

Informe del Proyecto 1: Computación Voluntaria y Sistemas Multi-Agentes

MCC603 - Computación Flexible en Cloud

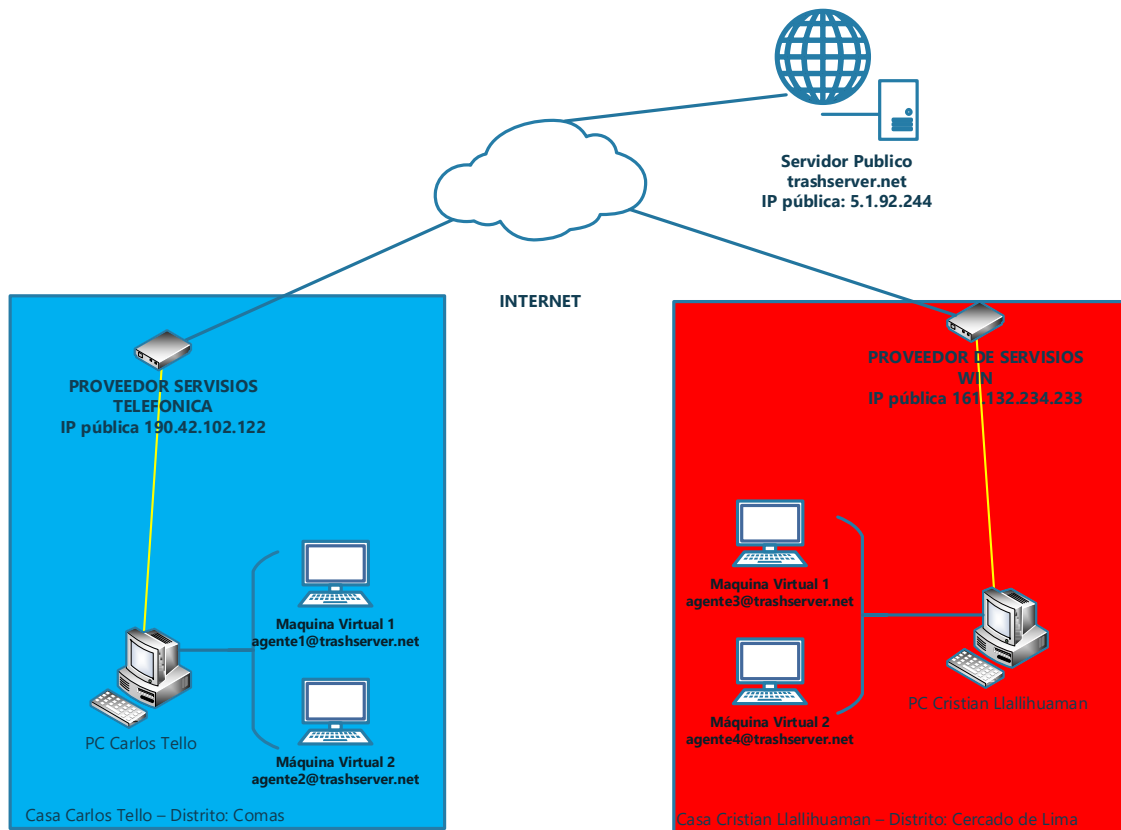
Docente: Dr. Yudith Cardinale

Integrantes: Ing. Carlos Tello Tapia

Cristian Llallihuaman Calderón

Arquitectura de lo implementado:

La estructura de nuestra red p2p está compuesta por un servidor XMPP, para el presente proyecto utilizamos el servidor público con dominio **trashserver.net** (IP pública 5.1.92.244) a nivel de comunicación de agentes tenemos 4 agentes corriendo en distintos sites, 2 agentes peer se encuentra dentro de la red del proveedor de servicios Telefonica del Perú con IP publica 190.42.102.122 y 2 agentes peer en otro site con IP publica 161.132.234.233 del proveedor de servicios WIN Perú.



Para tal fin procedemos a crear los agentes en el portal <https://trashserver.net/registriren/>

← → trashserver.net/registrieren/ Beachte bitte die Nutzungsbedingungen

Registrieren eine XMPP-Konto

Esta página te permite crear una cuenta XMPP este servidor XMPP. Tu JID (Jabber ID) será de la forma: nombreusuario@servidor. Por favor lee detenidamente las instrucciones para rellenar correctamente los campos.

