



Universidad ORT Uruguay

Facultad de Ingeniería

Ingeniería en Sistemas

Programación de Redes

Obligatorio 2

Marcel Cohen - 212426

Facundo Arancet - 219606

Grupo M6A

Docente: Roberto Assandri

Índice

Índice	2
Introducción	3
Arquitectura del sistema	3
Cambios en el diseño del servidor de juego	4
Diseño de servicios expuestos por remoting	5
Diseño de servidor administrativo	5
Diagrama de componentes	6
Comunicación de la nueva aplicación cliente	7
Diagrama de secuencia para obtención del log	7
Diagrama de secuencia para obtención de los usuarios registrados	7

Introducción

En esta versión del sistema, se incorporó:

Un servidor administrativo: Es un nuevo servidor que brinda servicios de alta baja y modificación de usuarios, provee el log de la última partida jugada, los 10 puntajes más altos y estadísticas de las últimas 10 partidas jugadas.

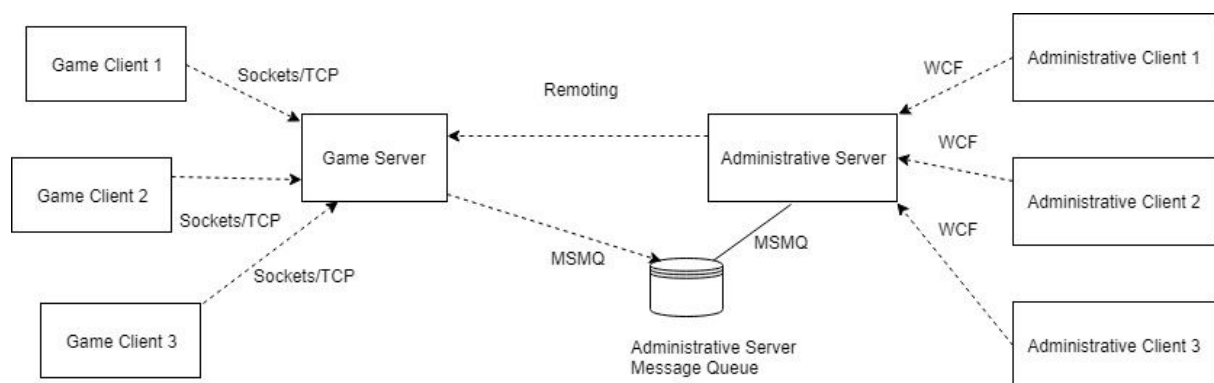
Un cliente administrativo: Es otra aplicación cliente que consume los servicios que brinda el nuevo servidor.

Para lograr la comunicación entre el servidor del juego (viejo), el servidor administrativo (nuevo) y el cliente administrativo (nuevo), se utilizaron las tecnologías vistas en la segunda mitad del curso, estas son:

- Message Oriented Middleware (MOM), se implemento mediante Microsoft Message Queues (MSMQ)
- Remote Procedure Call (RPC) y Object Request Broker (ORB). Se utilizó .NET remoting que es una implementación de la arquitectura ORB.
- Web Services. Se implementó utilizando el framework Windows Communication Foundation (WCF).

Arquitectura del sistema

El siguiente esquema ilustra las comunicaciones entre las 4 aplicaciones que constituyen el sistema, las dos aplicaciones servidor y las dos aplicaciones cliente.



La comunicación entre el cliente del juego y el servidor del juego se mantuvo igual, mediante sockets tcp/ip, no se realizó ningún cambio en esta versión.

El game server es quien tiene los datos almacenados (usuarios, estadísticas de partidas y puntajes), y expone mediante remoting servicios para obtener estos datos y modificarlos en el caso de los usuarios. Los servicios se exponen como singleton, para todas las demás aplicaciones que quieran acceder a ellos. De todas formas, este singleton no es el

almacenaje en sí, sino un servicio que accede al almacenaje que está en el gameServer, el cual no se expone. De esta forma, todas las aplicaciones, tanto las viejas como las nuevas acceden a los mismos datos.

