

ANÁLISIS DE LAS FUNCIONALIDADES MULTIJUGADOR DE OVERWATCH 2

Daniel Arroyo Cano



OVERWATCH 

The word "OVERWATCH" is written in a bold, dark grey sans-serif font. The letter "O" has a small white triangle cutout on its left side. To the right of "OVERWATCH" is a stylized letter "Z" enclosed in an orange square.

Evaluación continua de teoría, trabajo libre

Aplicaciones en red y multijugadores

Grado en Diseño de Videojuegos, curso 2025-2026

Universidad de Burgos (UBU)

ÍNDICE

ÍNDICE	2
INTRODUCCIÓN.....	3
¿Qué es Overwatch 2?.....	3
Blizzard: su desarrolladora	5
Uso de cliente y modalidades de juego	7
Algoritmos de emparejamiento y matchmaking para el juego competitivo	9
CARACTERÍSTICAS TÉCNICAS PRINCIPALES	12
Motor y lenguaje.....	12
Desarrollo.....	13
Arquitectura de red y protocolos.....	15
Medidas de seguridad	18
Sistema antitrampas y de reportes.....	19
DESAFÍOS TÉCNICOS	20
Número de servidores y multijugador masivo	20
Frenetismo del videojuego y gestión de la latencia	21
Cargado a tiempo real de mapas, modelos, texturas y efectos especiales.....	21
Optimización multiplataforma.....	23
COMPARACIÓN CON TÍTULOS SIMILARES.....	24
Marvel Rivals: uso de un motor gráfico menos personalizado.....	24
Paladins: uso de un sistema antitrampas externo.....	25
Valorant: mayor tasa de ticks por segundo	26
CONCLUSIÓN	27
BIBLIOGRAFÍA Y REFERENCIAS	28
Documentación general sobre la estructura de red	28
Documentación acerca de la programación y desarrollo técnico	28
Documentación acerca del apartado gráfico.....	29



INTRODUCCIÓN

¿Qué es Overwatch 2?

Overwatch 2 es un videojuego **hero shooter multijugador en línea a nivel global**, publicado originalmente por **Blizzard** el 24 de mayo de 2016 para PC a través del lanzador Battle.net, PlayStation 4 y Xbox One. Tiempo después, pasaría a renombrarse como Overwatch 2, conservando la mayor parte del núcleo jugable del título original y publicándose bajo este nuevo nombre el **4 de octubre de 2022**. Hoy en día, está disponible para todas las plataformas anteriores, a las cuales se añaden Nintendo Switch y Steam. En todas estas plataformas se trata de un **videojuego gratuito**, financiado a través de microtransacciones de aspectos y elementos cosméticos.

Los videojuegos del género **hero shooter** se caracterizan por hacer enfrentar a **dos equipos** en modalidad multijugador en línea. Estos equipos deberán tratar de conseguir un **objetivo definido por el mapa**. Sin embargo, las herramientas para lograrlo se basan en **personajes con armas y habilidades propias**, mediante las cuales deberán buscar eliminar a los jugadores del equipo rival para poder obtener ventaja numérica o posicional y **lograr dichos objetivos**.

Así, las partidas clásicas multijugador de Overwatch 2 también consisten en dos equipos, de cinco miembros cada uno, que deben competir en un mapa cerrado para lograr un objetivo determinado, dependiente del mapa. Para ello, los jugadores de ambos equipos tienen que organizarse para cubrir los **roles de tanque, daño y apoyo**, y utilizar a variados personajes acordes a cada rol. Aparte del rol, cada personaje cuenta con sus estadísticas, armas, habilidades, fortalezas y debilidades determinadas, añadiendo una capa más de estrategia en equipo, además de un componente de individualidad y expresión creativa personal.



Imagen 1, captura de pantalla de una partida de Overwatch 2.

El elemento diferenciador de Overwatch 2 respecto a otros **hero shooters**, aparte de su reconocida ambientación futurista, sus característicos mapas y sus originales personajes (en diseño, lore y jugabilidad), se trata de que **los jugadores pueden cambiar de personaje libremente en mitad de las partidas cuantas veces quieran**, llevando al límite el trabajo en equipo, sinergias en las composiciones y combinaciones o, simplemente, apelando a la flexibilidad y comodidad de los jugadores en todo momento.

Overwatch 2 también se diferencia de otros videojuegos similares gracias a su **vista en primera persona** para todos los jugadores, que premia su habilidad individual, entendimiento del espacio y su **puntería**, elemento central de todos los videojuegos **shooter**.

Esta jugabilidad propia, sumado a un desempeño técnico envidiable para cualquier videojuego multijugador en línea masivo, le valió a Overwatch para ganar el premio a **Mejor juego del año en The Game Awards en 2016**, siendo el primero de estas características en lograrlo. Después de eso, el videojuego ha continuado siendo uno de los más influyentes dentro del mercado de videojuegos multijugador, contando con actualizaciones regulares hasta hoy en día, incluso con el cambio de nombre y actualización general que se produjo con el cambio a Overwatch 2. Se trata, a su vez, de uno de los títulos **referentes en el mundo de las competiciones de deportes electrónicos**.



Imagen 2, portada conmemorativa de Overwatch para celebrar su premio a "Mejor juego del año" en The Game Awards 2016.

A día de hoy, a pesar de haber diversificado su público a través de muchas plataformas distintas, continúa siendo uno de los videojuegos como servicio más jugados en todo el mundo. Solo en Steam, considerando solo una fracción de su público total, se estima que unos **diez millones de jugadores** lo tienen instalado en su cuenta, siendo unos **cincuenta mil de ellos recurrentes y diarios solo en esta plataforma**. Eso sí, con cada actualización se producen picos de jugadores, curiosos por descubrir las novedades del videojuego (modos de juego nuevos, nuevos mapas, nuevos héroes, cambios de balance o temporadas temáticas).



Imagen 3, gráficas y estadísticas de jugadores de Overwatch 2 en Steam. Información extraída de SteamDB.

En la plataforma principal y original de juego, el lanzador de Blizzard Battle.net, el número de jugadores se dispara, aunque se traten de estimaciones realizadas por terceros. Aún así, estos datos hacen gala del robusto sistema multijugador del que consta el videojuego, a pesar de las cantidades masivas de jugadores a nivel mundial o los grandes picos repentinos cuando se producen actualizaciones.

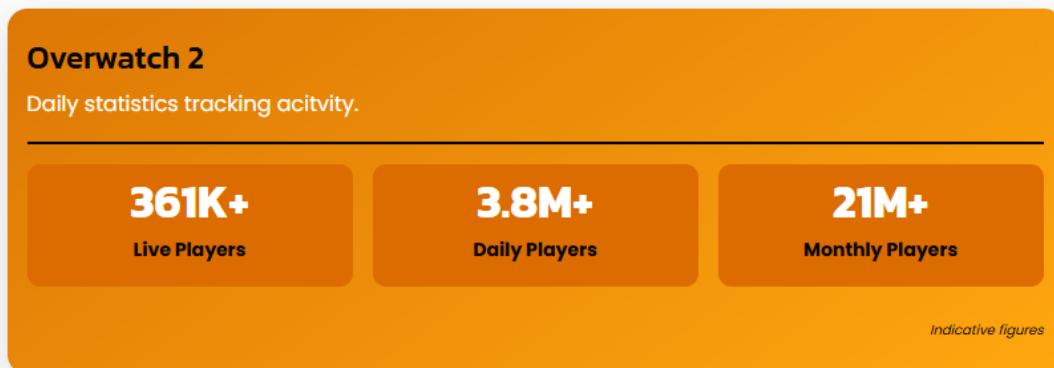


Imagen 4, estimación de la cantidad total de jugadores de Overwatch 2 a nivel global, teniendo en cuenta todas las plataformas de juego posibles.

Blizzard: su desarrolladora

Activision-Blizzard (anteriormente Blizzard) es la empresa estadounidense desarrolladora detrás de Overwatch 2. Se trata de una de las **empresas más reconocidas dentro de la industria del videojuego**. Posee a sus espaldas títulos muy reconocidos como World of Warcraft, Hearthstone, Diablo o Starcraft, que han sido muy influyentes y jugados en todos los rincones del planeta. De hecho, un componente común en estos videojuegos es que todos se tratan de grandes éxitos multijugador y masivos a nivel

mundial, por lo que Blizzard es una de las empresas referentes en este aspecto, sobre todo a nivel técnico y de gestión de grandes estructuras en red.

El caso de Overwatch 2 no ha sido diferente, y Blizzard ha aprovechado su experiencia en videojuegos multijugador para pulir aspectos que fallaban en anteriores títulos suyos: la mala gestión de picos altos de jugadores en World of Warcraft, los cambios de balance que apagaron la comunidad en Starcraft o la gestión de combates y momentos frenéticos dependientes de la latencia en Diablo han servido como antecedentes y aprendizaje para mejorar el acabado técnico de Overwatch 2.