Nombre de usuario:

1. No importa si usas mayúsculas: macbeth es lo mismo que MacBeth y Macbeth.
2. Caracteres no permitidos: * & /, < > @

Servidor:

Contraseña:

1. No le digas tu contraseña a nadie, ni siquiera a los administradores del servidor XMPP.
2. Puedes cambiar tu contraseña después, usando un cliente XMPP.
3. Algunos clientes XMPP pueden guardar tu contraseña en la máquina, pero solo deberías hacer esto en tu propia máquina personal por razones de seguridad.
4. Almacena tu contraseña, o apóñala en un papel en un lugar seguro. En XMPP no hay un método automatizado para recuperar la contraseña si la olvidas.

Verificación de la contraseña:

Teclas el texto que ves

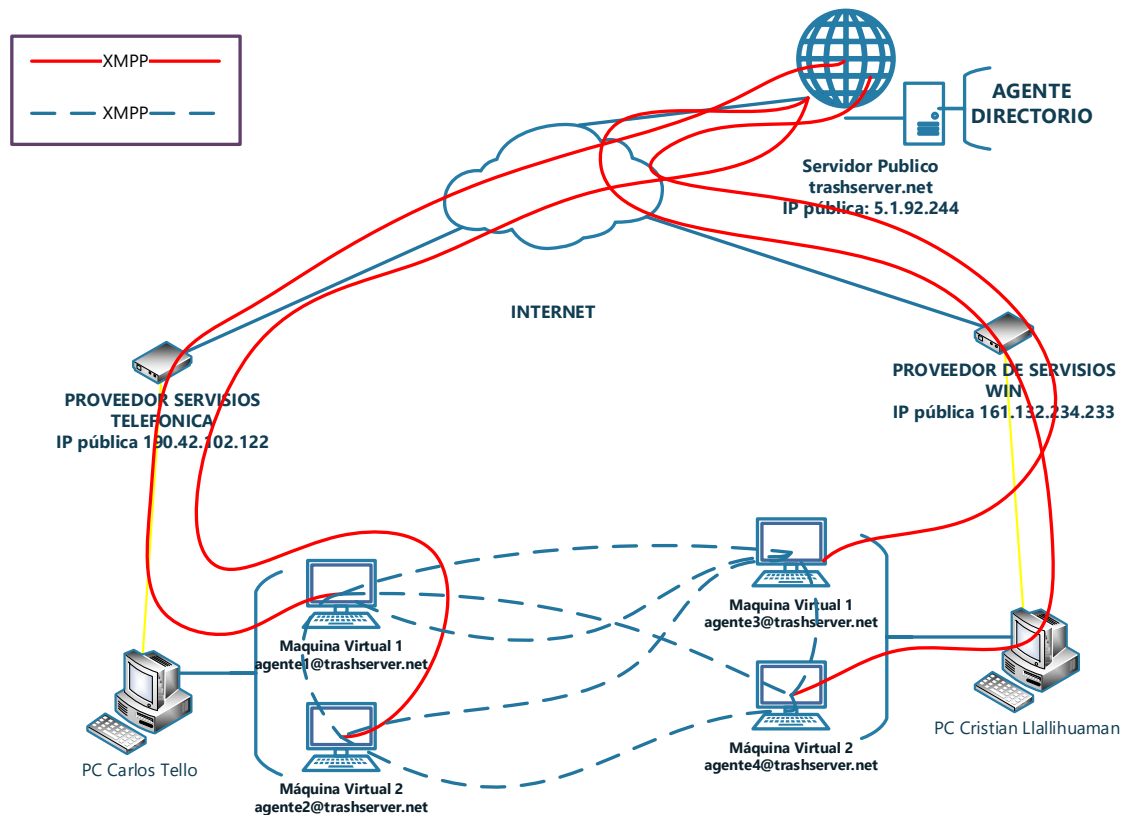


Se muestra a continuación los agentes creados:

| | Agentes Peer | Password |
|----------|--|----------|
| Agente 1 | mccagente1@trashserver.net | 12345678 |
| Agente 2 | mccagente2@trashserver.net | 12345678 |
| Agente 3 | mccagente3@trashserver.net | 12345678 |
| Agente 4 | mccagente4@trashserver.net | 12345678 |