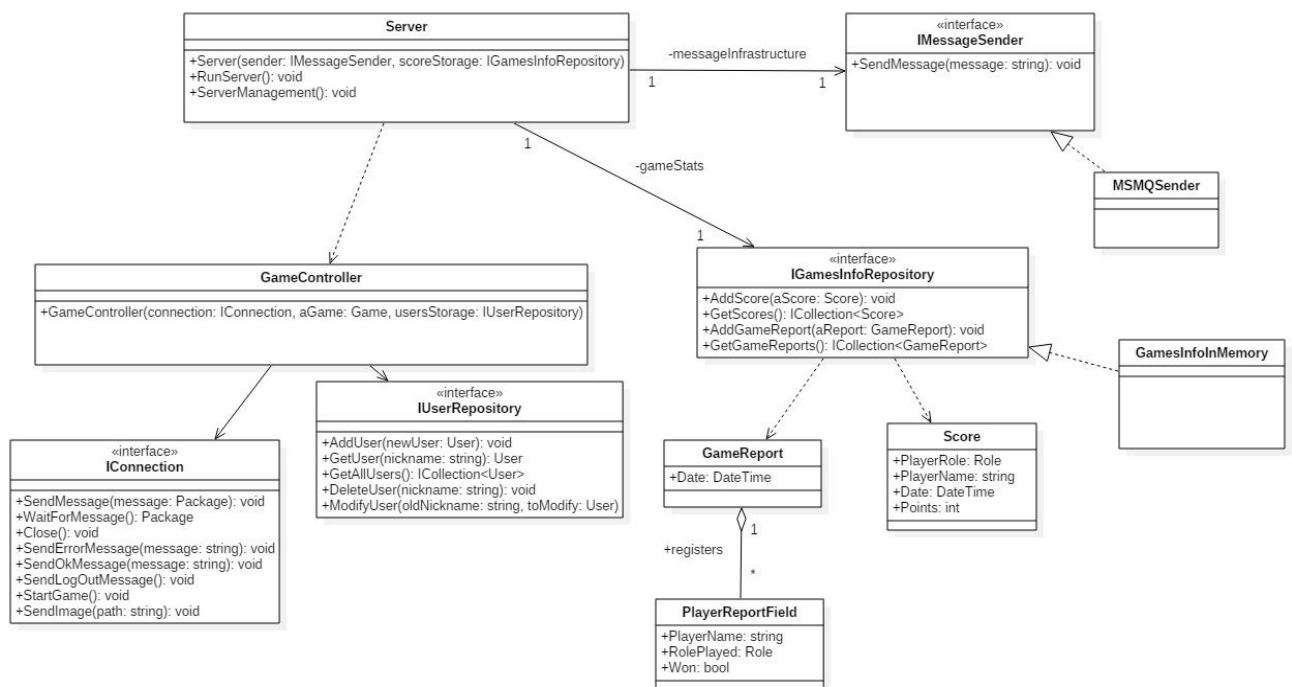
El servidor administrativo expone un servicio web con WCF, y el cliente administrativo tiene una referencia a este servicio. Para obtener y modificar datos, el cliente administrativo le envía solicitudes al servidor administrativo, y este a su vez accede a los servicios del servidor del juego remotamente, para realizar las acciones que le solicita el cliente.

Cada vez que se termina una partida, el servidor del juego manda el log de la partida por msmq a la cola de mensajes del servidor administrativo. Este servidor, toma el último mensaje de su cola de mensajes (el log de la última partida jugada) y lo envía a la aplicación cliente administrativo por WCF, cuando esta le solicita el log de la última partida jugada.

Cambios en el diseño del servidor de juego

En el siguiente diagrama ilustra el nuevo diseño del servidor a grandes rasgos.

Se crearon entidades para modelar el puntaje hecho por un jugador en una partida y para modelar la estadística de una partida.

