 CREATED
```

```
test_project.py:17: AssertionError
```

```
===== 1 failed, 7 passed in 0.29 seconds =====
```

```
Build step 'Ejecutar linea de comandos (shell)' marked build as failure
```

```
Finished: FAILURE
```

El repositorio en **GitHub** que contiene el código fuente de la solución se encuentra en https://github.com/julizuoso/Parcial2_SO_Juliana.git