

# Proyecto 2 RC

Tobias Briones

March 2021

## 1 Introducción

Se documentará la conexión establecida entre el PC cliente y el servidor remoto del proyecto #2.

## 2 Especificación

El proyecto consiste en un juego escrito en Go que genera masmorras aleatorias mediante el algoritmo de Prim y consiste en un servidor también escrito en Go que permite la conexión de  $n$  jugadores mediante web sockets. El servidor asigna un id único incremental a cada jugador (pueden tener el mismo nombre) y crea constantemente partidas aleatorias las cuales son emitidas a los jugadores cada  $x$  segundos. Cada jugador puede moverse por las masmorras recolectando diamantes los cuales les asignan 30 puntos. Si los diamantes de la partida actual se acaban, termina el juego y gana el que tiene más puntos. Sino se empezará una nueva partida automáticamente al terminar el tiempo. Se tienen dos ramas, una para que el servidor genere masmorras manejables por los jugadores y otra para que genere muchas masmorras. En la raíz del proyecto hay un archivo *user.json* para configurar el nombre del jugador.

El servidor se ha desplegado (no producción) en una instancia de AWS EC2 el cual ha permitido establecer las conexiones remotas de cada jugador al servidor del juego.

## 3 Procedimiento

El juego consiste en dos módulos: *game* y *server*. Luego de crear el VPS que alojará el servidor, instalar Go y clonar el repositorio del proyecto, se sigue que:

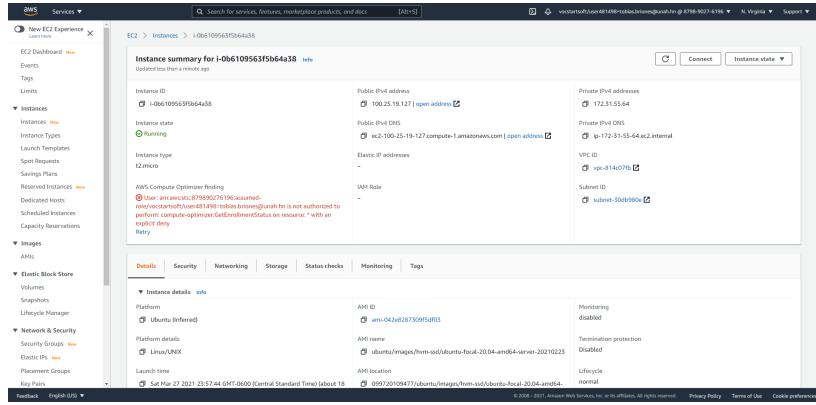


Figure 1: VPS

```
ubuntu@ip-172-31-55-64:~$ ssh -i "dungeon-mst.pem" ubuntu@ec2-100-25-19-127.compute-1.amazonaws.com
Welcome to Ubuntu 20.04.2 LTS (GNU/Linux 5.4.0-1038-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/adantage

System information as of Sun Mar 28 23:51:19 UTC 2021

System load: 0.0          Processes:           103
Usage of /: 28.4% of 7.69GB Users logged in: 0
Memory usage: 24%          IPv4 address for eth0: 172.31.55.64
Swap usage: 0%

* Introducing self-healing high availability clusters in MicroK8s.
Simple, hardened, Kubernetes for production, from RaspberryPi to DC.
https://microk8s.io/high-availability

37 updates can be installed immediately.
24 of these updates are security updates.
To see these additional updates run: apt list --upgradable

Last login: Sun Mar 28 23:48:26 2021 from 205.211.193.159
ubuntu@ip-172-31-55-64:~$ cd work/
ubuntu@ip-172-31-55-64:~/work$ cd src/
ubuntu@ip-172-31-55-64:~/work/src$ cd github.com/
ubuntu@ip-172-31-55-64:~/work/src/github.com$ cd tobiasbriones/
ubuntu@ip-172-31-55-64:~/work/src/github.com/tobiasbriones/dungeon-mst/
ubuntu@ip-172-31-55-64:~/work/src/github.com/tobiasbriones/dungeon-mst$ cd server/
ubuntu@ip-172-31-55-64:~/work/src/github.com/tobiasbriones/dungeon-mst/server$ go build .
ubuntu@ip-172-31-55-64:~/work/src/github.com/tobiasbriones/dungeon-mst/server$ ls
ai client.go go.mod hub.go model model.go server server.go
ubuntu@ip-172-31-55-64:~/work/src/github.com/tobiasbriones/dungeon-mst/server$ ./server
```

