

Servidores de Videojuegos en Línea y el Megaservidor.

Memoria del Trabajo de la
Asignatura.

INGENIERÍA DE SERVIDORES
Universidad de Granada

Índice

Resumen.	3
Introducción.	4
Servidores y Juegos en Línea.	6
Conexiones peer to peer:	6
Servidores de Escucha:	7
Servidores Dedicados:	7
Rendimiento en Videojuegos en Línea - Tickrate.	8
Rendimiento en Videojuegos en Línea - Lag.	9
Técnicas de Compensación de Lag.	10
Desde el punto de vista del cliente:	10
Desde el punto de vista del servidor:	11
El megaservidor.	12
Anexo:	
¿Cómo alojar nuestro propio Servidor Dedicado?	14

Índice de figuras

1. Ejemplo de una partida de MUD en la que se produce un enfrentamiento entre guerreros clérigos y ladrones.	4
2. Ejemplo del juego habitat de 1986.	5
3. Ejemplo de una instancia concurrida en Ultima Online.	5
4. Conexiones entre los jugadores en un juego peer to peer.	6
5. Conexiones entre los jugadores en un donde existe un servidor.	7
6. El intervalo que pasa entre tick y tick es irregular.	8

7.	Conexión entre cliente y servidor donde el tickrate no está sincronizado.	8
8.	Arquitectura de un videojuego en línea simplificada.	9
9.	Extrapolación en el videojuego League of Legends.	10
10.	Diferentes Servidores del Videojuego World of Warcraft.	12
11.	Modo JcJ (Jugador contra Jugador) en Elder Scrolls Online, todos los personajes que se ven son jugadores conectados en el megaservidor.	13
12.	Distintas opciones que se nos dan al ejecutar el servidor.	14
13.	Pantalla principal de nuestro Servidor Dedicado.	16
14.	Configuración de los parámetros del juego.	16
15.	El servidor nos proporciona unas sencillas herramientas de monitorización.	17
16.	Podemos ver los jugadores que están conectados a nuestro servidor.	17
17.	El servidor nos permite banear diferentes jugadores por su IP o desde la lista de jugadores.	18
18.	Podemos ejecutar comandos de la consola del juego desde el servidor.	18

Resumen.

En este trabajo se estudiará el funcionamiento de los servidores en videojuegos en línea, y las diferentes opciones de las que se disponen.

Estudiaremos la historia y los orígenes de estos, remarcando aquellos videojuegos que trajeron innovaciones a este sector.

También estudiaremos las medidas de rendimiento más características en videojuegos, concretamente el tickrate, y el fenómeno que ocurre cuando este no se sincroniza correctamente, el Lag.

Así como el Lag se estudiarán las distintas técnicas que se utilizan para tratar de evitarlo o reducirlo, tanto desde el cliente, como desde el servidor.

Por último, se hablará sobre una innovadora tecnología surgida recientemente, el Megaservidor, y las implicaciones que tiene en el futuro de los videojuegos en línea.

Introducción.

Los videojuegos en línea son aquellos videojuegos jugados vía internet independientemente de la plataforma[1]. Estos aparecen siendo una gran parte de los videojuegos actuales, y existen tanto en consolas como en PCs como en dispositivos móviles, y se disponen en diversos géneros, como MMOs, en donde pueden participar, e interactuar en un mundo virtual, un gran número de jugadores del orden de cientos o miles simultáneamente, conectados de la red, normalmente internet.

En el año 1979 surgió la posibilidad de utilizar un ordenador para jugar en redes telemáticas, ocurrió en la universidad de Essex, donde un grupo de estudiantes crearon una versión informática del conocido juego de rol Dungeons & Dragons.

Así fue como surgió MUD (Multi User Dungeon) [2] un tipo de juego de rol en línea que se ejecuta en un servidor, este es la base de la que parten los actuales MMORPG. Esta versión electrónica era multiusuario y estaba basada en el uso de textos alfanuméricos. Este juego en principio databa de una suscripción muy cara (en torno a los 100\$ la mensualidad) ya que a parte de no usar la red de internet, porque no estaba aún establecida, usaba una red telnet de la universidad, por lo que el juego solo ofrecía servicio a una pequeña cantidad de usuarios.

```
Chedey y Argentum Arkano llegan de dentro
Descubres a Moriath Eleith escondida.
Moriath Eleith comienza a formular un hechizo.
Chedey busca algo en la habitación.
rebanar x0,x1,x2,x3,x4,x5,x6,x7,x8,x9,x10,x11,x12,x13,x14,x15,x16,x17,x18,x19,x20,Kizzo Lua
seguir x0,x1,x2,x3,x4,x5,x6,x7,x8,x9,x10,x11,x12,x13,x14,x15,x16,x17,x18,x19,x20,Kizzo Lua
Consigues escapar de la búsqueda por muy poco. Tu corazón palpita.
Te han descubierto!
Rebanas a Chedey con una serie de ataques rápidos!
Hieres gravemente a Chedey con tu Cimitarra de la Exhalación abrasando su Broche de Gestur. [16] Pv.
Levantas tu Cimitarra de la Exhalación sobre tu cabeza, y asestas un
violento golpe a Chedey produciéndole, graves quemaduras de ácido.
Abrapas a Chedey con tu Cimitarra de la Exhalación hundiéndola con saña en su Anillo de Vigia Jefe. [20] Pv.
Levantas tu Cimitarra de la Exhalación sobre tu cabeza, y asestas un
violento golpe a Chedey produciéndole, graves quemaduras de ácido.
Hieres gravemente a Chedey con tu Cimitarra de la Exhalación abrasando su Guanteletes del maestro guerrero. [16] Pv.
Hieres gravemente a Chedey con tu Cimitarra de la Exhalación abrasando su cuerpo. [18] Pv.
Abrapas a Chedey con tu Cimitarra de la Exhalación hundiéndola con saña en su Pantalón de Combate. [20] Pv.
Abrapas a Chedey con tu Cimitarra de la Exhalación hundiéndola con saña en su Pendientes del Héroe Perdido. [20] Pv.
Abrapas a Chedey con tu Cimitarra de la Exhalación hundiéndola con saña en su Gran Yelmo de Mithril. [20] Pv.
Hieres gravemente a Chedey con tu Cimitarra de la Exhalación abrasando su Mochila Bandolera. [17] Pv.
```