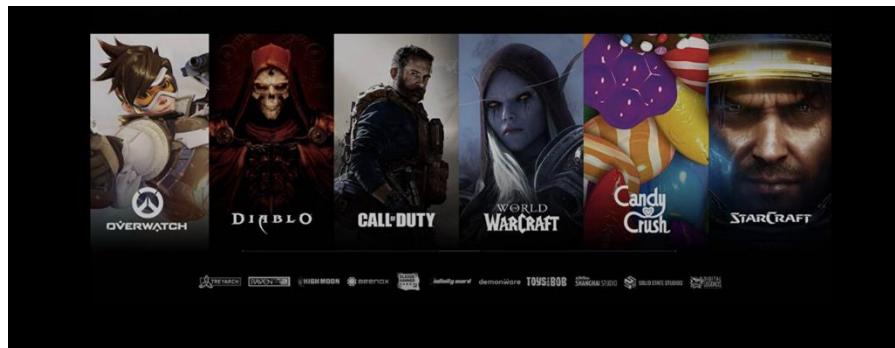


Imagen 5, portada que muestra las principales franquicias propiedad de Activision-Blizzard.

A parte de servir como ejemplos anteriores en el desarrollo de Overwatch 2, todos los videojuegos multijugador de Blizzard comparten varias características comunes, que influyen en su gestión de las redes y la seguridad. Estos videojuegos son todos **descargados y lanzados desde Battle.net**, el **launcher oficial de Blizzard** destinado a recopilar todos sus videojuegos, aparte de funcionar como centro de información, actualizaciones y lanzador de juego para una misma cuenta de usuario. Esto limita los métodos de descarga y control del usuario, pero a su vez facilita el control de versiones y actualizaciones, estandarizándolos para todos los jugadores. Los videojuegos de Battle.net también comparten un mismo sistema de seguridad, impidiendo la brecha de datos de las cuentas de jugadores y a su vez facilitando el control antitrampas dentro de los juegos.

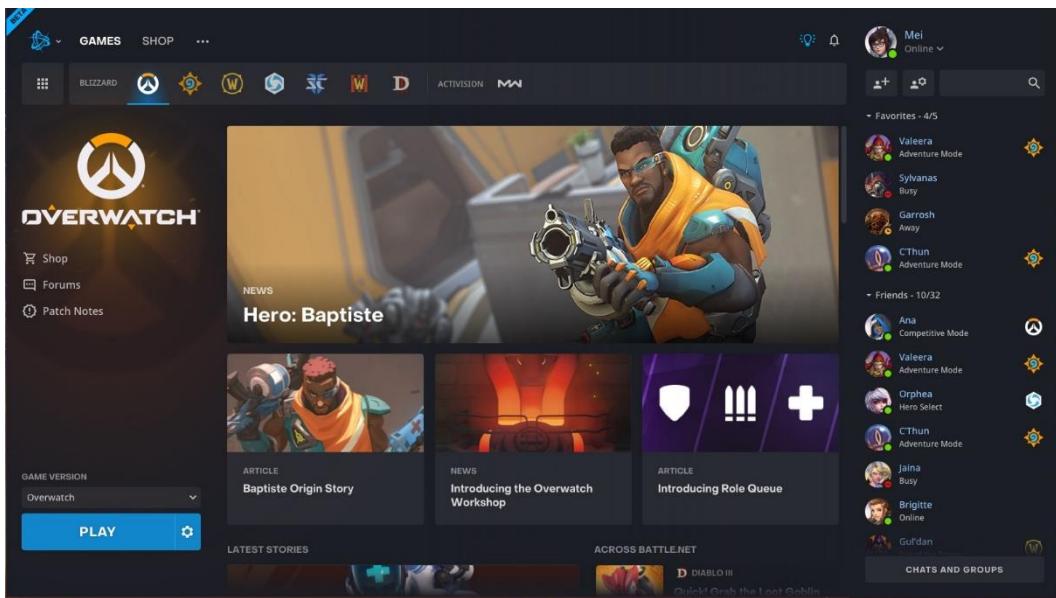


Imagen 6, captura de pantalla del lanzador de Overwatch dentro de Battle.net.

Uso de cliente y modalidades de juego

La aplicación de cliente Battle.net funciona como el centro de operaciones y el **sistema de cuentas obligatorio para Overwatch 2**, gestionando inicios de sesión, juego multiplataforma, selección de regiones de juego, listas de amigos y actualizaciones. Sin embargo, sus conexiones de red pueden causar problemas específicos, como desconexiones o alta latencia para los jugadores. A menudo, es necesario solucionar problemas como cambiar el DNS, usar una VPN o reiniciar el enrutador, a diferencia de los problemas comunes de Internet. Battle.net determina la ruta de conexión hacia Overwatch 2, lo que significa que los problemas con los servidores de Battle.net (pertenecientes a los servidores centrales de Blizzard) o sus rutas de red afectan directamente la experiencia de juego. Son dos entes inseparables en la estructura de red, correspondiéndose esta primera parte del lanzador del juego con una **red cliente – servidor**, donde la parte del cliente se trata de una cuenta de usuario determinada.

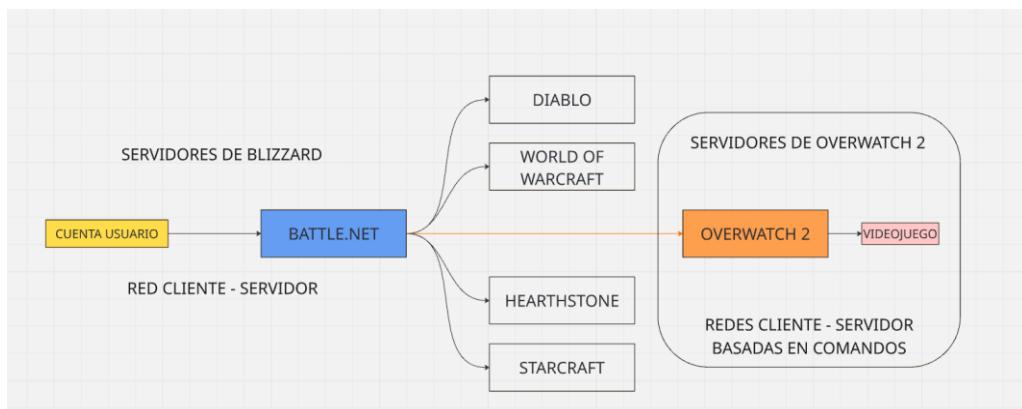


Imagen 7, esquema representativo de la estructura de red de Overwatch 2, separando servidores de Blizzard y servidores específicos del videojuego.



Una vez dentro de Overwatch 2, el jugador continúa actuando como cliente, pero esta vez en **redes basadas en comandos**. Dentro del videojuego, aparte de los catorce servidores regionales disponibles en los que se ubica al jugador en uno, encontramos **servidores dedicados destinados a funciones concretas**, como el almacenaje de partidas o la tienda de cosméticos y microtransacciones (conectada directamente al servidor central de Battle.net).

El servidor dedicado al almacenaje de partidas funciona de la siguiente manera: por cada partida creada, se le asocia un modo de juego de entre todos los disponibles para setear las normas, y se reserva un espacio en el servidor para dicha partida. Los espacios de partida almacenados persisten en el servidor hasta la siguiente actualización, que ocurre normalmente en un periodo de tiempo comprendido entre las dos semanas y un mes. Así, se impide que los servidores se queden sin espacio para almacenar más “salas” de partida efectivas. Los jugadores, a su vez, tienen tiempo para guardar en sus dispositivos las repeticiones de sus partidas preferidas, descargándose directamente un archivo de forma local. Lo que persiste en Overwatch 2 son, por tanto, los resultados de las partidas y los cambios de estadísticas y datos sucedidos tras ellas, almacenados en variables concretas.

Con ello, el videojuego no distingue entre servidores dedicados a distintos modos de juego, sino que utiliza un **servidor gigante para almacenar todas las lobbies de partida**, y a cada partida se le asocia su modo de juego concreto para corresponderse con la selección del jugador y ajustarse a las normas concretas de cada modo. En el caso del sistema de emparejamiento que encuentra los jugadores para estas partidas, este se aloja en los servidores centrales de Blizzard, no dentro del videojuego.

En cuanto al borrado de las “salas” de partida dentro del servidor, se trata de una medida para ahorrarse problemas de retrocompatibilidad tras cada parche, permitiendo al equipo de Overwatch 2 el lanzamiento de parches y actualizaciones continuas que mantengan el videojuego vivo o corrijan errores de forma rápida y eficaz. Y de paso, recuperan espacio en los servidores, haciendo que no lleguen nunca a llenarse o suponer una cuestión de ralentización en las partidas.

