



Universidad ORT Uruguay

Facultad de Ingeniería

Programación de Redes

Obligatorio 1

Entrega como requisito de la materia

Programación de Redes

Julieta Percivale 245079 - Hernán Reyes 235861

Tutor: Roberto Assandri - Delia Malvina Alvarez

22 de Noviembre de 2021

Índice

Descripción General del trabajo/sistema

Alcance del sistema

Principales decisiones

Funcionalidades

Descripción y justificación de diseño

Comunicaciones

Módulo cliente

Módulo servidor

Módulo WebApiAdministrativa

Módulo ServidorAdministrativo

Módulo LogServidor

Capa de Servicios

Diagrama de paquetes

Anexo

Principales decisiones entregas anteriores

Descripción General del trabajo/sistema

En esta tercera instancia se implementaron dos nuevos módulos, el servidor de logs y el servidor administrativo.

Contamos así con cuatro módulos en nuestra plataforma. Cliente, servidor cliente, servidor administrativo y servidor de logs.

A nivel funcional nuestro cliente y servidor cliente no sufrieron cambios por tanto siguen respetando las decisiones tomadas en entregas anteriores así como también cuentan con la misma estructura con algunas modificaciones menores.

Por otra parte se implementaron dos nuevos módulos. El servidor de logs y el servidor administrativo.

Para el servidor de logs se implementaron los requerimientos funcionales exigidos en la letra como es la recepción de logs desde el servidor principal y el filtrado de los mismos.

Por último, para el servidor administrativo, también se realizaron todos los requerimientos especificados como es el ABM de usuarios, ABM de juegos y asociar/desasociar un juego a un usuario.

En cuanto a requerimientos no funcionales se utilizaron las tecnologías de GRPC, RabbitMQ y REST Api así como también evitamos el uso de Threads como fue solicitado.

Alcance del sistema

En esta entrega pudimos cumplir con todos los requerimientos funcionales y no funcionales que se especificaba en la letra.

Dentro de los requerimientos funcionales se tomaron algunas decisiones, además de las decisiones de nuestras entregas anteriores, donde se especifica en detalle en la sección de [“Principales decisiones - Funcionalidades”](#).

- Mejoras a tener en cuenta

-
- Nuestra WebApi de logs no fue dividida en capas, como si fue dividida la WebApiAdministrativa, por temas de tiempo.
La misma debería de contar con una capa de servicios y otra capa de repositorios.

Principales decisiones

Además de las decisiones realizadas para las instancias anteriores, [“Principales decisiones entregas anteriores”](#), realizamos las siguientes.