Figure 2: VPS

La dirección IPv4 pública de la instancia se asigna al cliente del juego:

```

dne 1st New Migrat... Code Behavior Page Books Go Window Help dungeon-mst [!dungeon-mst-client.go]
Project *  Recent  Editor  Run  Build  Go  Window  Help  dungeon-mst [!dungeon-mst-client.go]
dungeon-mst game client client.go
  > Project
    > Recent
      > Game
        > server
        > gomod
        > Docker
        > unit tests
        > External Libraries
        > Scratches and Consoles
  > Editor
  > Run
  > Build
  > Go
  > Window
  > Help
dungeon-mst-client.go
1 package client
2
3 // Copyright (c) 2021 Tobias Briones. All rights reserved.
4
5 import (
6     "fmt"
7     "log"
8     "net/http"
9     "log/zap"
10    "github.com/tobiashbriones/dungeon-mst/server"
11    "github.com/tobiashbriones/dungeon-mst/client"
12    "github.com/tobiashbriones/dungeon-mst/client/config"
13    "github.com/tobiashbriones/dungeon-mst/client/player"
14    "github.com/tobiashbriones/dungeon-mst/client/response"
15    "github.com/tobiashbriones/dungeon-mst/client/socket"
16    "github.com/tobiashbriones/dungeon-mst/client/util"
17    "github.com/tobiashbriones/dungeon-mst/client/world"
18    "github.com/tobiashbriones/dungeon-mst/client/world/room"
19    "github.com/tobiashbriones/dungeon-mst/client/world/room/exit"
20    "github.com/tobiashbriones/dungeon-mst/client/world/room/room"
21    "github.com/tobiashbriones/dungeon-mst/client/world/room/room/door"
22    "github.com/tobiashbriones/dungeon-mst/client/world/room/room/room"
23    "github.com/tobiashbriones/dungeon-mst/client/world/room/room/room/exit"
24    "github.com/tobiashbriones/dungeon-mst/client/world/room/room/room/room"
25    "github.com/tobiashbriones/dungeon-mst/client/world/room/room/room/room/room"
26    "github.com/tobiashbriones/dungeon-mst/client/world/room/room/room/room/room/room"
27    "github.com/tobiashbriones/dungeon-mst/client/world/room/room/room/room/room/room/room"
28    "github.com/tobiashbriones/dungeon-mst/client/world/room/room/room/room/room/room/room/room"
29    "github.com/tobiashbriones/dungeon-mst/client/world/room/room/room/room/room/room/room/room/room"
30    "github.com/tobiashbriones/dungeon-mst/client/world/room/room/room/room/room/room/room/room/room/room"
31    "github.com/tobiashbriones/dungeon-mst/client/world/room/room/room/room/room/room/room/room/room/room/room"
32    "github.com/tobiashbriones/dungeon-mst/client/world/room/room/room/room/room/room/room/room/room/room/room/room"
33    "github.com/tobiashbriones/dungeon-mst/client/world/room/room/room/room/room/room/room/room/room/room/room/room/room"
34    "github.com/tobiashbriones/dungeon-mst/client/world/room/room/room/room/room/room/room/room/room/room/room/room/room/room"
35    "github.com/tobiashbriones/dungeon-mst/client/world/room/room/room/room/room/room/room/room/room/room/room/room/room/room/room"
36    "github.com/tobiashbriones/dungeon-mst/client/world/room/room/room/room/room/room/room/room/room/room/room/room/room/room/room/room"
37    "github.com/tobiashbriones/dungeon-mst/client/world/room/room/room/room/room/room/room/room/room/room/room/room/room/room/room/room/room"
)