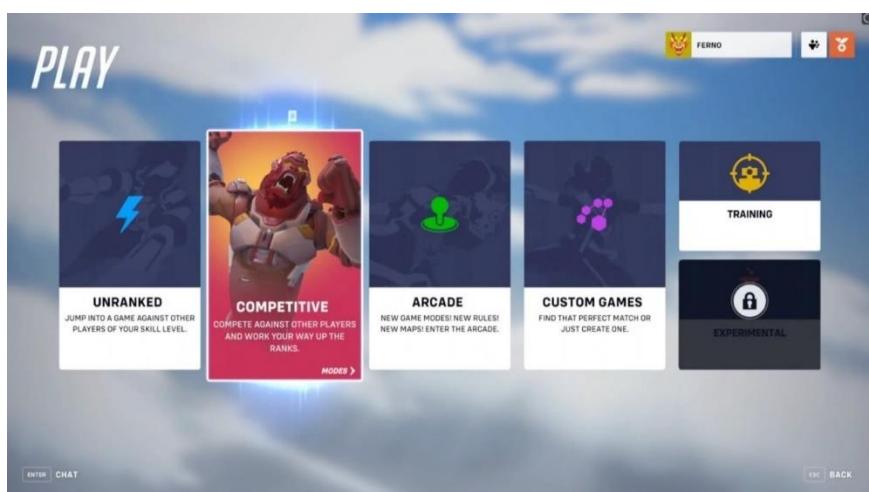


Imagen 8, pantalla de selección de modo de juego y tipo de cola dentro de Overwatch 2.

Algoritmos de emparejamiento y matchmaking para el juego competitivo

Dentro de todos los modos de juego de Overwatch 2, el que más destaca, sin duda alguna, es el modo competitivo o “ranked”. En él, los jugadores juegan partidas más largas, serias y disputadas en busca de escalar a través de las ocho divisiones del videojuego, siendo el objetivo final el estar entre los mejores jugadores y aparecer en la clasificación de los mejores 500 en cada temporada. Para regular este sistema competitivo y que no se sienta frustrante para el jugador, el emparejamiento de Overwatch 2 utiliza un **sistema oculto de Calificación de Emparejamiento (MMR)**, similar a la clasificación ELO (característica de deportes como el ajedrez) para estimar la habilidad de los jugadores. Los jugadores no pueden visualizar nunca este valor, es **interno del sistema**. En su lugar, visualizan en qué división se encuentran y su progreso restante hasta el siguiente rango en forma de una barra de progreso y un porcentaje aproximado.

Con este índice a mano, y teniendo en cuenta el rango visible del jugador (uno comprendido entre las ocho divisiones, por ejemplo, oro), el servidor busca y crea una sala de partida lo más cercana posible al 50% de probabilidades de victoria **agrupando jugadores con MMR similares** que anduvieran buscando partida, equilibrando roles preferidos o demandados con el **sistema de cola Role Delta** y minimizando la latencia gracias a la proximidad del servidor para todos los participantes en la partida. Todo esto es gestionado por los **servidores centralizados** encargados de la creación de partidas, que procesan estos datos para crear partidas justas y equilibradas rápidamente.



Imagen 9, pantalla de subida de rangos tras una partida en el modo de juego competitivo.

Así, los elementos que intervienen dentro del **algoritmo de matchmaking** de Overwatch 2 son:

- **MMR (Calificación oculta de Emparejamiento):** Cada jugador tiene un MMR numérico oculto, no vinculado directamente a su rango visible (como Plata u Oro). Es una estimación de habilidad en una curva de campana, donde 0 es el promedio; el sistema intenta encontrar jugadores con MMR similares para formar partidas equilibradas. El MMR se ajusta tras las victorias y derrotas según la diferencia parcial entre el MMR total de ambos equipos, con cambios mayores inicialmente para que los nuevos jugadores se adapten más rápido y adquieran su rango lo más rápido posible.
- **Rol Delta:** Sistema en forma de cola que clasifica y organiza a los jugadores en función de su rol preferido o seleccionado para jugar la partida, ya que el videojuego solicita un equilibrio de un tanque, dos daños y dos apoyos por equipo. También le da prioridad en la cola a los tanques, ya que es un rol más complicado de encontrar al solicitar únicamente un jugador.
- **Tiempo de espera y latencia (ping):** El algoritmo de emparejamiento de prioridad para entrar en partida a aquellos jugadores que más tiempo llevan buscando partida. A su vez, también selecciona entre aquellos con menores condiciones de latencia dentro del servidor regional para que la partida no se sienta ralentizada o frustrante.
- **Tamaños de grupo:** En caso de jugar con compañero de equipo predefinidos en el mismo equipo, el sistema busca equilibrar los tamaños de los grupos dentro de ambos equipos (si un equipo cuenta con un trío predefinido de tres jugadores, el sistema va a buscar otro trío de jugadores en cola para equilibrar el equipo rival). En caso de no ser posible, se aplicarán disparidades de rangos competitivos para compensar.
- **Objetivo de Equidad:** La filosofía principal del sistema es crear partidas donde ambos equipos tengan aproximadamente un 45-55% de posibilidades de ganar de forma estadística, haciendo que la compenetración del equipo y, a su vez, la habilidad y la estrategia individuales sean el factor decisivo en cada partida, al tratarse de un modo competitivo.

Aún así, se sospecha dentro de la comunidad de jugadores que existe un algoritmo oculto, aparte del que gestiona el objetivo de equidad, ya que se observan muchas rachas disparatadas de victorias y derrotas con frecuencia dentro del modo competitivo. Podría considerarse una estrategia psicológica para enganchar al jugador, ya que si este consigue una racha continua de victorias se sentirá poderoso y verá mucho progreso repentino, pero si tiempo después este jugador sufre una racha muy continua de derrotas se verá obligado a continuar jugando partidas competitivas para recuperar el rango perdido.

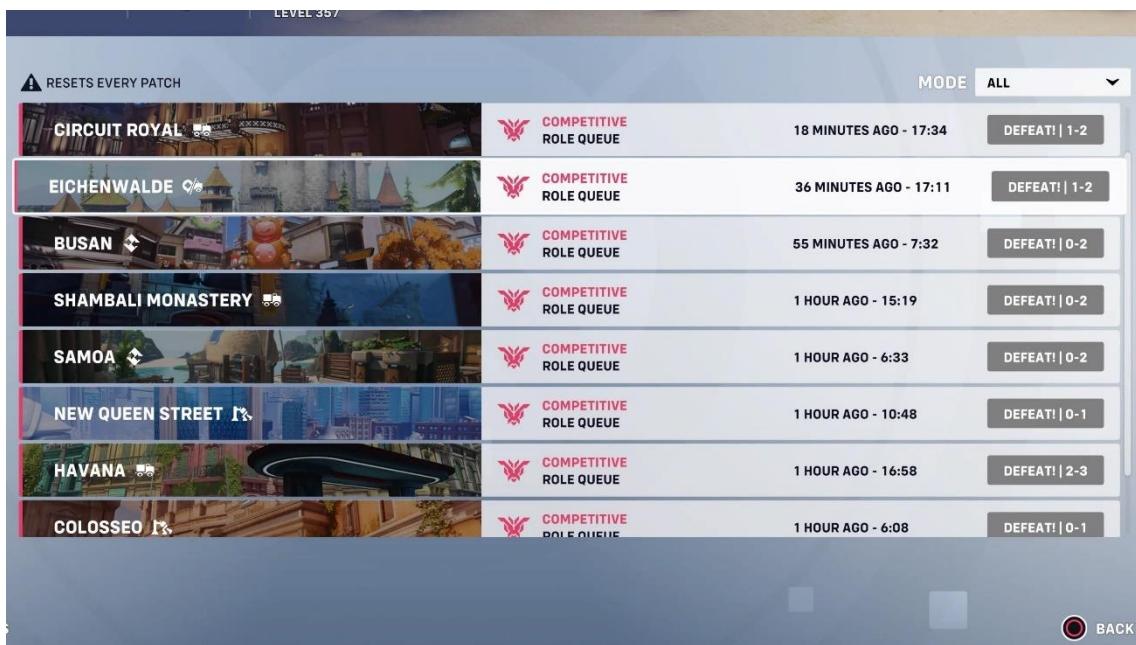


Imagen 10, ejemplo de historial de racha de derrotas sospechoso dentro del modo competitivo.

En el caso de la distribución de red, se gestiona igual que como se ha explicado en el apartado anterior. La compleja lógica de emparejamiento no se distribuye a las máquinas cliente (jugadores), sino que se ejecuta en los servidores *backend* centralizados de Blizzard. Al entrar en cola, tu cliente envía la solicitud a estos servidores, que buscan en sus grupos de jugadores aplicando los criterios anteriores del algoritmo de emparejamiento. Una vez formada la partida, el servidor de juego (que suele estar más cerca de los jugadores para un ping bajo) aloja y almacena la partida, pero **las decisiones de emparejamiento se toman del lado del servidor central**.