Figura 1: Ejemplo de una partida de MUD en la que se produce un enfrentamiento entre guerreros clérigos y ladrones.

Más adelante en 1986 se produjo otra mejora en este tipo de juegos con 'Hábitat' creado por Lucas Films Games fué el primer juego multiusuario que incorporó imágenes.



Figura 2: Ejemplo del juego habitat de 1986.

De aquí surgieron varios juegos, pero la verdadera revolución de los juegos en red surgió en el año 1993 junto a la creación de Internet. [3] Y con esto comenzó la necesidad de disponer de diferentes tecnologías avanzadas de servidores para poder alojar a la cantidad de clientes simultáneos que requerían los videojuegos que empezaron a surgir; entre estos, MMOs (Massive Multiplayer Online) con Ultima Online; Arena FPSs (First Person Shooter) con Doom; o RTSs (Real Time Strategy) con Command and Conquer.



Figura 3: Ejemplo de una instancia concurrida en Ultima Online.

Servidores y Juegos en Línea.

Los servidores en videojuegos (también llamados instancias o hosts) son servidores locales o remotos utilizados por clientes de videojuegos para jugar videojuegos multijugador[4]. El servidor transmite la suficiente información sobre su estado interno para permitir a los clientes que tenga conectados que puedan mantener una versión precisa del juego para mostrar a sus clientes, además de recibir y procesar la entrada del usuario. También estos servidores guardan la información del estado de cada jugador, según el videojuego que se trate (Por ejemplo, en algunos videojuegos los jugadores poseen un nivel y una serie de objetos y estadísticas que quedan guardadas en el servidor).

Según las necesidades concretas de cada título se pueden utilizar diferentes tipos de servidores o conexiones entre usuarios, se diferencian:

- **Conexiones peer to peer:**

En esta alternativa no existe servidor, es la alternativa más barata para la empresa distribuidora, cada peer recibe y envía información de y hacia todo el resto de peers[5].

Esta alternativa está obsoleta en videojuegos de acción pero se sigue utilizando en videojuegos de estrategia en tiempo real (RTS), ya que en ellos existen pocos usuarios y la información se puede dar en forma de órdenes (en lugar de enviar la posición de cada tropa, se envía la orden de dirigir esas tropas a la posición dada). Aunque esta alternativa tiene muchas desventajas, entre ellas, la imposibilidad de introducir un nuevo jugador a la partida, y la dificultad de sincronizar a todos los peers, que conlleva que todos los jugadores deben esperar al peer más lento, haciendo un efecto de cuello de botella. Para solucionar estos problemas se utilizan servidores.

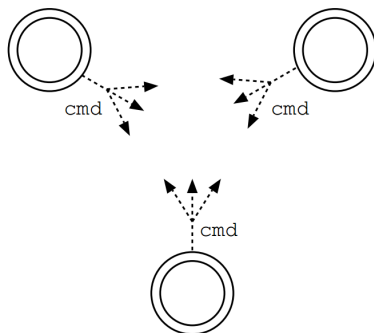


Figura 4: Conexiones entre los jugadores en un juego peer to peer.

■ Servidores de Escucha:

Los servidores de escucha son procesos que se ejecutan en una máquina cliente[6]. Funcionan como un pequeño servidor, en el cual uno de los clientes de la partida hace de host de este, es decir los jugadores se comunican con el jugador que actúa de host, y realizan peticiones con su servidor de escucha.

Servidores de escucha son utilizados en juegos de acción como Call of Duty y, aunque esto no supone coste a la empresa, y funciona de forma aceptable; tiene una serie de desventajas, ya que no es capaz de soportar una gran cantidad de jugadores, y el límite de jugadores suele de ser de 16 en videojuegos que utilizan esta alternativa. También el rendimiento se ve reducido por las limitaciones de la máquina que corre el servidor de escucha; además de que el jugador que actúa de host tiene una ventaja en la latencia con respecto al resto de jugadores, al ser su propia máquina el servidor.