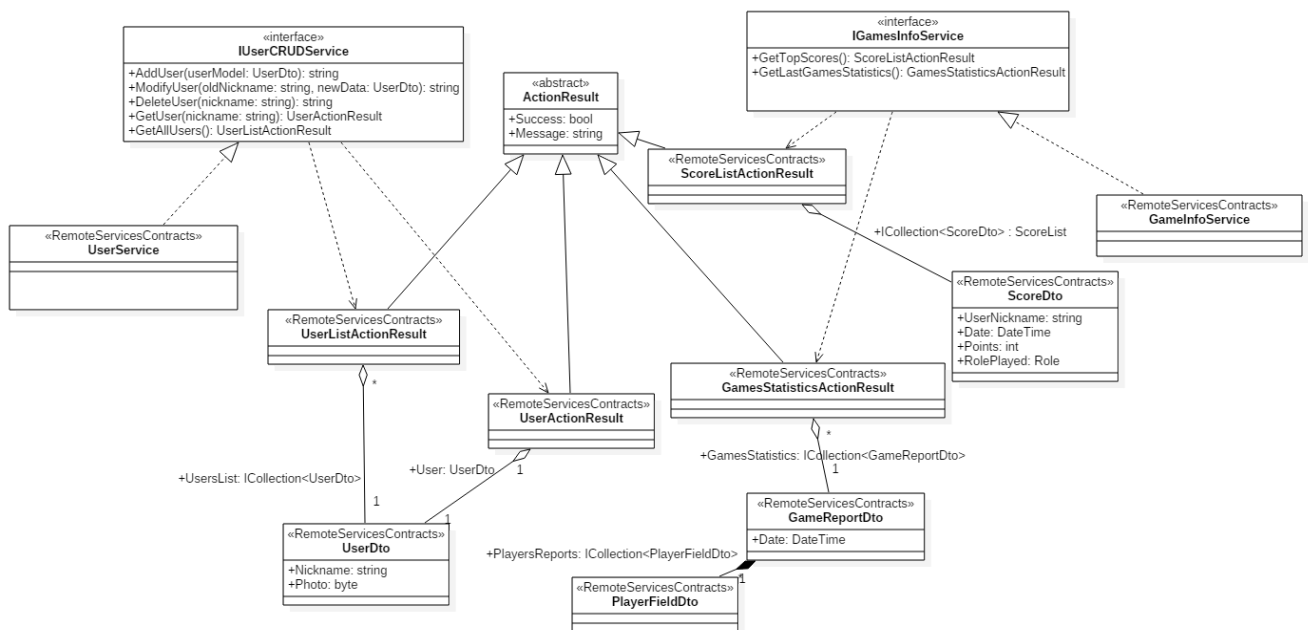


En este nuevo diseño, la clase Server tiene dos nuevas dependencias: una abstracción de un servicio de mensajería, que es por donde el servidor envía los logs de las partidas que terminan, y una abstracción de un repositorio de puntajes y reportes de partidas. La clase Server desconoce de sus implementaciones, el servicio de mensajería está implementado con MSMQ como ya se mencionó antes, y el repositorio está implementado en memoria.

Por último, se agregan más funciones al contrato del repositorio de usuario, ya que ahora también se pueden borrar y modificar usuarios.

Diseño de servicios expuestos por remoting

Para implementar el remoting, como ya se mencionó, se exponen dos servicios, uno para el abm de usuarios y el servicio de puntajes y estadísticas de partidas. Para esto, se crea el proyecto RemoteServicesContracts, que contiene las interfaces de los servicios y los dtos a intercambiarse. Este es el proyecto común a ambas aplicaciones que se comunican mediante remoting, define el “contrato” de los servicios, qué funcionalidades ofrece y que dtos se intercambian.



Se definen dos tipos de objetos: los Dtos, que son representaciones de los objetos del dominio, y ActionResults, que son objetos que representan la respuesta del servicio. El ActionResult tiene un atributo que indica si la acción se hizo exitosamente o no, un mensaje, que puede ser de confirmación o de error, y el dato solicitado (Dto). Estas clases son la forma que se tiene de informar sobre errores, ya que no se pueden lanzar excepciones de una aplicación a otra.

Diseño de servidor administrativo

El servidor administrativo expone un servicio web. La implementación de este servicio web tiene una referencia a un servicio que devuelve el log de la última partida, y a los dos servicios que accede por remoting, del servidor del juego. El servicio de logs, (ILogManager), está implementado con MSMQ, consigue el log de la última partida de la cola de mensajes del servidor. Como ya se dijo, el servidor del juego envía los logs a esta

cola. Lamentablemente, no se pudo aprovechar las ventajas de estas interfaces definidas, por que no se logró inyectar las dependencias al servicio web que se expone.