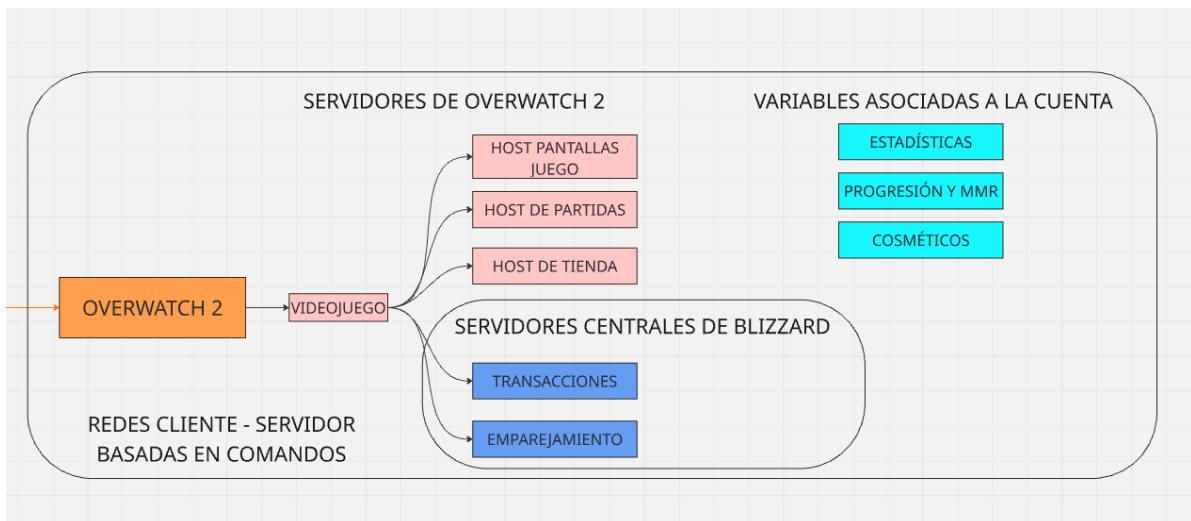


Imagen 11, esquema representativo de la arquitectura de red de las distintas funciones de Overwatch 2.

CARACTERÍSTICAS TÉCNICAS PRINCIPALES

Motor y lenguaje

Overwatch 2 está desarrollado utilizando un **motor propietario personalizado basado en el motor original de Overwatch**, que era una versión mejorada de un prototipo previo de la empresa que data del año 2014. Blizzard ha seguido desarrollando y actualizando este motor internamente, en lugar de cambiar a uno nuevo como Unreal Engine, todo con el objetivo de mantener el control completo y adaptarlo específicamente a sus necesidades, como aquellas funciones agregadas con el paso a Overwatch 2. El nombre de este motor personalizado **no ha sido nunca desvelado**, más allá de ser referido como “the Overwatch 2 engine” o rumoreado como “Prometeus”.

Al no conocer detalles en profundidad acerca del motor de desarrollo, ni siquiera su nombre, es complicado conocer con exactitud el lenguaje mediante el cual ha sido programado Overwatch 2. Sin embargo, observando las funciones de las que hace uso el videojuego, se puede deducir que se ha realizado en su mayoría utilizando **lenguajes de programación orientados a objetos**, como C++ para cuestiones gráficas, de lógica general de juego y para las habilidades (al igual que Unreal Engine). **Se descarta el uso de C#**, ya que el ensamblado del videojuego exportado no se trata de un ensamblado .net, sino de un **ensamblado personalizado**, más sencillo de lograr mediante C++.

Se conoce, además, que dentro de este motor de desarrollo personalizado, Blizzard incorporó **Statescript**, un **sistema de programación visual propio realizado en C++**, el cual ayuda al equipo en tareas como la lógica interna de los héroes del videojuego y sus distintos estados y habilidades, así como de los estados de las partidas y modos. Se trata de **programación visual mediante nodos**, a la que se añade también una contraparte de programación basada en texto, por lo que resulta ser muy **similar al sistema de Blueprints** que presenta Unreal Engine.

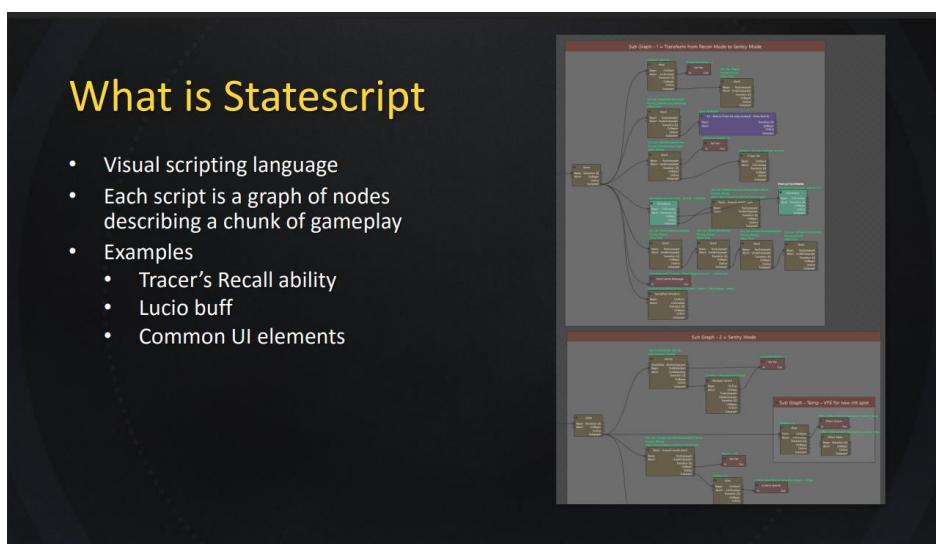


Imagen 12, captura de la presentación de Blizzard sobre el desarrollo de Statescript. Extraído de GDC 2017, expuesta por Dan Reed.

Desarrollo

El comienzo del desarrollo de Overwatch supuso una nueva época dentro de Blizzard. Si bien ya disponían de otros grandes y exitosos videojuegos multijugador, como es el caso de World of Warcraft, lo cierto es que tenían que enfrentarse a un novedoso género de videojuego, así como a su planificación a largo plazo y sus altos requerimientos técnicos.

Primeramente, el equipo de desarrollo descartó el uso de cualquier motor anterior previamente desarrollado. No les servía el motor propio que ya existía para el World of Warcraft, al tratarse de dos géneros de videojuego tan dispares e incompatibles, además de la estimación de que dicho *software* no aguantaría las demandas técnicas de Overwatch. También descartaron utilizar cualquier otro tipo de motor no propietario, como Unity o Unreal Engine, ya que, aunque soportaran las necesidades técnicas, con ellos perdían la personalización y el control de las actualizaciones, produciendo que los nuevos parches salieran de forma más limitada y con contenido menos original. Aunque fue una tarea ardua, decidieron utilizar un motor de desarrollo propio, que les permitiera mantener el completo control técnico y creativo.

El primer prototipo del motor de Overwatch se desarrolló en 2014. A partir de ahí, el equipo, por aquel entonces conformado por unas **cien personas**, continuó iterando sobre él, hasta alcanzar la **salida de Overwatch al mercado en 2016**. Pasado un tiempo, se reaprovecharía y se seguiría actualizando este motor constantemente hasta la llegada de Overwatch 2, incluyendo nuevas funcionalidades y haciendo gala del por qué de la preferencia de un motor personalizado.



Imagen 13, captura de prototipo jugable de Overwatch dentro del motor propietario de Blizzard. Extraída de la presentación de GDC2017 expuesta por Dan Reed.

Al tratarse de un *software* completamente nuevo y hecho a medida, los primeros pasos de este motor fueron duros y desmotivantes: resultaba ser una gran sobrecarga repentina de trabajo y a contrarreloj para un resultado que solo podrían ver unas ochenta personas a lo sumo. Además, **no existía documentación sólida** ni se encontraban implementadas todas las herramientas necesarias, por lo que el equipo creativo tenía que dar bandazos con cada creación de una nueva característica para el juego. Conforme se avanzó en el desarrollo del motor, el trabajo en equipo se volvió más gratificante, considerando ya avances reales en el desarrollo.

A día de hoy, **se sigue utilizando el mismo motor que por aquel entonces**, pero desarrollado al máximo, con multitud de herramientas y facilidad para implementar estados o mecánicas nuevas dentro de Overwatch 2. El motor también simplifica y agiliza enormemente las actualizaciones periódicas, que suceden siempre cada dos semanas.