■ Servidores Dedicados:

Los servidores dedicados simulan mundos de juego sin soportar una entrada o salida directa, excepto la requerida para su administración[7]. Sino que los jugadores deben conectarse con programas clientes específicos para poder ver e interactuar con el juego.

Estos servidores se pueden hostear en data centers profesionales, con todas las ventajas de rendimiento que eso conlleva. Además así se eliminan las ventajas mencionadas anteriormente de ser host de una partida multijugador. El único problema es que estos servidores cuestan dinero para crear y mantener, pero los costes muchas veces entran en el presupuesto de los desarrolladores, y a veces por grupos de jugadores (clanes) [8] ya que hay empresas que ofrecen pequeños servidores dedicados para diversos juegos por una serie de cuotas mensuales como por ejemplo <https://www.gameservers.com/>. La mayoría de juegos además de servidores dedicados ofrecen soporte a servidores de escucha.

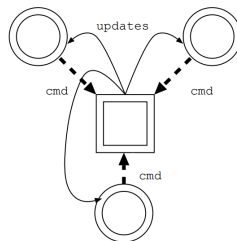


Figura 5: Conexiones entre los jugadores en un donde existe un servidor.

Rendimiento en Videojuegos en Línea - Tickrate.

Los videojuegos en línea, al requerir una gran cantidad de reflejos y un rendimiento muy estable para no causar frustración a los jugadores, tienen una serie de medidas específicas, siendo la más característica de estas el tickrate.

El tickrate es la frecuencia con la que un servidor ejecuta pasos de su simulación, siendo un “tick” un número asociado con un paso de simulación que se retransmite a los clientes para ayudarles a sincronizarse con el servidor.

Esta frecuencia se limita a un cierto máximo por tres razones:

- Para conservar ancho de banda del servidor y del cliente.
- Para conservar uso de CPU.
- Para permitir a los clientes saber cuánto tiempo pasa entre cada tick, esto es importante para los juegos en línea, ya que las actualizaciones desde el servidor pueden llegar en intervalos distintos o desordenados.

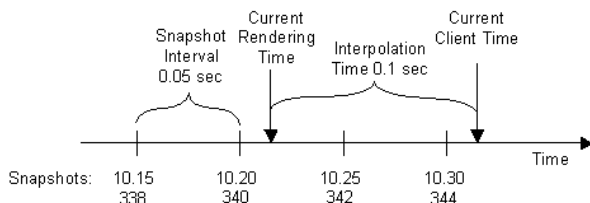


Figura 6: El intervalo que pasa entre tick y tick es irregular.

Cuando el tickrate del cliente y el tickrate del servidor no están sincronizados, por ejemplo, porque la máquina del cliente no es capaz de soportar los requerimientos de red o del juego para conectarse correctamente, se produce el fenómeno conocido como lag, del que hablaremos a continuación.

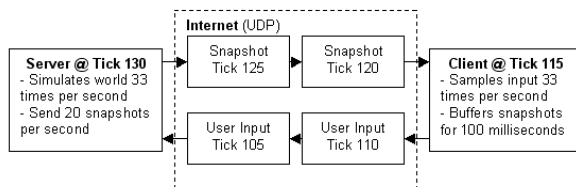


Figura 7: Conexión entre cliente y servidor donde el tickrate no está sincronizado.

Rendimiento en Videojuegos en Línea - Lag.

El lag en videojuegos online es cuando se produce un retraso visible entre la acción del jugador y la respuesta del servidor.[11]

Este fenómeno suele ser frustrante especialmente en juegos de acción, donde la velocidad y los reflejos son clave.

Se causa ya que mientras un videojuego reproduce una copia local del juego, un videojuego en línea requiere ser replicado a partir de un servidor central para evitar inconsistencias entre clientes, por lo que el cliente no tiene control directo sobre el estado del juego y sólo puede enviar peticiones de cambio al servidor, y así actualizar su copia local recibiendo actualizaciones del servidor, y esta necesidad de comunicación causa un retraso entre los clientes y el servidor. Hay tres posibles causas de lag[12]:

- **Rendimiento pobre de la red.**
- **Insuficiente capacidad de procesamiento del cliente.**
- **Insuficiente capacidad de procesamiento del servidor,** que es la que nos interesa, ya que estamos estudiándolo desde el punto de vista del servidor y ocurre cuando el usuario realiza un pedido que no se encuentra en la caché y la respuesta requiere una gran cantidad de procesamiento antes de poder ser enviada. Esto afecta generalmente en las horas de mayor actividad donde el servidor se ve exigido por tener que procesar la información de una enorme cantidad de usuarios conectados.

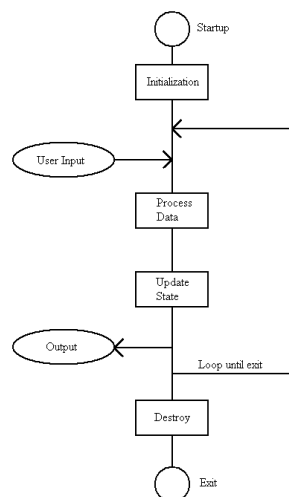


Figura 8: Arquitectura de un videojuego en línea simplificada.