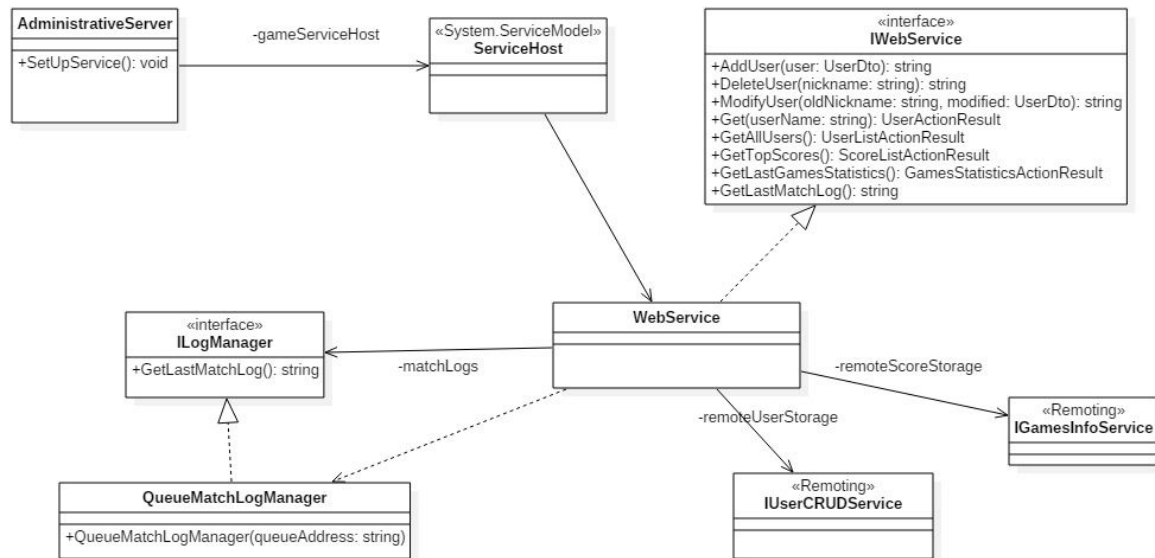
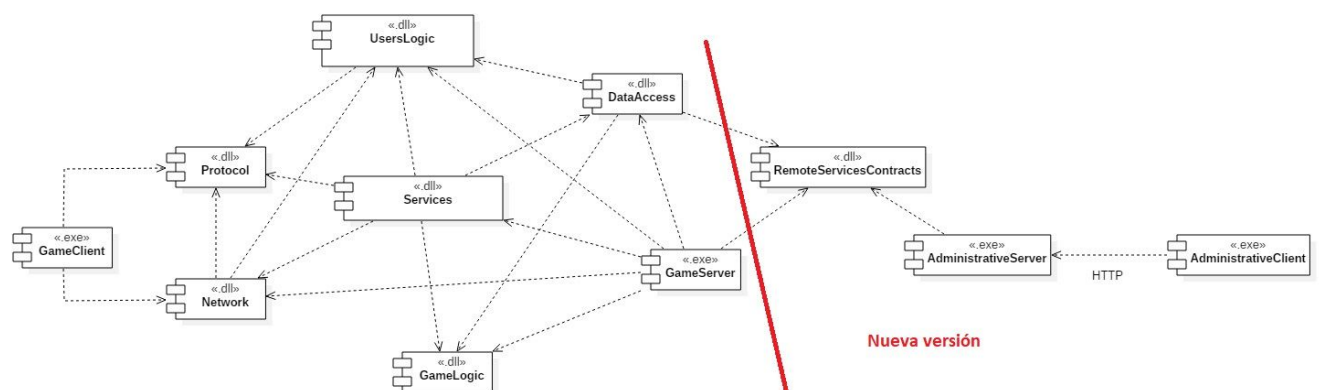


Diagrama de componentes

En esta versión se crean 3 proyectos nuevos:

- RemoteServicesContracts, que contiene los contratos de los servicios remotos y los objetos de transferencia de datos que estos intercambian.
- AdministrativeServer, que contiene la aplicación del servidor administrativo. Es la que contiene el servicio web.
- AdministrativeClient, que contiene la aplicación cliente que consume el servicio web.



Comunicación de la nueva aplicación cliente

Diagrama de secuencia para obtención del log

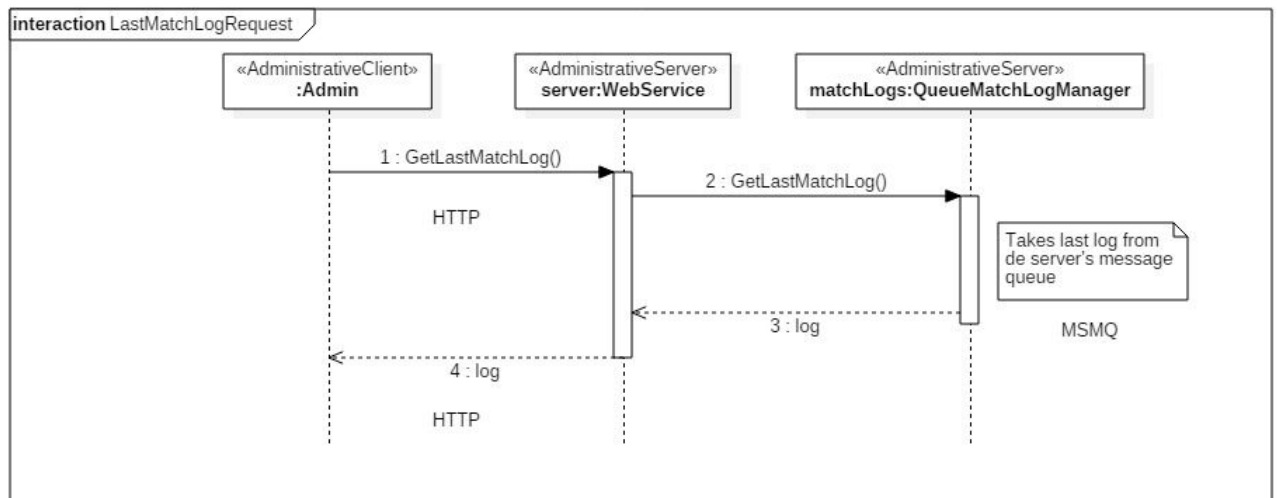


Diagrama de secuencia para obtención de los usuarios registrados