Más allá de la programación de Overwatch y su motor de desarrollo, la planificación se llevó a cabo, al mismo tiempo, en otros ámbitos del título. Los responsables artísticos de los héroes y los mapas acordaron un sistema de trabajo **muy organizado e iterativo**, con pruebas prácticamente en todas las fases de concepción de los personajes, desde el modelado hasta la animación y las referencias. A pesar de encontrarse limitados en un principio, desarrollaron un trabajo riguroso y con numerosas capas de comprobación, hasta alcanzar a los carismáticos y únicos personajes que se pueden disfrutar ya en el videojuego.



Imagen 14, captura de la presentación sobre diseño y animación del héroe Mei. Expuesta por David Anthony Gibson en GDC2016.

Arquitectura de red y protocolos

La arquitectura global de Overwatch 2, aunque se encuentre planificada y bien gestionada a gran escala, puede resultar muy compleja y confusa. El videojuego utiliza **dos centros de servidores** de forma general, **los servidores de Blizzard** y los **servidores dedicados al propio Overwatch**. Todas aquellas acciones relacionadas con el lanzador del videojuego, actualizaciones o transacciones pasan por el servidor de la empresa, mientras que las acciones internas del videojuego pasan por el servidor dedicado a él. Existen dos excepciones a esta distribución general:

- **El sistema de emparejamiento o *matchmaking***, a pesar de encontrarse dentro del videojuego, se produce externamente en los servidores de Blizzard, mientras que la propia partida, una vez formada, se aloja dentro de los servidores de Overwatch.
- Algunos **datos de persistencia**, en especial los **relacionados con compras** y transacciones, se almacenan en la cuenta de Battle.net, y no en el perfil dentro de Overwatch.

Aun así, se deduce una **conexión entre ambos centros de servidores**, aunque se produzcan varios saltos entre ellos en casos concretos. Esto podría resultar ser muy poco óptimo, pero las tareas en las que se producen dichos saltos son concretas y especializadas y en ningún momento ocupan la mayor parte del volumen de entradas del videojuego.

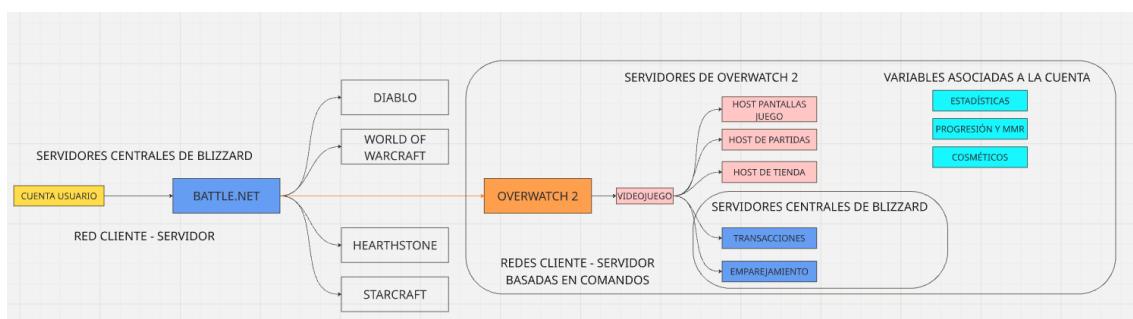


Imagen 15, esquema global de la arquitectura de red de Overwatch 2.

Entrando en los **protocolos de intercambio de paquetes**, Overwatch utiliza principalmente **UDP**, debido a que se trata de un juego en tiempo real. Su alta velocidad y su baja sobrecarga ayudan a minimizar al máximo la latencia, a pesar de su complejidad. Sin embargo, también utiliza **TCP** para datos concretos y que requieren cierta seguridad, como el chat dentro del videojuego, las credenciales del inicio de sesión y las direcciones de las actualizaciones, utilizando puertos específicos, especialmente dentro de Battle.net.

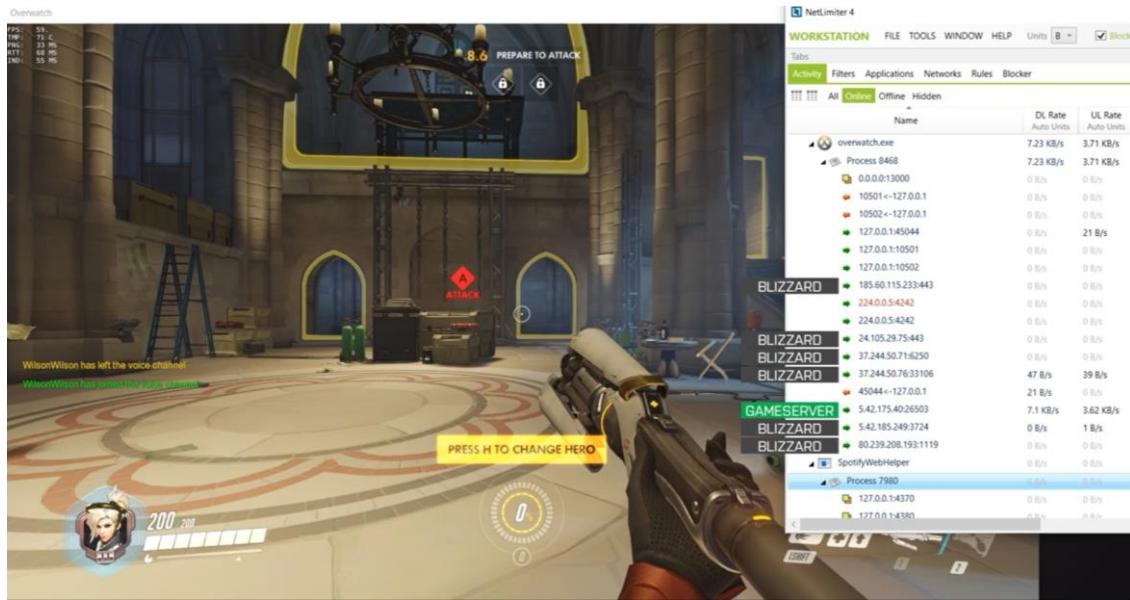


Imagen 16, captura de pantalla de los subprocessos relativos a los servidores de Blizzard que suceden mientras se juega Overwatch 2.

Entrando a la arquitectura del videojuego de forma más concreta, Overwatch se ha programado utilizando el **sistema ECS (Entity Component System**, o sistema de componentes y entidades en castellano).

Este sistema separa los datos en tres grandes grupos: **entidades** (ítems individuales vacíos), **componentes** (características o estados que se le agregan a las entidades) y **sistemas** (correspondientes a comportamientos lógicos o mecánicas). ECS facilita la estructuración de la lógica del juego, ya que así toda recae en los sistemas, que se encargan de procesar las entidades que participan en ellos. También mejora la flexibilidad, porque gracias a esta estructuración se pueden agregar muchos estados o comportamientos nuevos sin pasar por un extenso árbol de código y herencias que afecte al videojuego en todos los aspectos.

Para poder hacer realidad el ECS, todos los sistemas y comportamientos centrales del videojuego se han realizado mediante el uso del **patrón singleton**, que permite el total acceso de todos los elementos del videojuego a estos sistemas.

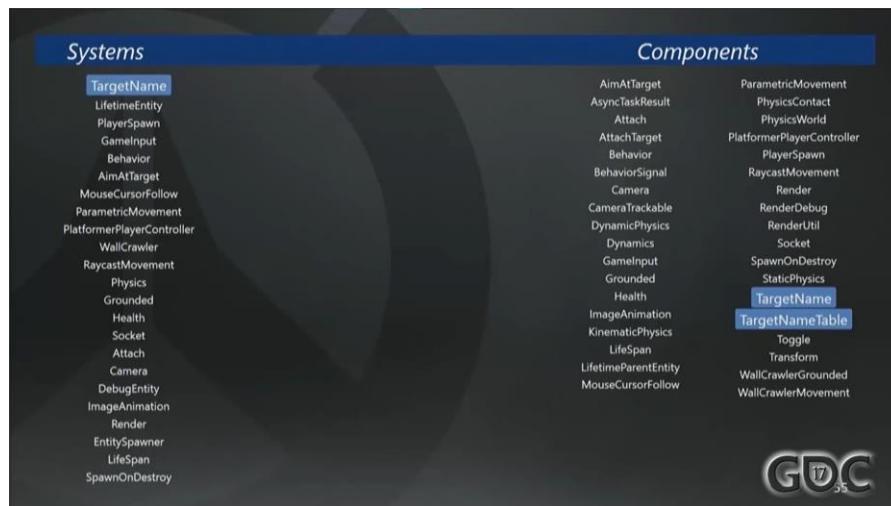


Imagen 17, captura de pantalla sobre la presentación del funcionamiento del sistema ECS. Expuesta por Timothy Ford en GDC2017.