Técnicas de Compensación de Lag.

Existen diversos métodos para reducir o camuflar el Lag, algunos de estos aplicables desde el punto de vista del cliente y otros desde el lado del servidor.

Desde el punto de vista del cliente:

A los clientes no se les permite cambiar el estado del juego, sino que deben recibirlo del servidor, pero sí que puede tratar de mostrar el juego de la forma más precisa posible[13], y para ello es necesario realizar predicciones, para ello existen dos métodos:

- **Interpolación**, que funciona guardando en buffer el estado del juego, y renderizándolo con un ligero y constante retraso, así cuando un paquete llega del servidor, en lugar de actualizar la posición de un objeto inmediatamente, el cliente interpolará su posición, comenzando por su última posición conocida. Durante ese periodo de interpolación, el objeto se moverá de forma suave entre las dos posiciones, idealmente este intervalo será el mismo que el retardo entre paquetes, pero este es raramente el caso.
- **Extrapolación**, es un intento de estimar el estado futuro del juego. Tan rápido como llega un paquete nuevo, la posición del objeto se actualiza a la nueva posición, y mientras espera al siguiente, la siguiente posición se extrapola basándose en la posición actual y el movimiento en el momento de la actualización. Esencialmente el cliente asume que un objeto se continuará moviendo en la misma dirección, y cuando un nuevo paquete llega la posición se tendrá que corregir ligeramente, ya que no se puede predecir el movimiento exacto que se dará. Un ejemplo de esto es la opción de predicción de movimiento del videojuego League of Legends[14].

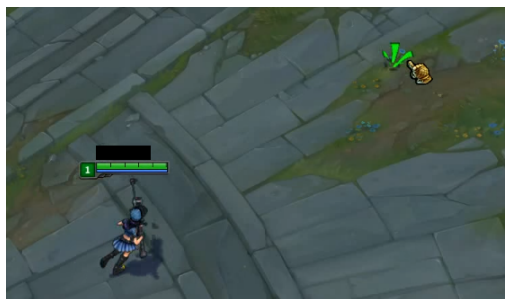


Figura 9: Extrapolación en el videojuego League of Legends.

Desde el punto de vista del servidor:

A diferencia de los clientes, el servidor si conoce el estado exacto del juego, y dicha predicción no es necesaria. El propósito principal de la compensación desde el servidor es proporcionar efectos precisos de las acciones de los clientes. Esto es importante porque entre que un cliente realiza una acción y esta llega pasa un tiempo, y por tanto el juego ya no estará en el estado en el que el jugador efectuó la acción. Un ejemplo de esto es la detección de impacto en los FPS (First Person Shooters) en los que los márgenes son pequeños y causarían errores si no se maneja de forma correcta.

Posibles soluciones son:

- **No hacer nada:**

Una posible solución es ignorar el problema, en videojuegos como RTSs, esta solución es posible ya que es posible un retardo entre que el jugador comanda una acción y esta se produce sin resultar frustrante, en FPSs hasta el más mínimo retardo causaría frustración.

- **Rebobinar el tiempo:**

Otra solución sería guardar el estado de la partida cuando el jugador efectuó la acción, y rebobinar al estado en el que se encontraba cuando se procesa esta. Para rebobinar un tiempo apropiado para por ejemplo determinar adónde apuntaba el jugador cuando disparó se usa la latencia del jugador. Aunque esto resultará en que el juego verá al jugador disparando a una posición donde ya no está el objetivo y dando, lo cual puede resultar confuso para el desarrollo de la partida.

- **Confiar en los clientes:**

Es posible permitir a los clientes decir lo que están haciendo y que el servidor confíe en esa información que recibe. Aunque este método es susceptible a trampas, ya que los jugadores podría enviar información falsa al servidor, mediante mensajes externos hacia él, con software de trampas. Aunque esta técnica se utiliza en juegos cuya escala hace técnicas como rebobinado imposibles, como por ejemplo Battlefield 3[15], donde el sistema sólo hace un test en el que ve si es posible que el disparo fuera acertado según la posición nueva.

- **Obligar a los clientes a extrapolar:**

Una solución menos común sería hacer a los clientes extrapolar para cubrir su latencia, aunque esto produce resultados incorrectos a no ser que los clientes se muevan de forma constante, dando ventaja a aquellos que se muevan de forma inconsistente, ya que el resto de jugadores los verían en posiciones distintas a donde realmente están[16].

El megaservidor.

En el género de los MMOs, donde se pretende crear un mundo persistente donde los jugadores interactúan, es donde la tecnología de servidores debe ser más avanzada para así permitir una gran cantidad de jugadores en un mismo lugar[18].

Tradicionalmente, estos juegos separan sus jugadores en distintos servidores independientes (reinos), donde los jugadores y sus datos se guardan. Esta tecnología limita al videojuego en muchísimos aspectos, ya que los jugadores de diferentes reinos no pueden interactuar entre sí.