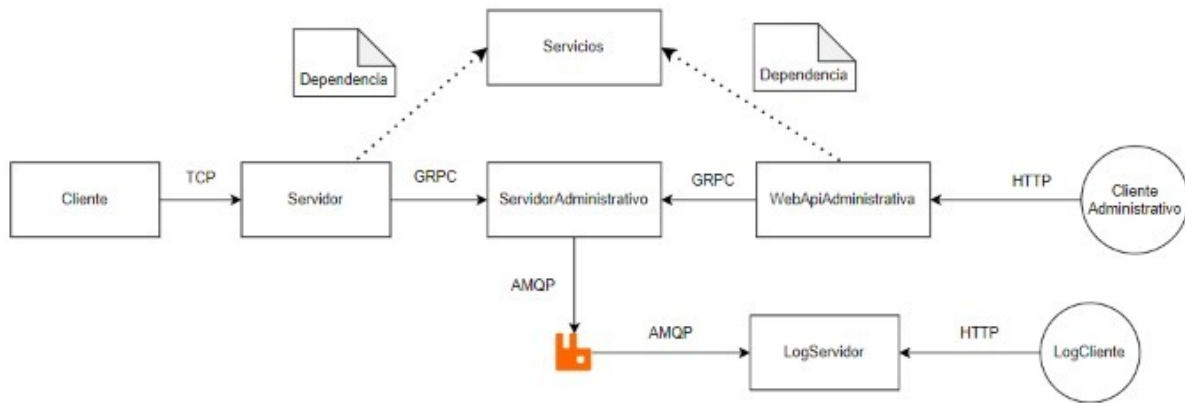
Funcionalidades

- **IMPORTANTE.** Por decisiones de entregas anteriores el obtener usuario del sistema devuelve un usuario existente o en caso de no existir lo da de alta no se pudo realizar el endpoint de obtener un usuario específico. A su vez el endpoint de alta de usuario siempre va a devolver un código 201 (Created) dada nuestra lógica de negocio. Si el usuario no existe lo doy de alta, caso contrario devuelve el existente.
- El servidor de logs es capaz de filtrar los mismos por fecha, usuarios y juegos. En caso de recibir los 2 o 3 parámetros se filtran todos los logs que cumplan al menos con una coincidencia de algún filtro. NO es un filtrado excluyente.
- El servidor de logs es capaz de filtrar los mismos por fecha, usuarios y juegos. En caso de recibir los parámetros de juego y/o usuario el filtrado es el siguiente. Si recibimos por ejemplo el nombre de usuario “Mario” el servidor verifica en los logs de tipo usuario la ocurrencia de la palabra “Mario” en dichos logs, mismo caso para juegos.

Descripción y justificación de diseño

Nuestro sistema fue diseñado para que cada módulo cumpla con una única responsabilidad.

En el siguiente diagrama podemos ver a grandes rasgos la comunicación entre los distintos módulos del sistema.



Comunicaciones

- Cliente-Servidor: Se conectan mediante TCP.
- Servidor-ServidorAdministrativo: Se conecta el servidor al servidor administrativo con grpc.
- WebApiAdministrativa-ServidorAdministrativo: Se conecta la webApi al servidor administrativo con grpc.
- ServidorAdministrativo-LogServidor: El servidor administrativo manda mensajes a la cola mediante RabbitMQ.
- ClienteAdministrativo-WebApiAdministrativa: Se conecta el cliente a la WebApi mediante HTTP.
- LogCliente-LogServidor: Se conecta el cliente log a la WebApi mediante HTTP.

Módulo cliente

Nuestro módulo cliente es el encargado de recibir y manejar las peticiones de los clientes de consola.

Módulo servidor

Nuestro módulo servidor es el encargado de recibir y manejar las peticiones de los clientes de consola.

El mismo es un cliente GRPC que se conecta al servidor GRPC (ServidorAdministrativo). Cuando el cliente realiza una petición este servidor utiliza una capa de servicios que realiza toda la lógica de nuestro negocio y una vez procesada la petición la delega al ServidorAdministrativo.

Módulo WebApiAdministrativa

Dicho módulo implementa una WebApi donde clientes a través de HTTP le realizan distintas request.

Dicho módulo trabaja prácticamente de igual manera que el módulo Servidor. Los clientes a través de HTTP realizan sus peticiones y nuestro módulo utiliza la capa de servicios para realizar la lógica de dicha petición y una vez procesada la petición la delega al ServidorAdministrativo.

Módulo ServidorAdministrativo

El mismo implementa un servidor GRPC. También es un productor de RabbitMQ, donde por cada request realizada ingresa a la cola un nuevo log.

El flujo de dicho módulo consiste en que clientes a través grpc realizan sus peticiones ya procesadas por el módulo de la capa de servicios y la única responsabilidad del ServidorAdministrativo es recibir las mismas y dependiendo realizar modificaciones sobre nuestros datos o devolverlos.