Para la jerarquización de la jugabilidad, Overwatch usa una **red cliente – servidor autoritativo y basada en comandos**, pero con una conexión un poco más profunda y planificada. El servidor central en el que se aloja la partida es el que contiene los resultados concretos a tiempo real de la partida, así como la cola de ejecución de las acciones ejecutadas por los jugadores. Es el ente que “manda” y define el estado del juego.

En un videojuego frenético a tiempo real como lo es Overwatch, en la mayoría de las ocasiones los estados de la partida o las acciones varían en apenas milisegundos, por lo que al servidor central no le daría para enviar la información “real” de vuelta a las pantallas de los jugadores a tiempo. Los jugadores, entonces, cuentan con un **punto de vista “atrasado”** de lo que sucede en la partida.

Para solventar la diferencia temporal, el servidor central utiliza un **sistema de predicción** de las posiciones o acciones de los jugadores, salvando así ese tiempo de **buffer** mientras se envía el estado del juego actualizado. Salvando los errores de transmisión de paquetes, latencia o predicciones erróneas, el sistema multijugador cuenta con una especie de **rollback**, característico de los videojuegos de lucha modernos (los cuales comparten la naturaleza de juego frenética) para corregir estos fallos de forma rápida e imperceptible.



Imagen 18, captura de pantalla representativa del funcionamiento de recibido y envío de información del servidor de Overwatch en tiempo real. Expuesta por Timothy Ford en GDC2017.

En cuanto a la **latencia**, puede ser el **peor enemigo del sistema multijugador**, haciendo que la información resulte inconexa en las pantallas de los jugadores. Con el objetivo de evitar altos valores de latencia en la mayoría de lo posible y guardar la coherencia con la jugabilidad, Overwatch 2 actúa de forma **determinista a través de su servidor central** para resolver las interacciones en las partidas a tiempo real. El servidor registra los comandos de los jugadores, permitiendo que no exista margen de error en la ejecución de las acciones.

Los videojuegos multijugador cuentan con una tasa de actualizaciones por segundo que son capaces de registrar y enviar a los jugadores. En Overwatch 2, esta **tasa de tickrate es de 64 hercios**, una cantidad estándar la cual consigue que, sin llegar a sufrir niveles extra de latencia, el videojuego resulte coherente y justo con sus acciones, sobre todo con aquellas relacionadas con la puntería, crucial en un videojuego *shooter*.

Medidas de seguridad

En el modelo multijugador de Overwatch 2, la arquitectura centrada en un servidor autoritativo cumple un rol esencial en la seguridad: al ser el servidor quien valida y determina el estado real de la partida, la información crítica nunca reside en el cliente, lo que reduce de forma significativa la capacidad de manipulación externa y limita posibles intrusiones o *cheats* basados en alteración de archivos locales del juego. Esta estructura mantiene a los jugadores en un entorno controlado, donde las decisiones importantes (como movimiento, daño, colisiones, habilidades o estado de la partida) no dependen del dispositivo del usuario, sino de una instancia gestionada directamente por Blizzard.

Asimismo, el uso de software propietario en el motor y en las herramientas internas de Overwatch 2 dificulta la explotación de vulnerabilidades comunes, ya que las técnicas habituales usadas contra motores abiertos o documentados en Internet no resultan aplicables de forma directa. Sin embargo, aunque Blizzard es reconocida por la solidez de su infraestructura global, la seguridad específica de sus servidores es en gran medida desconocida y sin información o documentación al respecto, lo cual se trata de una

excelente señal en materia de seguridad y habla bien de la fiabilidad de la estructura del videojuego.

Sistema antitrampas y de reportes

El sistema antitrampas de Overwatch 2 **no cuenta con un componente dedicado o externo** similar a Blizzard Warden, el sistema antitrampas utilizado en World of Warcraft. En lugar de ello, la seguridad se basa en el **registro exhaustivo de todas las acciones realizadas en el servidor** durante las partidas. Este registro permite identificar comportamientos que podrían considerarse sospechosos mediante algoritmos internos no divulgados públicamente, funcionando como un mecanismo de detección y prevención de trampas basado en análisis de patrones y comportamientos anómalos.

Además, la arquitectura del juego dificulta la introducción de software de trampas a nivel local, ya que Overwatch 2 se ejecuta a través de otras plataformas, esencialmente Battle.net o Steam. Esta integración limita la manipulación directa del cliente de juego, aumentando la dificultad para que los trampos puedan introducir o modificar código de forma maliciosa sin ser detectados. Complementariamente, el sistema incluye un mecanismo de reportes dentro del juego que permite a los jugadores denunciar comportamientos inapropiados o sospechosos, junto con un sistema de honor que incentiva el juego limpio y la conducta positiva para los jugadores.

Para reforzar la experiencia y la seguridad, Overwatch 2 implementa **listas de evitación** (*avoidlist*) y **listas de bloqueo** (*blocklist*). La *avoidlist* permite a los jugadores evitar emparejamientos con usuarios previamente reportados o considerados problemáticos, mientras que la *blocklist* actúa como un **filtro más estricto** para impedir la interacción con jugadores sancionados o con historial negativo. Estos sistemas colaboran para mantener un entorno competitivo más justo y agradable, complementando las medidas técnicas y de supervisión implementadas en el *backend* del juego.

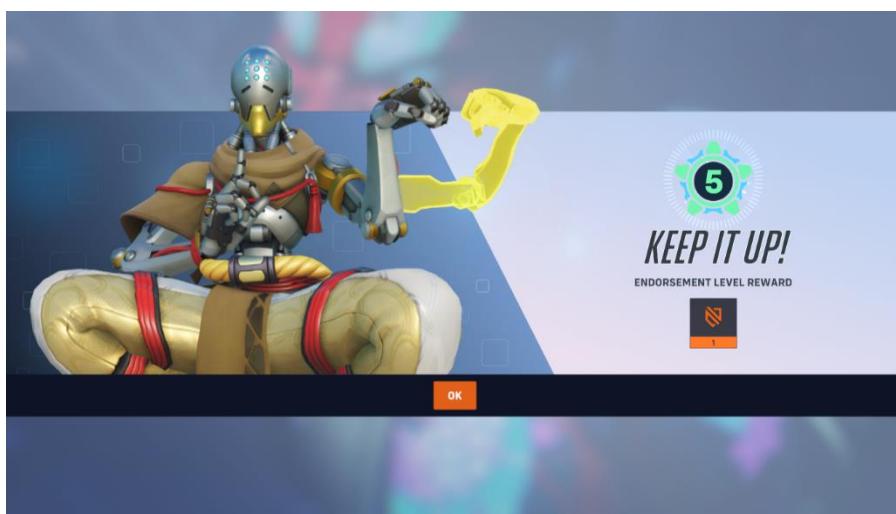


Imagen 19, captura de pantalla dentro de Overwatch 2, recompensando el nivel alto de honor de un jugador.

DESAFÍOS TÉCNICOS

Número de servidores y multijugador masivo

El mantenimiento de la infraestructura de servidores para Overwatch 2 representa un desafío técnico significativo debido a la necesidad de operar aproximadamente **quince servidores de gran tamaño**, distribuidos estratégicamente en ciudades clave de distintas regiones globales y continentes. Esta distribución geográfica es fundamental para minimizar la latencia y garantizar una experiencia de juego fluida y competitiva para los usuarios en diferentes ubicaciones. La complejidad aumenta al integrar múltiples plataformas y habilitar el *crossplay* entre distintas consolas y plataformas de juego, lo que requiere una sincronización precisa y una adaptación de los recursos para que las partidas funcionen, a pesar de la diversidad de hardware y diferentes protocolos de conexiones de red.

La operación de estos servidores está respaldada por una infraestructura centralizada en los centros de datos de Blizzard, que actúan como nodos principales autoritativos para la coordinación y el control del tráfico de datos. Este modelo solo es viable gracias a una organización rigurosa, un presupuesto considerable y un equipo de profesional es muy amplio, características típicas de un estudio de desarrollo triple A. La magnitud del proyecto demanda no solo la implementación inicial, sino también una planificación detallada para la escalabilidad, la redundancia y la seguridad, asegurando que el sistema pueda manejar picos de usuarios y posibles fallos sin interrupciones significativas.

Solo una empresa con una capacidad financiera y técnica solventes como Blizzard puede garantizar la estabilidad, seguridad y calidad del servicio, permitiendo que millones de jugadores disfruten de una experiencia competitiva y estable en Overwatch 2.