Realm Name	Type	Characters	Population
Dragonmaw (4.0.3)	(PVP)	(3)	Medium
Black Dragonflight (4.0.3)	(PVP)	(1)	High
Bonechewer (4.0.3)	(PVP)	(1)	Medium
Borean Tundra (4.0.3)	Normal		New Players
Mok'Nathal (4.0.3)	Normal		New Players
Muradin (4.0.3)	Normal		New Players
Nesingwary (4.0.3)	Normal		New Players
Quel'dorei (4.0.3)	Normal		New Players
The Underbog (4.0.3)	(PVP)		New Players
Aegwynn (4.0.3)	(PVP)		Low
Agamaggan (4.0.3)	(PVP)		Low
Andorhal (4.0.3)	(PVP)		Low
Anetheron (4.0.3)	(PVP)		Low
Anub'arak (4.0.3)	(PVP)		Low
Auchindoun (4.0.3)	(PVP)		Low

Figura 10: Diferentes Servidores del Videojuego World of Warcraft.

Por ello se crea la tecnología de megaservidor, donde los jugadores no se dividen en reinos sino que todos se encuentran en un mismo grupo. Esta tecnología fue introducida por Zenimax Online, desarrolladora detrás de Elder Scrolls Online, introducida en este mismo título, estrenado en 2014; y ha sido adoptada recientemente por ArenaNet, en Guild Wars 2 y por Carbine, en Wildstar.

El megaservidor es esencialmente un sistema de servidores donde el juego completo está hosteado en un sistema con múltiples instancias, en lugar de en diferentes servidores con instancias individuales[19].

En un MMO tradicional, cada uno de los reinos son iteraciones completas y únicas del videojuego, todos los jugadores de cada reino existen en el mismo mundo y tiempo cada instante. Y cada uno de estos reinos está completamente desconectado de todos los demás, si todos los jugadores de un reino deciden ir al mismo lugar, todo el mundo aparecería en ese lugar hasta que la zona no pudiera soportar más jugadores y negara el acceso o crasheará.

Con el sistema de megaservidor, todos los servidores individuales se enlazarían dinámicamente en un único array de servidores[20]. Y los jugadores que entran entran a ese único servidor en lugar de en servidores completamente separados, y así conforme los jugadores van entrando, en lugar de denegar el acceso, se crearía una nueva instancia del mundo para ir llenando con los jugadores que vayan llegando; así cuando un jugador entra en una zona que está llena, simplemente se crea una nueva instancia de esta zona.

Estas instancias, a diferencia de los servidores independientes, permiten a los jugadores intercambiarse entre ellas, de forma que puedes jugar con cualquier jugador, simplemente cambiando de instancia, ya que el megaservidor tiene enlazados los servidores físicos.

En el megaservidor las instancias en lugar de ser instancias del juego completo, son instancias de cada una de las zonas, y cada vez que uno cambia de zona, cambiará también de instancia, estando cada una de estas instancias alojadas en un servidor físico, en el caso de Elder Scrolls Online, el sistema posee dos megaservidores, uno localizado en Europa y otro en Estados Unidos, de esta forma no se necesita cambiar a diferentes servidores para evitar lag o jugar con tus amigos.

De todas formas, ya que no es posible que todos los jugadores coexistan en el mismo espacio, el servidor crea las diferentes instancias, y usa un sistema de canales para unir a personas por su localización o idioma. El megaservidor no es un servidor físico único, sino una red de servidores combinados formando una unidad lógica única.



Figura 11: Modo JcJ (Jugador contra Jugador) en Elder Scrolls Online, todos los personajes que se ven son jugadores conectados en el megaservidor.

Anexo:

¿Cómo alojar nuestro propio Servidor Dedicado?

Para instalar un servidor dedicado del juego que deseemos, necesitamos que se nos proporcione un ejecutable con el servidor a ejecutar. En nuestro caso crearemos un servidor de Counter Strike, por lo que el ejecutable será *hlds.exe* (half-life dedicated server), que se nos proporciona junto a los archivos del videojuego.

Realizaremos este proceso en Windows Server.

En primer lugar descargamos la aplicación de <http://cs.gamebanana.com/gamefiles/3011> ya que la versión oficial perdió soporte en 2013.

Tras instalar el servidor, tendremos los archivos del juego en nuestro disco duro, así como el ejecutable del servidor, ya que este servidor es de un juego bastante antiguo, las especificaciones no son muy altas, se recomienda tener una velocidad de subida de 128kbps como mínimo y una memoria RAM de al menos 256MB para alojarlo.