```

Terminal: Local (2) +

```

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/powershell

PS T:\dungeon-mst>

```

Figure 3: Cliente del juego

Como ya está corriendo el servidor, se pueden conectar los clientes:

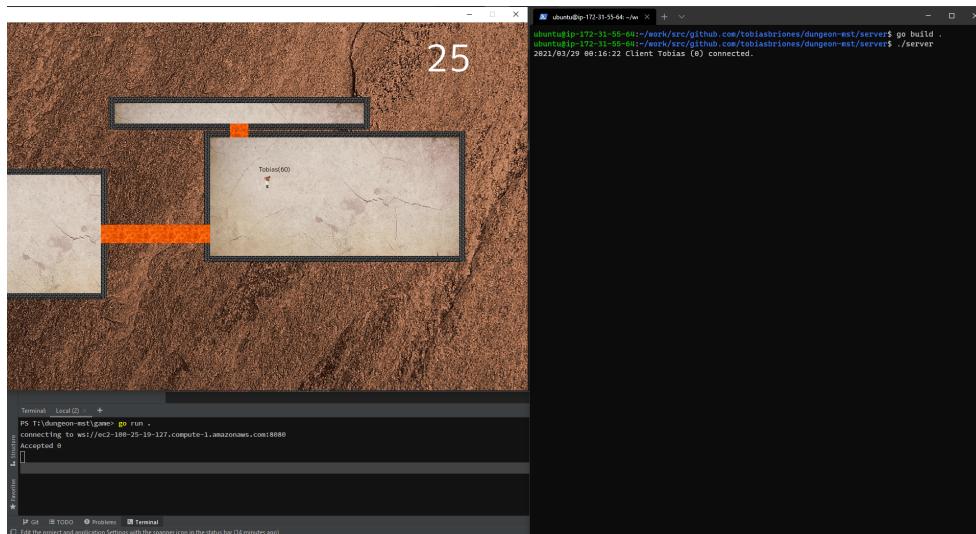


Figure 4: Juego

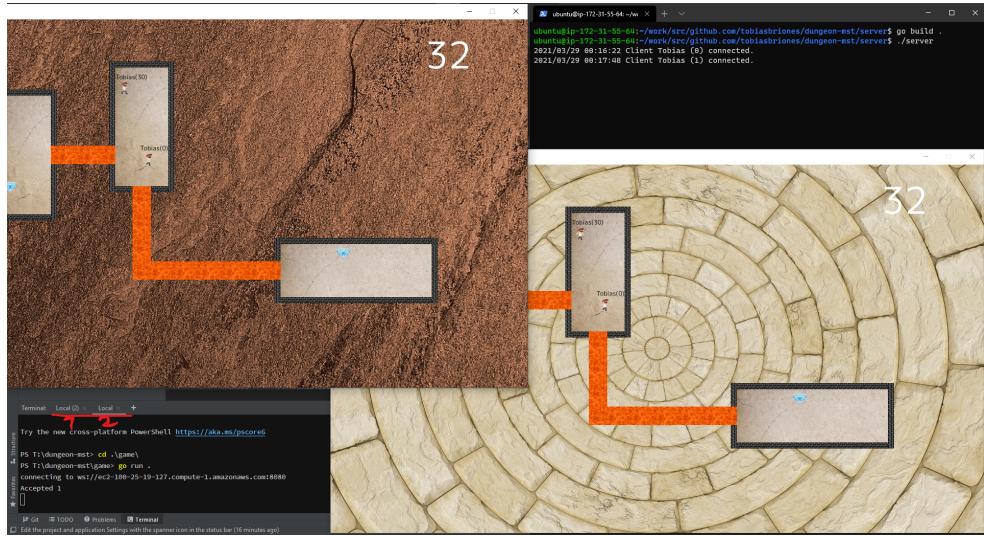


Figure 5: Juego

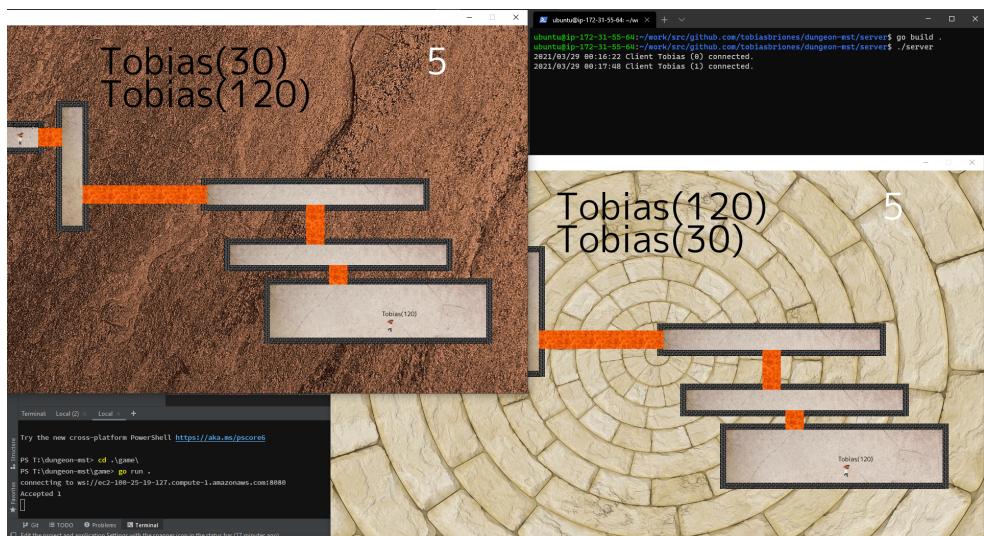


Figure 6: Juego