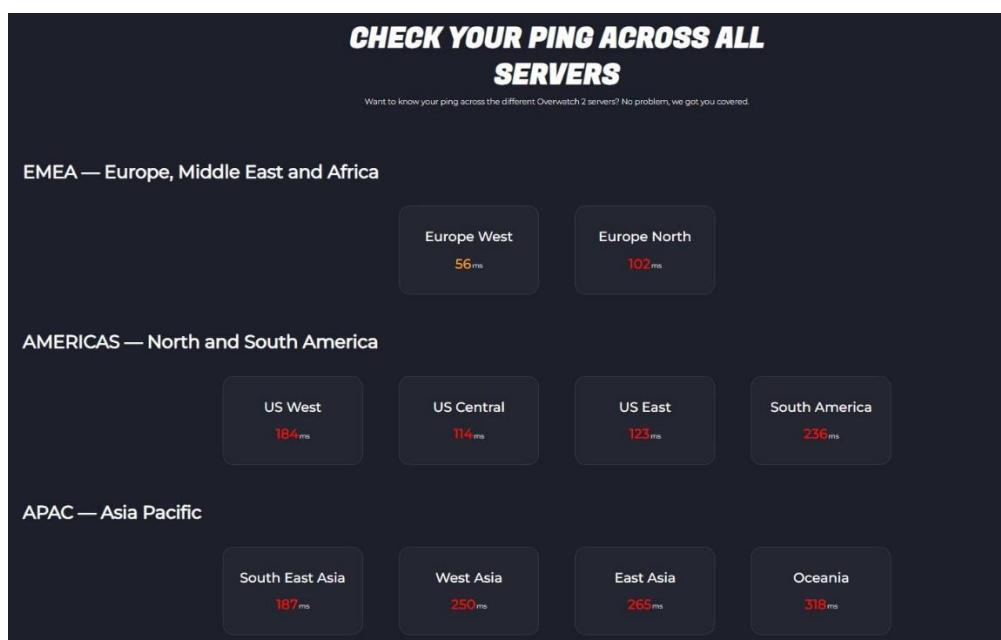


Imagen 20, visualización general de los servidores de juego disponibles por regiones en Overwatch 2.

Frenetismo del videojuego y gestión de la latencia

La gestión del frenetismo característico de Overwatch 2 en un entorno *online* representa un desafío técnico considerable debido a la necesidad de sincronizar múltiples jugadores en tiempo real con alta precisión. Para abordar esta complejidad, el sistema utiliza un enfoque basado en *rollback*, similar al empleado en juegos de lucha, pero adaptado para manejar un número mucho mayor de participantes simultáneos. En este esquema, **el servidor actúa como la autoridad principal**, registrando y prediciendo las acciones de los jugadores para corregir discrepancias y mantener la coherencia del estado del juego. Esta técnica requiere una optimización exhaustiva, más si se implementa para diez jugadores al mismo tiempo en un mapa tridimensional, para funcionar eficazmente. Cualquier retraso o error en la predicción puede afectar a la partida o a la visualización de las acciones en las pantallas de los jugadores.

En el caso de la latencia, podría ser uno de los principales enemigos y amenazas en la arquitectura de Overwatch 2, ya que puede comprometer la precisión de la predicción del servidor y distorsionar la percepción del jugador sobre el desarrollo de la partida. Una latencia elevada puede ser fatal para un título competitivo y rápido como Overwatch 2, haciendo que las posiciones, acciones, habilidades y punterías se vieran fuertemente distorsionadas y fueran consideradas como injustas. Para mitigar este riesgo, Blizzard implementa una red de servidores regionales estratégicamente ubicados, que minimizan el tiempo de respuesta y aseguran que la latencia se mantenga dentro de límites aceptables, garantizando así una experiencia fluida y justa para todos los jugadores dentro de un mismo servidor.

Cargado a tiempo real de mapas, modelos, texturas y efectos especiales

El cargado en tiempo real de mapas, modelos, texturas y efectos especiales en Overwatch 2 representa un desafío técnico considerable debido a la complejidad y la exigencia de mantener un rendimiento óptimo en un entorno competitivo. Para conseguirlo, se han cuidado muchos aspectos gráficos distintos del videojuego, con el objetivo de que cada *frame* renderizado no ralentice el juego, mientras que a la vez presente una estética gráfica atractiva, definida y reconocible. La buena calidad gráfica y optimización de esta ayudan, en cierto modo, a que la jugabilidad y acciones basadas en puntería o visión espacial a tiempo real no resulten complicadas o frustrantes.

Los mapas se precargan en los servidores según el modo de juego seleccionado, y su ejecución **comienza durante la fase de selección de héroes**. Durante esta etapa, el jugador visualiza un fondo prerenderizado del mapa, lo que permite que la carga completa del entorno se realice de manera transparente y sin interrupciones perceptibles. Este método asegura que, al iniciar la partida, el mapa ya esté completamente cargado y listo, sin tiempos de espera adicionales.



Imagen 21, pantalla de selección de héroes, previa a una partida de Overwatch 2.

A nivel técnico e interno, el motor gráfico de Overwatch 2 es similar en arquitectura a Unreal Engine, lo que facilita el uso de técnicas avanzadas de renderizado y optimización. En particular, el juego utiliza **iluminación preconfigurada** para los entornos estáticos, apoyándose en mapas de luz precalculados y sondas para objetos dinámicos. Esta técnica permite obtener imágenes visualmente impactantes con un alto rendimiento, **evitando el uso de trazado de rayos en tiempo real**, que sería inviable dada la velocidad y naturaleza competitiva del juego. **La iluminación indirecta y los rebotes de luz se calculan previamente y se cargan junto con el mapa**, garantizando un rendimiento fluido durante la partida.

El proceso de renderizado incluye un sistema de **shadow depth** (profundidad de sombreado) que parte de luces direccionales y omnidireccionales (como la luz solar o de ambiente), seguido por un **prepass** en texturas gigantes que integran múltiples capas como **base color, albedo, metallness, roughness, emissive, view depth y shadow patterns**. Estas capas se combinan mediante **mipmaps** para generar **occlusion ambiental (AO)** y reflejos, integrándose finalmente con la interfaz de usuario para producir la imagen final **en un solo frame**. Esta *pipeline* optimizada es fundamental para mantener la calidad visual sin comprometer la velocidad de actualización necesaria en un juego *online*.

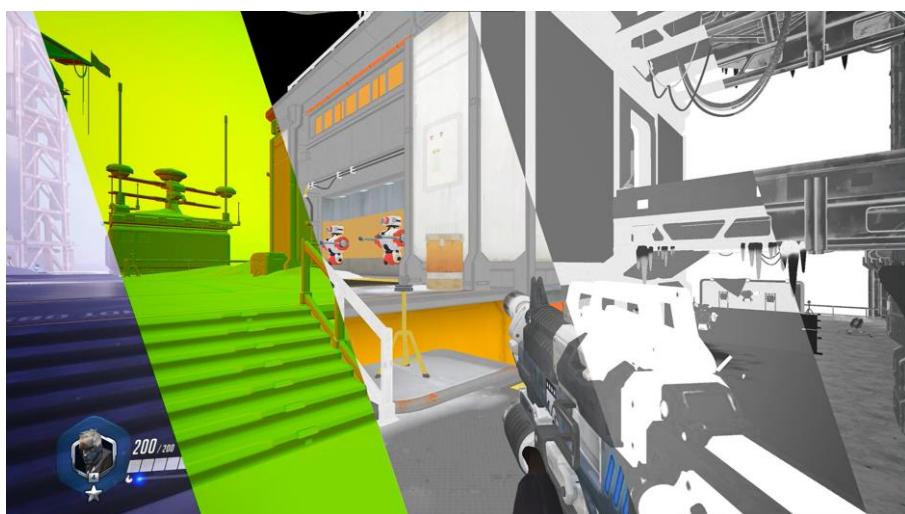


Imagen 22, recreación de las distintas capas gráficas cargadas simultáneamente en un solo frame del juego.

Para gestionar los efectos especiales, estos **están integrados dentro de la programación de las habilidades de los héroes**, lo que permite que tanto la lógica como los gráficos se carguen simultáneamente. Esta integración reduce la carga de trabajo y mejora la eficiencia, ya que evita procesos separados para la gestión visual y funcional de las habilidades, contribuyendo a una experiencia de juego más coherente y responsiva.



Imagen 23, ejecución en tiempo real de los efectos especiales relativos a una habilidad dentro de una partida.

Optimización multiplataforma

Overwatch 2 ha abordado con éxito varios desafíos técnicos relacionados con la optimización del juego en un entorno multiplataforma, donde la complejidad de las partidas y las múltiples interacciones entre jugadores pueden variar significativamente según la plataforma utilizada. El desarrollo se ha centrado inicialmente en la versión para PC, considerada el núcleo del sistema, desde donde se trasladan y adaptan las mecánicas y funcionalidades para su posterior implementación y exhaustiva prueba en consolas.

En cuanto a la infraestructura de servidores, Overwatch 2 ha implementado una arquitectura diferenciada que incluye servidores exclusivos para consolas y para partidas con *crossplay*, lo que facilita la gestión eficiente de las conexiones y la latencia, optimizando la experiencia de juego para todos los usuarios, independientemente del dispositivo. Además, se integra el soporte de Battle.net como plataforma centralizada para la gestión de tiendas digitales y servicios en línea, incluso en las consolas, lo que unifica y simplifica la distribución y actualización del juego, garantizando compatibilidad y sincronización.