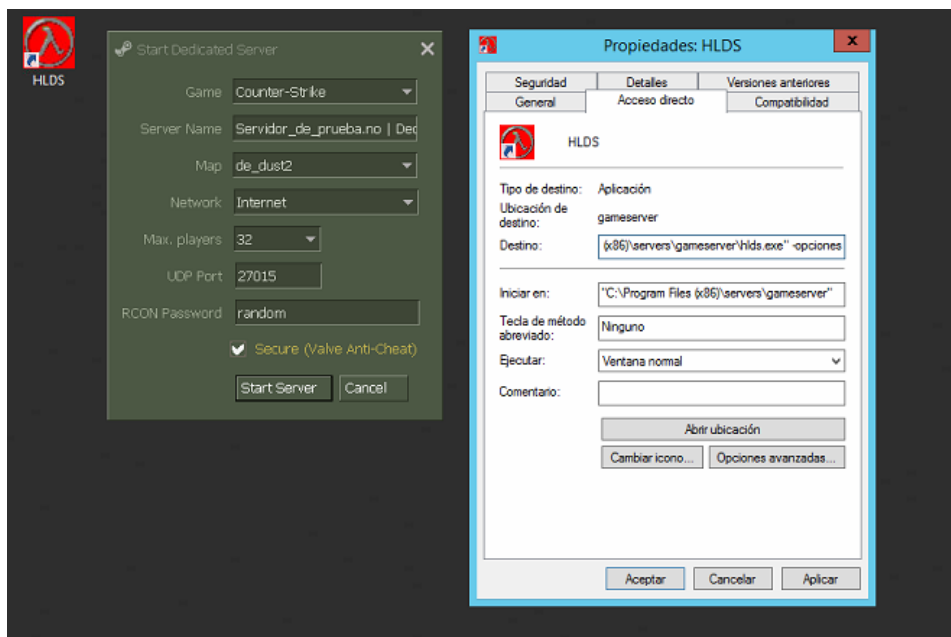


Figura 12: Distintas opciones que se nos dan al ejecutar el servidor.

Las opciones de las que disponemos son:

- El juego, que será Counter-Strike en nuestro caso.
- El nombre del servidor.
- El mapa del juego concreto.
- La red que utilizará: LAN o Internet.
- El número máximo de jugadores.
- El puerto UDP que utilizará nuestro servidor.
- La contraseña de acceso como administrador.
- Si la seguridad antitrampas se activará o no.

Las opciones del lanzador serán:

- **-nomaster**: Para deshabilitar la autenticación del servidor desde Valve, la empresa que lanzó el juego.
- **-heapsize [KBs]**: Asignación de RAM al servidor en KBs.
- **+sv_lan [0/1]**: Determina si se el servidor se ejecutará en la red local o en internet.
- **+exec server.cfg**: Ejecuta el archivo de configuración, donde se pueden modificar parámetros de la partida concretos al juego (tiempo de ronda, dinero inicial. . .).
- **+ip [IPnumber]**: Establecemos la IP que tendrá nuestro servidor, que será la IP de nuestra máquina, la podemos mirar usando ipconfig.

En mi caso esta línea será:

```
"C:/Program Files (x86)servers/gameservers/hlds.exe" -nomaster  
-heapsize 256000 +ip 192.168.147.131 +sv_lan 0 +exec server.cfg
```

Iniciamos el servidor pulsando en Start Server y nuestro servidor quedará iniciado.

A continuación mostraré diferentes capturas de las opciones que se nos dan para administrar y monitorizar el servidor.

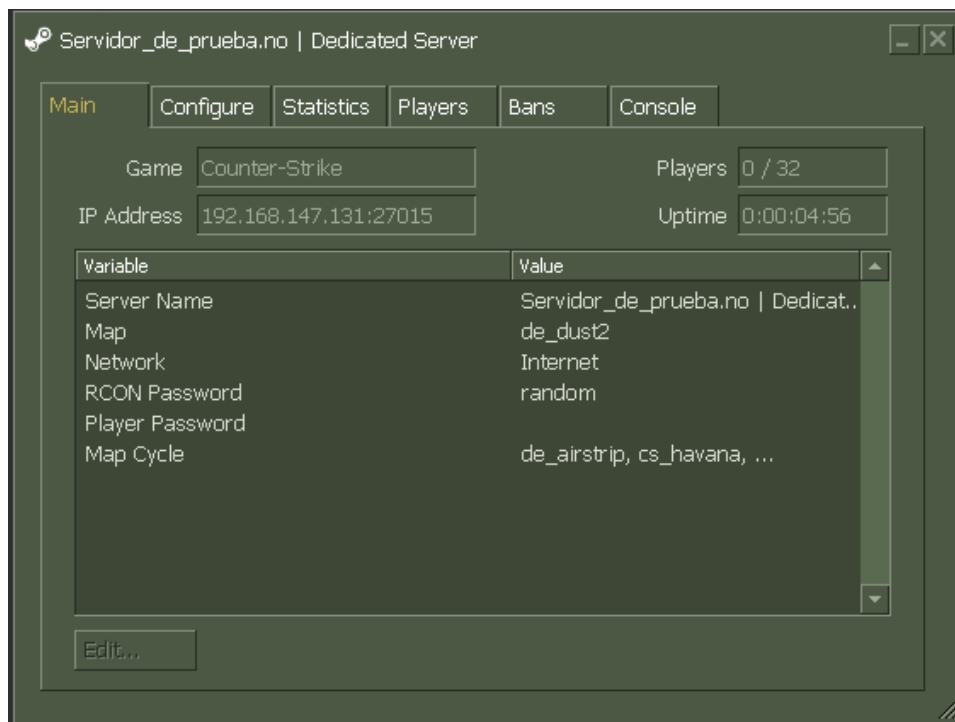


Figura 13: Pantalla principal de nuestro Servidor Dedicado.