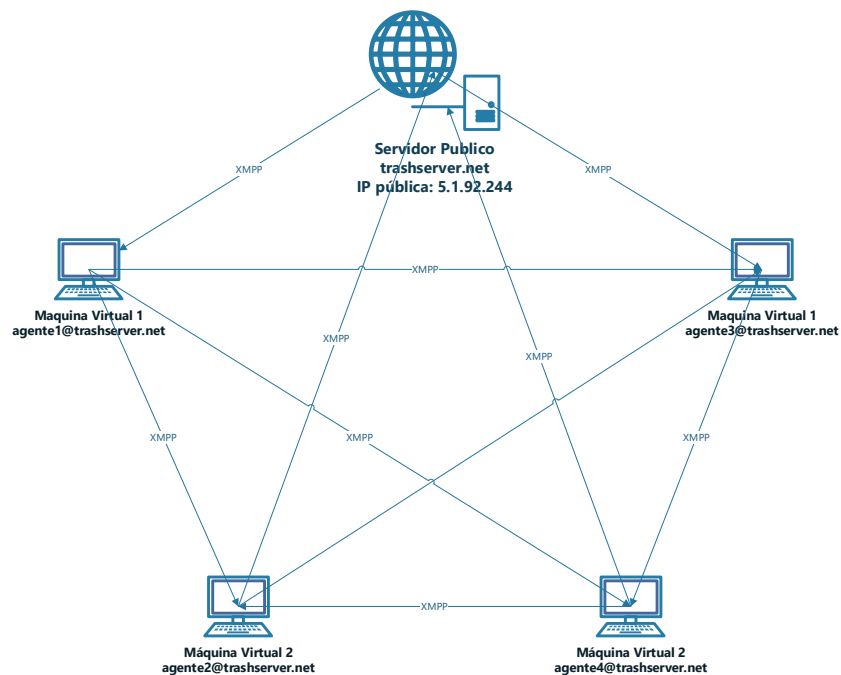
Diseño Lógico:

A continuación, se muestra la red p2p a nivel lógico, los establecimientos del protocolo XMPP a nivel público se muestra con líneas rojas y entre agentes se muestra con líneas azules delineadas, así mismo el agente directorio alojado en el servidor público (mas referencia en [FAQ - Häufig gestellte Fragen | trashserver.net XMPP](#)).



Red P2P desplegada con agentes SPADE:

La siguiente imagen muestra los agentes peer de la red y servidor XMPP de manera lógica



Pruebas de la red p2p implementada:

Como objetivos del presente proyecto obtuvimos %de uso de CPU de agentes, obtención de la IP pública del agente consultado, obtención de ruta de carpeta compartida, así mismos archivos contenidos en el directorio compartido, otro de los objetivos era cuando un peer tiene uso de CPU por debajo de 40% El peer solicitante enviará un script de compilación y ejecución de su programa y el código fuente.

La siguiente imagen muestra un porcentaje de uso 2%, es menor al 40% con lo cual se puede generar un ejecutable para uso de computo

```
NIRQA@LAPTOP-UR0P10G6 MINGW64 ~/Documents/
$ python agente2vf.py
Agente receptor Iniciado
Agente receptor en funcionamiento

---Directorio:---
/home/agent1/compartir

---Archivos:---
cancion2.mp3
cancion5.mp3
cancion4.mp3
cancion1.mp3
cancion3.mp3
cancion6.mp3

---IP PUBLICA:---
190.42.102.122

---%CPU USADO:---
2.0

Agente finalizado
```

La siguiente imagen muestra un porcentaje de uso 44.7%, mayor al 40% con lo cual no se generara un ejecutable para uso de computo

```
NIRQA@LAPTOP-UR0P10G6 MINGW64 ~/Document
$ python agente2vf.py
Agente receptor Iniciado
Agente receptor en funcionamiento

---Directorio:---
/home/agent1/compartir

---Archivos:---
cancion2.mp3
cancion5.mp3
cancion4.mp3
cancion1.mp3
cancion3.mp3
cancion6.mp3

---IP PUBLICA:---
190.42.102.122

---%CPU USADO:---
44.7

Agente finalizado
```