COMPARACIÓN CON TÍTULOS SIMILARES

Marvel Rivals: uso de un motor gráfico menos personalizado

Marvel Rivals utiliza **Unreal Engine** como base para su desarrollo, lo que le permite alcanzar un nivel gráfico comparable al de Overwatch 2 en términos de calidad visual. Sin embargo, a pesar de esta paridad estética, Marvel Rivals presenta una **demandas de recursos significativamente mayor** en comparación con Overwatch 2. Esta mayor exigencia se traduce en un rendimiento menos eficiente, especialmente en sistemas con **hardware** de gama media o baja, donde la optimización técnica del juego resulta insuficiente para garantizar una experiencia fluida.

En contraste, Overwatch 2 ha sido desarrollado con un enfoque más riguroso en la optimización, permitiendo que el juego funcione de manera más eficiente en una amplia variedad de configuraciones de sistemas. Esta optimización se refleja en una mejor gestión de los recursos del sistema, lo que facilita una experiencia de juego más accesible para usuarios con ordenadores menos potentes. Así, aunque ambos títulos ofrecen gráficos de alta calidad, Overwatch 2 destaca por su capacidad para mantener un rendimiento sólido y estable en un espectro más amplio de configuraciones técnicas, lo que mejora la experiencia general del jugador.



Imagen 24, captura relativa a una partida de Marvel Rivals.

Paladins: uso de un sistema antitrampas externo

Paladins implementa **EasyAntiCheat (EAC)** como una solución externa para la detección y prevención de trampas, lo que garantiza un nivel elevado de seguridad y actualización constante frente a nuevas amenazas de *hackeo* o *softwares* externos. Sin embargo, esta dependencia de un programa antitrampas externo conlleva ciertos inconvenientes técnicos, entre ellos un coste adicional en términos de integración y mantenimiento, así como un **impacto en la velocidad de procesamiento del dispositivo** del jugador, al tener que comprobar el estado del videojuego y del jugador *frame por frame*. En particular, EasyAntiCheat puede ser **más invasivo** en cuanto al uso de recursos del sistema, ya que **opera en tiempo real durante las partidas**, consumiendo capacidad de procesamiento que podría afectar el rendimiento general del juego, especialmente en equipos con *hardware* limitado. También supone un **coste extra de contratación y la pérdida de autoridad y control** en dicho aspecto del videojuego, que queda “subcontratado”.

En comparación, Overwatch 2 ha desarrollado un sistema antitrampas más integrado y optimizado dentro de su propia arquitectura, lo que reduce la necesidad de recursos adicionales y minimiza la interferencia con el rendimiento del juego.

Así, mientras Paladins apuesta por una solución externa que, aunque efectiva, puede impactar negativamente en el rendimiento, Overwatch 2 prioriza una implementación interna que optimiza el uso de recursos y mejora la estabilidad durante el juego.



Imagen 25, captura relativa a una partida de Paladins.

Valorant: mayor tasa de ticks por segundo

La diferencia en el *tickrate* entre Overwatch 2 (64 ticks por segundo) y Valorant (128 ticks por segundo) tiene implicaciones técnicas significativas en la precisión y el realismo de las interacciones dentro del juego.

Un *tickrate* más alto, como el de Valorant, permite una **actualización más frecuente del estado del juego**, lo que se traduce en una **mayor precisión en la detección de impactos y movimientos**, crucial para un título que se basa intensamente en la puntería y la ejecución rápida de habilidades en **mapas relativamente pequeños**. Además, las partidas de Valorant suelen ser **más cortas y estructuradas en rondas**, lo que facilita la asignación de recursos para mantener un *tickrate* elevado sin comprometer la estabilidad o el rendimiento del servidor.

Por otro lado, Overwatch 2 presenta un diseño de mapas más grandes y partidas con rondas más largas, lo que implica una mayor complejidad en el cálculo y sincronización de múltiples habilidades y efectos simultáneos. Incrementar el *tickrate* en este contexto supondría un aumento exponencial en la carga de procesamiento y en el volumen de datos transmitidos, lo que podría derivar en pérdidas de paquetes y problemas de sincronización, generando saltos en el tiempo y el espacio dentro del juego.

Mantener un *tickrate* de 64 en Overwatch 2 representa un compromiso técnico que equilibra la necesidad de precisión con la capacidad de gestionar un entorno de juego más complejo y dinámico que los servidores sean capaces de soportar en un lapso de tiempo casi inmediato.



Imagen 26, captura relativa a una partida de Valorant.

CONCLUSIÓN

Overwatch 2 se ha consolidado como **uno de los videojuegos más emblemáticos en el ámbito del multijugador online masivo** y de los juegos como servicio, destacando especialmente por su excelencia técnica. Este reconocimiento no es casualidad, sino el resultado de una meticulosa planificación y una vasta experiencia acumulada por Blizzard, una empresa con recursos significativos que ha sabido **equilibrar la alta demanda técnica del juego con una optimización rigurosa y una estructura ordenada**. La arquitectura de servidores diseñada para Overwatch 2 permite gestionar eficazmente la complejidad inherente a partidas con múltiples interacciones en tiempo real, garantizando estabilidad, una experiencia de juego funcional en diversas plataformas y consolas al mismo tiempo y una victoria momentánea frente al gran enemigo de los videojuegos multijugador a tiempo real: la latencia.

La optimización técnica también se extiende al aspecto gráfico y jugable, donde se ha logrado un balance entre calidad visual y rendimiento, asegurando que el juego pueda mantenerse vigente y moderno hasta hoy en día, casi diez años después. La estructura de servidores, combinada con técnicas avanzadas de optimización, no solo soporta la carga de trabajo actual, sino que también está preparada para adaptarse a futuras actualizaciones y expansiones, las cuales suceden de forma regular.

En conjunto, Overwatch 2 ofrece un **modelo ejemplar** de cómo la planificación estratégica y la organización técnica pueden converger para crear una **experiencia multijugador eficiente y sostenible en el tiempo**, constituyendo un referente imprescindible para este tipo de videojuegos y para la industria entera.



Imagen 27, vista general de varios héroes jugables de Overwatch 2, uno de los principales atractivos del videojuego.

BIBLIOGRAFÍA Y REFERENCIAS

Documentación general sobre la estructura de red

Daposto. (2020, 31 julio). *Game Networking (9) — Bonus, Overwatch Model.* Medium.com. Recuperado 23 de noviembre de 2025.

<https://daposto.medium.com/game-networking-9-bonus-overwatch-model-4faba078cf05>

Battle(non)sense. (2016, 7 mayo). *Overwatch Netcode analysis* [Vídeo]. YouTube. <https://www.youtube.com/watch?v=vHO6Ky-w0UQ>

Ford, Timothy. 'Overwatch' Gameplay Architecture and Netcode. [Educational Resource, GDC 2017].

<https://www.gdcvault.com/play/1024001/-Overwatch-Gameplay-Architecture-and>

Liquidambar. *A guide to understanding netcode - Overwatch -*. GameReplays.org. (2016, 13 abril). Recuperado 25 de noviembre de 2025.

<https://www.gamereplays.org/overwatch/portals.php?show=page&name=overwatch-a-guide-to-understanding-netcode>

Documentación acerca de la programación y desarrollo técnico

Reed, Dan. *Networking Scripted Weapons and Abilities in «Overwatch»*. [Educational Resource, GDC 2017].

<https://www.gdcvault.com/play/1024653/Networking-Scripted-Weapons-and-Abilities>

<https://youtu.be/odSBJ49rzDo?si=y47ReYnpIDjh1QKJ>

Referencia bibliográfica anexada dentro del repositorio (formato PDF).

McDiffet, Lance. (2018, 10 noviembre). *The Math Behind Your Competitive Overwatch Match*. Medium.com. Recuperado 25 de noviembre de 2025.

<https://medium.com/data-science/the-math-behind-your-competitive-overwatch-match-a5184fc5a50f>

David Anthony Gibson. (2022, 28 marzo). *Overwatch Animation - How a Hero Is Mei'd GDC 2016* - David Anthony Gibson [Vídeo]. YouTube. <https://www.youtube.com/watch?v=FCzwvd80-iA>

Documentación acerca del apartado gráfico

Rendering Engine Architecture Conference. (2025, 3 julio). *REAC 2025 Evolving Global Illumination in Overwatch 2.* [Vídeo]. YouTube. <https://www.youtube.com/watch?v=0PlxPCq-DbQ>

Alain Galvan. (2020, 13 abril). *Overwatch Rendering Analysis | GameArch* [Vídeo]. YouTube.

https://www.youtube.com/watch?v=wvILHa_o4u8