Figura 14: Configuración de los parámetros del juego.

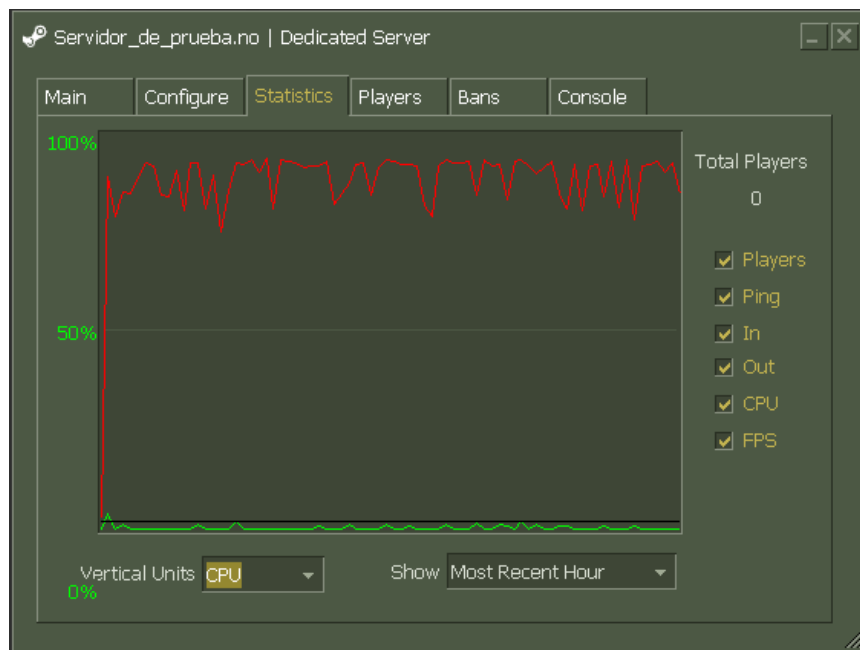


Figura 15: El servidor nos proporciona unas sencillas herramientas de monitorización.

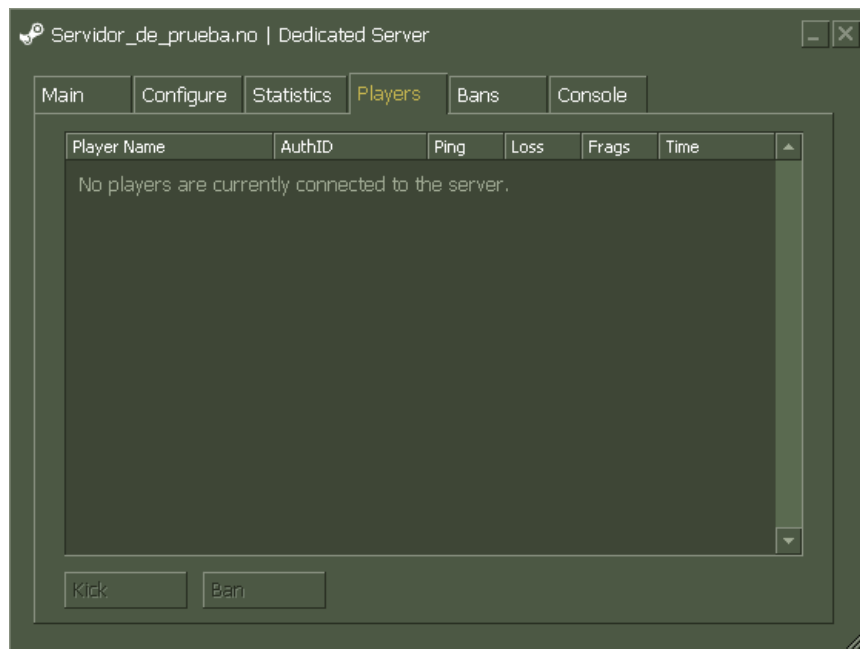


Figura 16: Podemos ver los jugadores que están conectados a nuestro servidor.