La siguiente imagen se realizó desde la consola nativa, la cual muestra un porcentaje de uso 1%, es menor al 40% con lo cual se puede generar un ejecutable para uso de computo

```
C:\Users\rmuno\Documents\test_python>python agente2vf.py
Agente receptor Iniciado
Agente receptor en funcionamiento

---Directorio:---
/home/agente1/compartir

---Archivos:---
cancion2.mp3
cancion5.mp3
cancion4.mp3
cancion1.mp3
cancion3.mp3
cancion6.mp3

---IP PUBLICA:---
190.42.102.122

---%CPU USADO:---
1.0

Agente finalizado
```

Verificación de directorio generado

```
1 Agente finalizado
1 NIRQA@LAPTOP-UR0PIOG6 MINGW64 ~/Documents/test_python
1 $ ls
1 Directorio_generado_agente2/ agente2vf.py spade.py.bak test_spade.py.bak
1 __pycache__/ pingrec.py spade_test.py
1 agente2.py pingrec.py.bak test_spade.py
1 NIRQA@LAPTOP-UR0PIOG6 MINGW64 ~/Documents/test_python
```

Ingresamos al directorio

```
Directorio_generado_agente2/ agente2.py pingrec.py spade.py.bak test_spade.py
__pycache__/ agente2vf.py pingrec.py.bak spade_test.py test_spade.py.bak

NIRQA@LAPTOP-UR0PIOG6 MINGW64 ~/Documents/test_python
$ cd Directorio_generado_agente2/

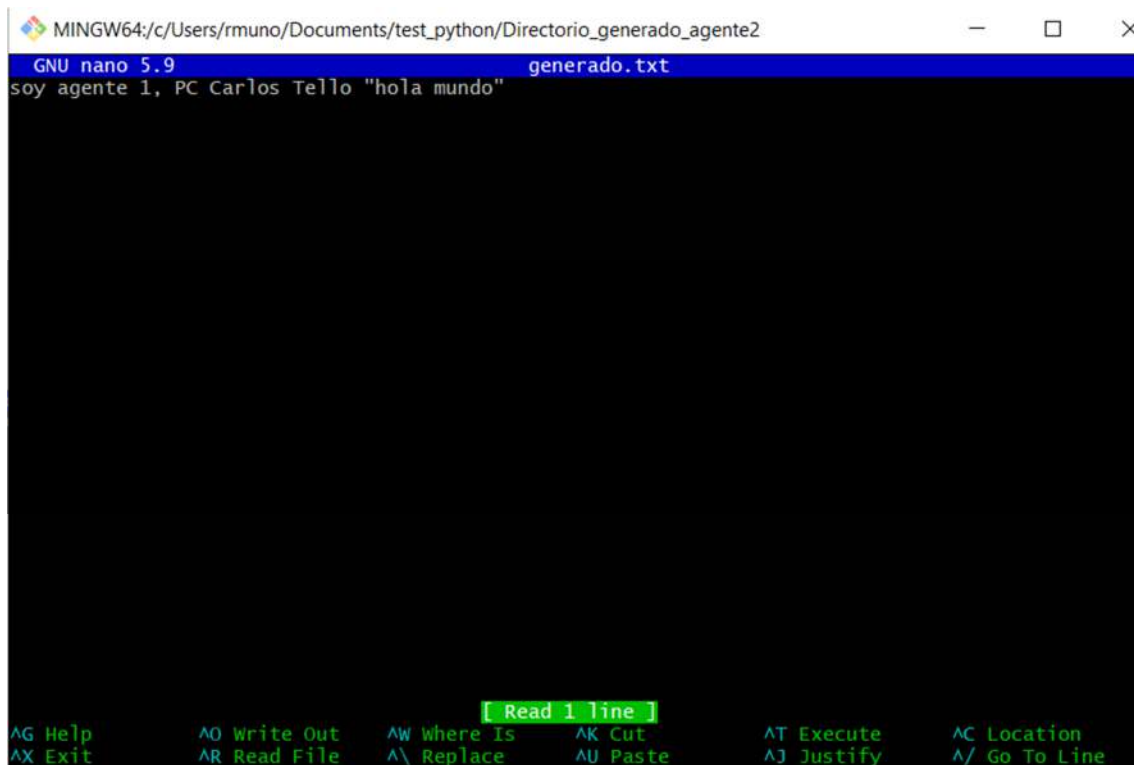
NIRQA@LAPTOP-UR0PIOG6 MINGW64 ~/Documents/test_python/Directorio_generado_agente2
$

NIRQA@LAPTOP-UR0PIOG6 MINGW64 ~/Documents/test_python/Directorio_generado_agente2
$ pwd
/c/Users/rmuno/Documents/test_python/Directorio_generado_agente2

NIRQA@LAPTOP-UR0PIOG6 MINGW64 ~/Documents/test_python/Directorio_generado_agente2
$ ls
generado.txt

NIRQA@LAPTOP-UR0PIOG6 MINGW64 ~/Documents/test_python/Directorio_generado_agente2
$
```

