

Práctica 3 - Sistemas Distribuidos

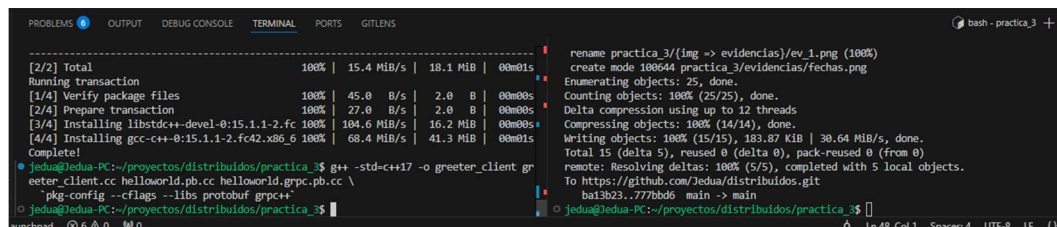
Ejercicio 1: Servicios SayHello, siguienteFecha y sumaDiasFecha

Se implementó un servicio gRPC con Python que permite invocar tres métodos remotos:

- SayHello y SayHelloAgain: saludan al usuario.
- siguienteFecha: calcula el día siguiente a una fecha dada.
- sumaDiasFecha: suma una cantidad de días a una fecha dada.

También se realizó la interoperabilidad con cliente en C++ utilizando los archivos generados por protoc.

Los servicios fueron probados exitosamente entre ambos lenguajes.



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS GITLENS
[2/2] Total 100% | 15.4 MiB/s | 18.1 MiB | 00m01s
Running transaction
[1/4] Verify package files 100% | 45.0 B/s | 2.0 B | 00m00s
[2/4] Prepare transaction 100% | 27.0 B/s | 2.0 B | 00m00s
[3/4] Installing libstdc++-devel-0:15.1.1-2.fc 100% | 104.6 MiB/s | 16.2 MiB | 00m00s
[4/4] Installing gcc-c++-0:15.1.1-2.fc42.x86_64 100% | 68.4 MiB/s | 41.3 MiB | 00m01s
Complete!
jedu@jedua-PC:~/proyectos/distribuidos/practica_3$ g++ -std=c++17 -o greeter_client greeter_client.cc helloworld.pb.cc helloworld.grpc.pb.cc \
-pkg-config --cflags --libs protobuf grpc++
jedu@jedua-PC:~/proyectos/distribuidos/practica_3$
rename practica_3(img -> evidencias/ev_1.png (100%)
create mode 100644 practica_3/evidencias/fechas.png
Enumerating objects: 25, done.
Counting objects: 100% (25/25), done.
Delta compression using up to 12 threads
Compressing objects: 100% (14/14), done.
Writing objects: 100% (15/15), 183.87 KiB | 30.64 MiB/s, done.
Total 15 (delta 5), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (5/5), completed with 5 local objects.
To https://github.com/jedu/distribuidos.git
ba13b23..777bbd6 main -> main
jedu@jedua-PC:~/proyectos/distribuidos/practica_3$
```

Evidencia: Cliente C++ invocando servidor Python.

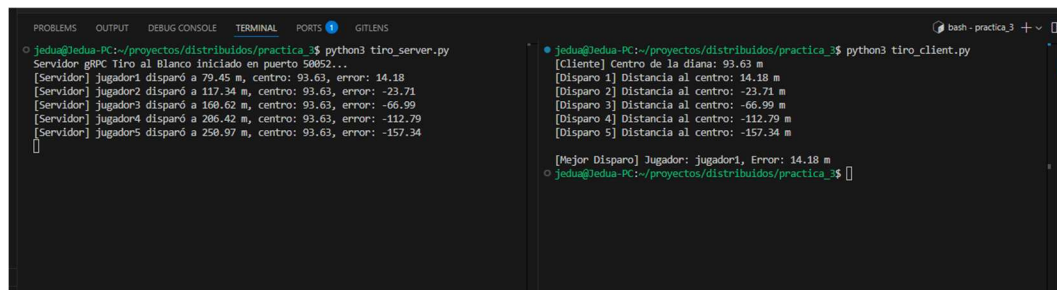
Ejercicio 2: Tiro al blanco

Se desarrolló un servidor gRPC que simula un juego de tiro al blanco. El cliente puede:

- Solicitar la ubicación del centro de la diana.
- Realizar disparos con un ángulo y velocidad determinados.
- Consultar quién ha realizado el mejor disparo hasta el momento.

El servidor calcula la distancia del disparo al centro usando la fórmula física del alcance.

El mejor disparo se define como el que tiene menor error absoluto respecto al centro.



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS GITLENS
jedu@jedua-PC:~/proyectos/distribuidos/practica_3$ python3 tiro_server.py
Servidor Tiro al Blanco iniciado en puerto 50052...
[Servidor] jugador1 disparó a 79.45 m, centro: 93.63, error: 14.18
[Servidor] jugador2 disparó a 117.34 m, centro: 93.63, error: -23.71
[Servidor] jugador3 disparó a 160.62 m, centro: 93.63, error: -66.99
[Servidor] jugador4 disparó a 206.42 m, centro: 93.63, error: -112.79
[Servidor] jugador5 disparó a 250.97 m, centro: 93.63, error: -157.34

jedu@jedua-PC:~/proyectos/distribuidos/practica_3$ python3 tiro_client.py
[Cliente] Centro de la diana: 93.63 m
[Disparo 1] Distancia al centro: 14.18 m
[Disparo 2] Distancia al centro: -23.71 m
[Disparo 3] Distancia al centro: -66.99 m
[Disparo 4] Distancia al centro: -112.79 m
[Disparo 5] Distancia al centro: -157.34 m

[Mejor Disparo] Jugador: jugador1, Error: 14.18 m
jedu@jedua-PC:~/proyectos/distribuidos/practica_3$
```

Evidencia: Cliente Python interactuando con el servidor de tiro al blanco.