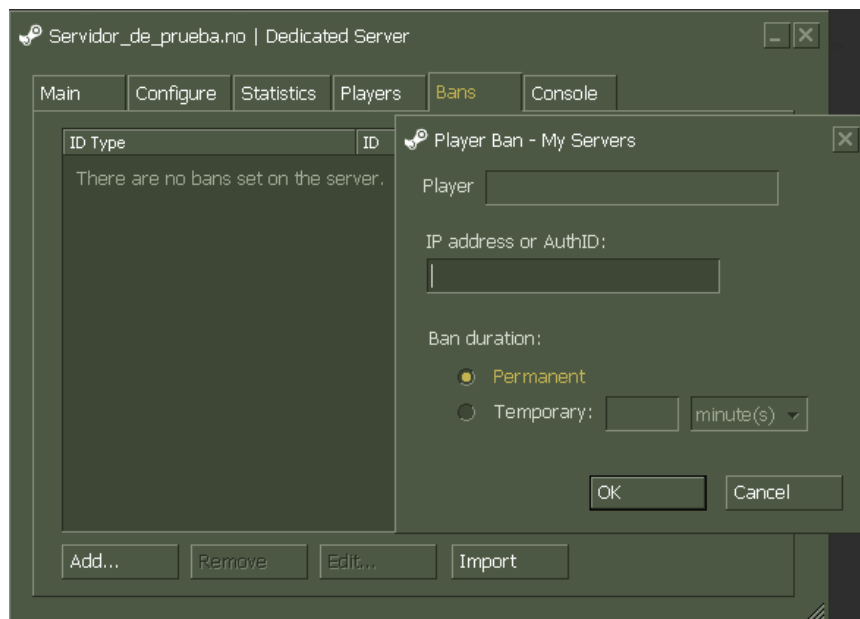


Figura 17: El servidor nos permite banear diferentes jugadores por su IP o desde la lista de jugadores.

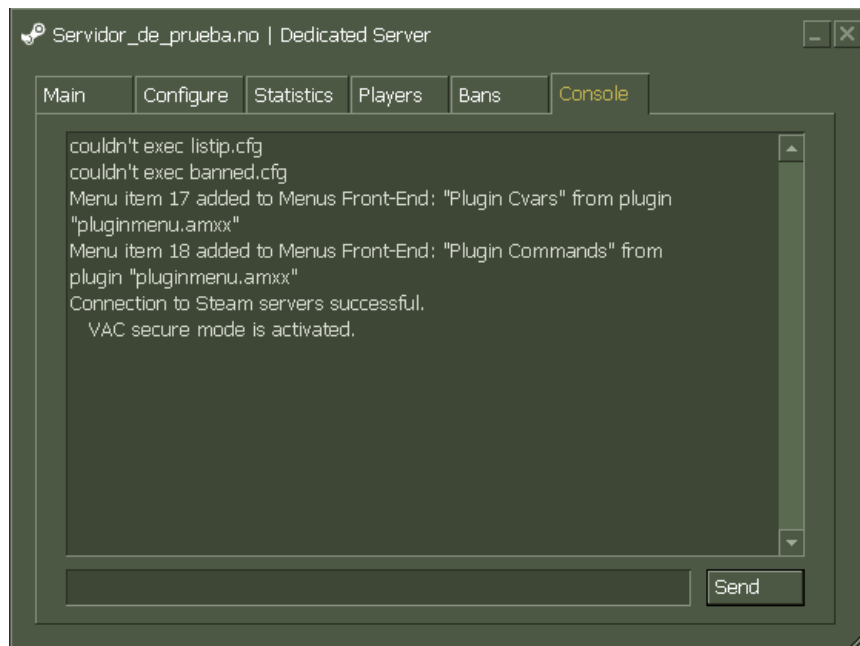


Figura 18: Podemos ejecutar comandos de la consola del juego desde el servidor.

Referencias

1. <http://www.wisegeek.com/what-is-online-gaming.htm>
2. <http://lambda.moo.mud.org/pub/M00/papers/mudreport.txt>
3. <http://www.pcmag.com/article2/0,2817,2390917,00.asp>
4. https://en.wikipedia.org/wiki/Game_server
5. <http://gafferongames.com/networking-for-game-programmers/what-every-programmer-needs-to-know-about-game-networking/>
6. <https://community.callofduty.com/thread/200621388>
7. <http://www.gdcvault.com/play/1015337/Dedicated-Servers-In-Gears-of>
8. <http://www.techradar.com/news/gaming/how-to-set-up-a-dedicated-games-server-1053429>
9. <http://battlelog.battlefield.com/bf4/news/view/addressing-netcode-in-bf4/>
10. https://developer.valvesoftware.com/wiki/Source_Multiplayer_Networking
11. <https://en.wikipedia.org/wiki/Lag>
12. <http://www.eecg.toronto.edu/~ashvin/courses/ece1746/2003/reading/cronin-umtr01.pdf>
13. https://developer.valvesoftware.com/wiki/Latency_Compensating_Methods_in_Client/Server_In-game_Protocol_Design_and_Optimization#Lag_Compensation
14. <http://forums.na.leagueoflegends.com/board/showthread.php?t=2004658&page=1#post22988898>
15. https://www.reddit.com/r/battlefield3/comments/n2oiy/we_need_someone_to_create_a_guide_for_the_new/c35xc2m
16. <http://forums.tripwireinteractive.com/showpost.php?p=662931&postcount=9>
17. https://www.reddit.com/r/gamedev/comments/1w746u/interested_in_mmo_server_architecture/
18. http://www.mmorpg.com/blogs/FaceOfMankind/052013/25185_A-Journey-Into-MMO-Server-Architecture
19. <http://www.tesoelite.com/2013/04/elder-scrolls-online-mega-server-explained/>
20. <https://forums.bladeandsoul.com/forums/index.php?/topic/134194-carbine-not-really-using-megaserver-technology/>