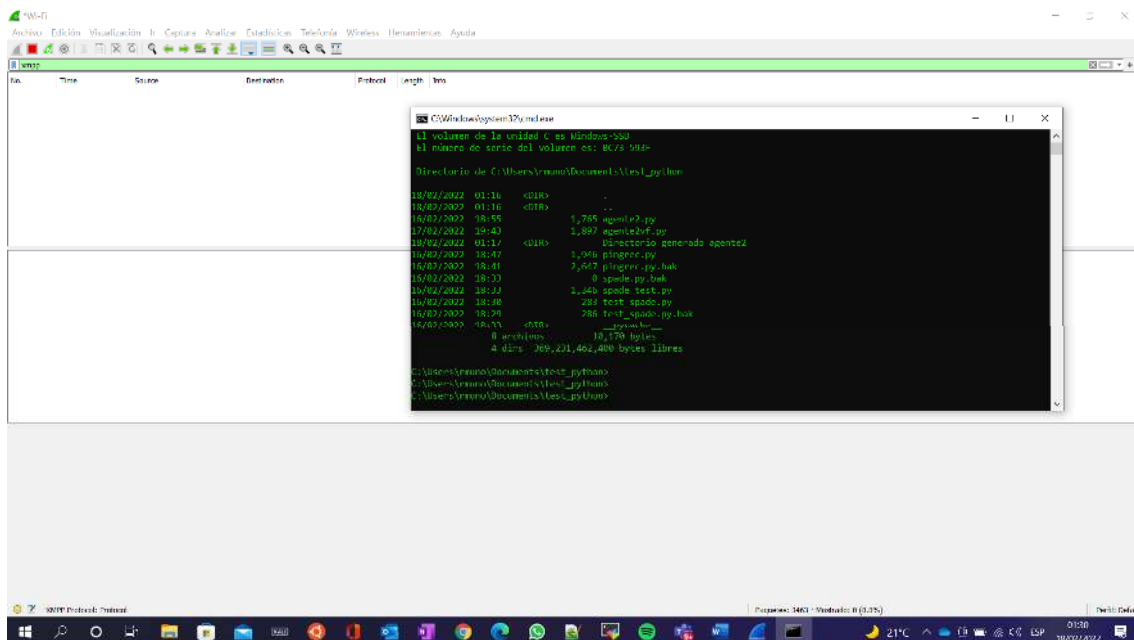
Leemos el archivo generado:



The screenshot shows a terminal window with the title bar 'MINGW64:/c/Users/rmuno/Documents/test_python/Directorio_generado_agente2'. The text editor 'GNU nano 5.9' is open, editing the file 'generado.txt'. The content of the file is 'soy agente 1, PC Carlos Tello "hola mundo"'. The bottom status bar shows various keyboard shortcuts for nano, including 'Read 1 line' which is highlighted in green.

Las siguientes imagenes muestra las transacciones que se realizan desde agente que hace petición, para dicho fin hacemos uso de wireshark y filtrado por protocolo XMPP

Estado inicial previo a la consulta:



Estado final, luego de lanzar el script en agente receptor