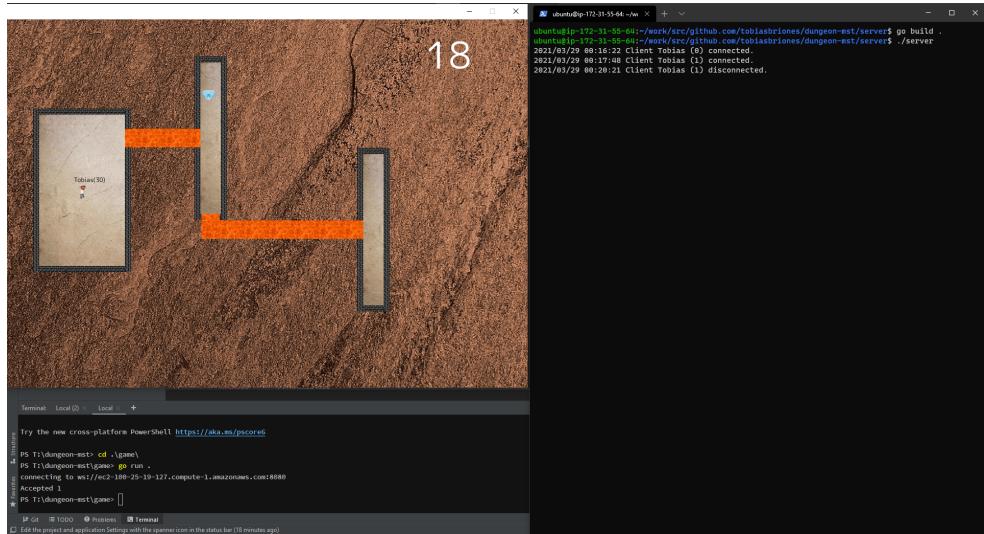


Figure 7: Juego

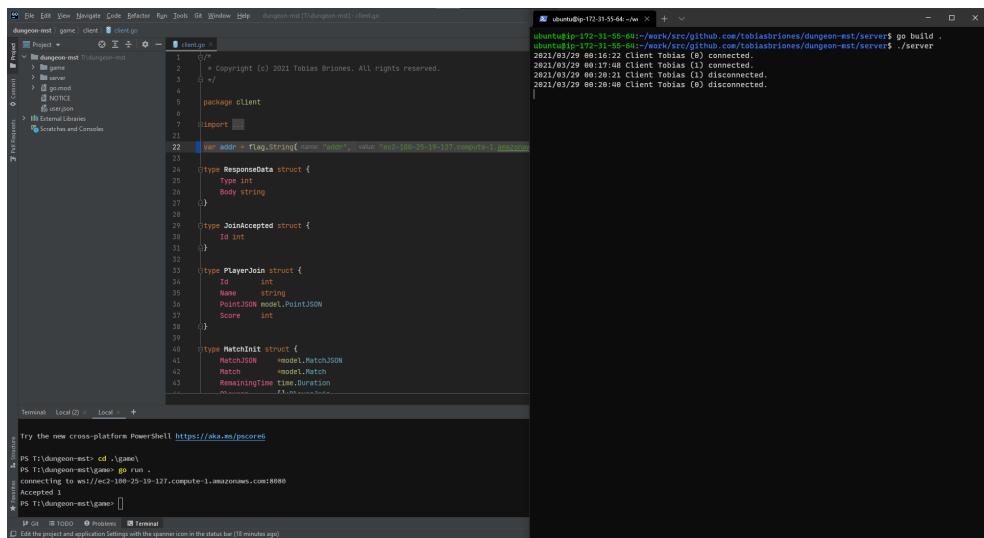


Figure 8: Juego

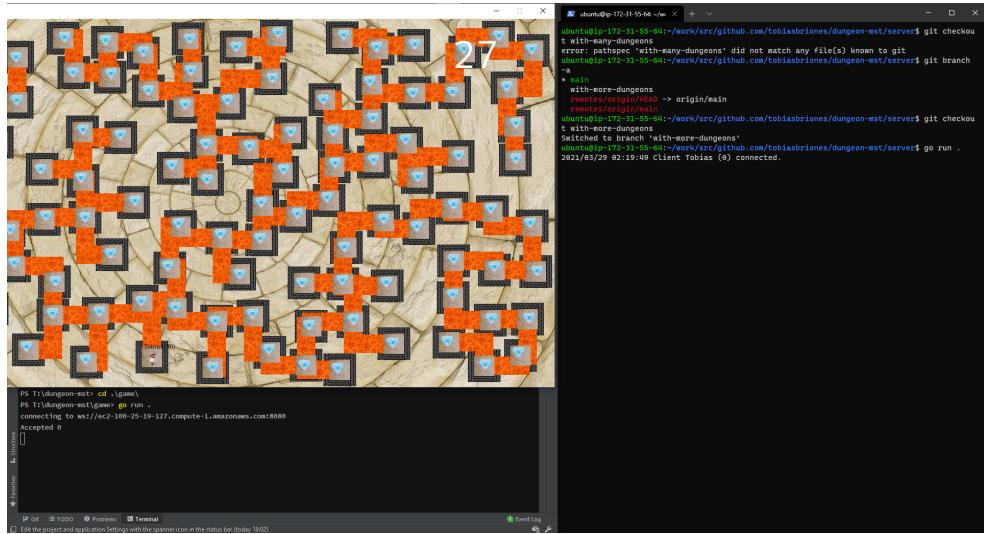


Figure 9: Generación de muchas masmorras