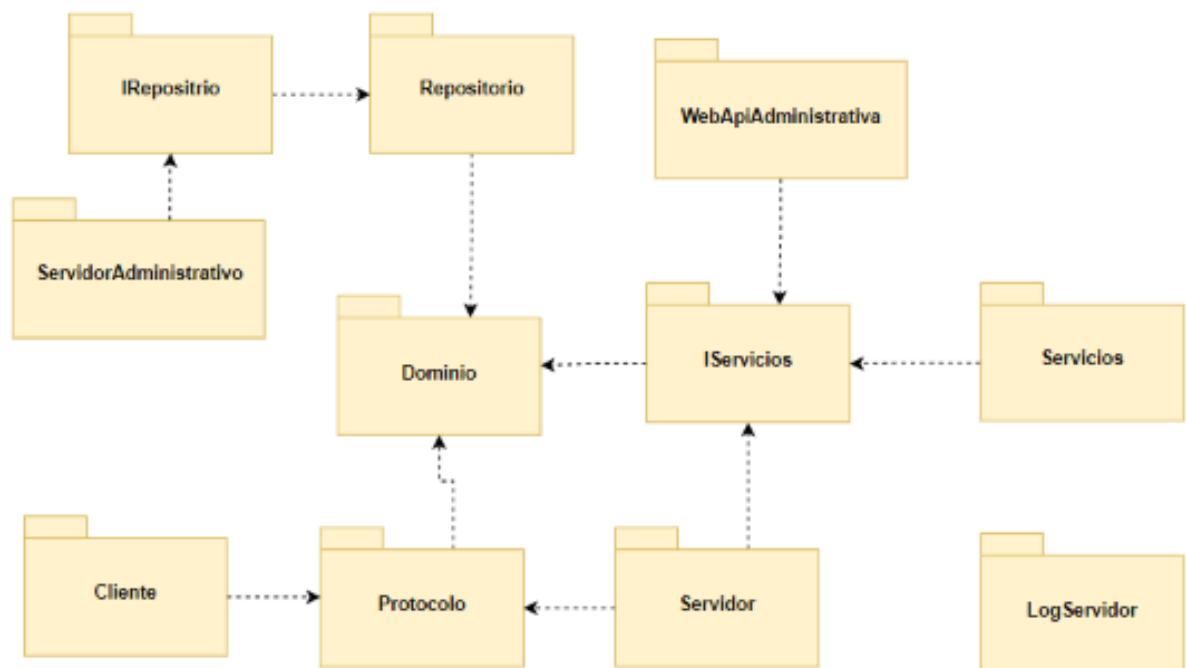
Módulo LogServidor

Nuestro LogServidor implementa una WebApi además de ser un consumidor de RabbitMQ. Sus principales responsabilidades es consumir de la cola, producida por el ServidorAdministrativo, los distintos logs generados. Por otra parte también recibe peticiones de clientes a través de HTTP para poder consultar los distintos logs del sistema.

Capa de Servicios

La única responsabilidad de dicha capa es realizar, a partir de peticiones, distintas validaciones correspondientes a nuestra lógica de negocio.

Diagrama de paquetes



Anexo

Principales decisiones entregas anteriores

Funcionalidades

- Se decidió que el servidor solo cuente con las operaciones de cerrar el servidor y ver lista de juegos. El resto de las funcionalidades pedidas quedan del lado del cliente siendo procesadas por el servidor.
- Existen usuarios, los cuales solo sirven para identificar quién realizó una calificación. No cuentan con contraseña y solo puede existir un único nombre de usuario en el sistema. Si se intenta iniciar sesión con un usuario que no existe se da de alta automáticamente y se inicia sesión. En caso de existir el usuario solo iniciamos sesión con el mismo.
- Buscar por calificación. En una calificación se ingresa cantidad de estrellas (entero del 1 al 5). Al buscar por calificación se está buscando por el promedio de esas estrellas, es decir, la suma de cada estrella por calificación dividido por la cantidad de calificaciones.
Ej. Si cuento con un juego con una calificación de 5 estrellas y otra calificación para el mismo juego de 3 estrellas el promedio va a ser 4. Por tanto al buscar por calificación 4 nos devolverá dicho juego.
- El título de un juego debe ser único.
- Si se elimina la carátula del lado del servidor (eliminar el archivo de la carpeta) el cliente ya no podrá ver el detalle de dicho juego. Dado que el archivo (carátula) fue eliminado.
- Para las funcionalidades de baja y modificación de usuarios del lado del servidor se permiten realizar las mismas únicamente si dicho usuario a eliminar o modificar no está logueado en el sistema en dicho momento.
Por tanto si queremos realizar las operaciones debe ser un usuario inactivo, es decir un usuario que no está siendo utilizado en dicho momento por un cliente.