

Universidad ORT Uruguay

Facultad de Ingeniería

Bernard Wand Polak



Programación de Redes

26 de Octubre de 2021

[Repositorio](#)

Marco Fiorito N° Est.: 227548

Nicolás Praderi N° Est.: 214540

Grupo N6A

Docente: Roberto Assandri

Declaración de autoría	3
Alcance de la aplicación	4
Aplicación de Servidor Administrativo	5
Aplicación de Servidor de logs	5
Arquitectura de la aplicación	6
Diseño los componentes.	6
Diagrama de deploy	6
Diagrama de clases:	7
Cliente:	7
Protocol:	8
ServerGRPC:	9
Admin Server:	10
Logger:	12
Mecanismos de comunicación de los componentes	15
Aplicación Cliente y aplicación Server	15
Datos de prueba	17
Aclaraciones para ejecutar la aplicación	18

Declaración de autoría

Nosotros, Marco Fiorito y Nicolás Praderi, declaramos que el trabajo que se presenta en esa obra es de nuestra propia mano. Podemos asegurar que:

- La obra fue producida en su totalidad mientras realizábamos Programación de Redes.
- Cuando hemos consultado el trabajo publicado por otros, lo hemos atribuido con claridad.
- Cuando hemos citado obras de otros, hemos indicado las fuentes. Con excepción de estas citas, la obra es enteramente nuestra.
- En la obra, hemos acusado recibo de las ayudas recibidas.
- Cuando la obra se basa en trabajo realizado conjuntamente con otros, hemos explicado claramente qué fue contribuido por otros, y qué fue contribuido por nosotros.
- Ninguna parte de este trabajo ha sido publicada previamente a su entrega, excepto donde se han realizado las aclaraciones correspondientes.

Alcance de la aplicación

La aplicación realizada comprende todas las funcionalidades solicitadas.

El sistema cuenta con dos aplicaciones, un servidor en el que se maneja la información relacionada a los juegos que pueden obtener los usuarios y archivos relacionados con los mismos. Además, contamos con una aplicación cliente para dicho servidor que se encargará de la interacción de los usuarios con el sistema. Adicionalmente ahora se comprende la generación de logs del sistema y su filtrado y un Servidor Administrador.

En este documento se presentan todos los detalles de la implementación del sistema.

Nuevas funcionalidades

Aplicación de Servidor Administrativo

SARF1. ABM de Usuarios. Se deberá dar de alta usuarios tomando en cuenta las **restricciones ya existentes..**

SARF2. ABM de Juegos. Se deberá dar de alta juegos tomando en cuenta las **restricciones ya existentes.**

SARF3. Asociar y desasociar un juego a un usuario. Luego de que un juego fue dado de alta,
será posible asociarlo y desasociarlo a un usuario.

Aplicación de Servidor de logs

SLRF1. Recepción de logs desde el servidor principal. El servidor de logs debe recibir logs de los eventos generados en el servidor principal.

SRF2. Filtrado de logs. El servidor de logs debe ser capaz de filtrar los logs por juegos, usuarios y fecha.

Arquitectura de la aplicación

A continuación se presenta una descripción de la arquitectura presentada como solución, haciendo énfasis en la infraestructura utilizada, la estructura interna de cada uno de los componentes, y los mecanismos de comunicación entre ellos.

Diseño los componentes.

Diagrama de deploy

1. **Servidor Vapor:** Corresponde al servidor donde se reciben las peticiones de los clientes.
2. **Cliente:** Corresponde al cliente el cual consume los servicios que ofrece el servidor.
3. **Server Administrativo:** Corresponde al servidor administrativo que **comunica al servidor vapor mediante protocolo GRPC**
4. **Servidor de Logs:** Corresponde al servidor de logs que **consume las interacciones del servidor mediante colas de mensaje MQ.**

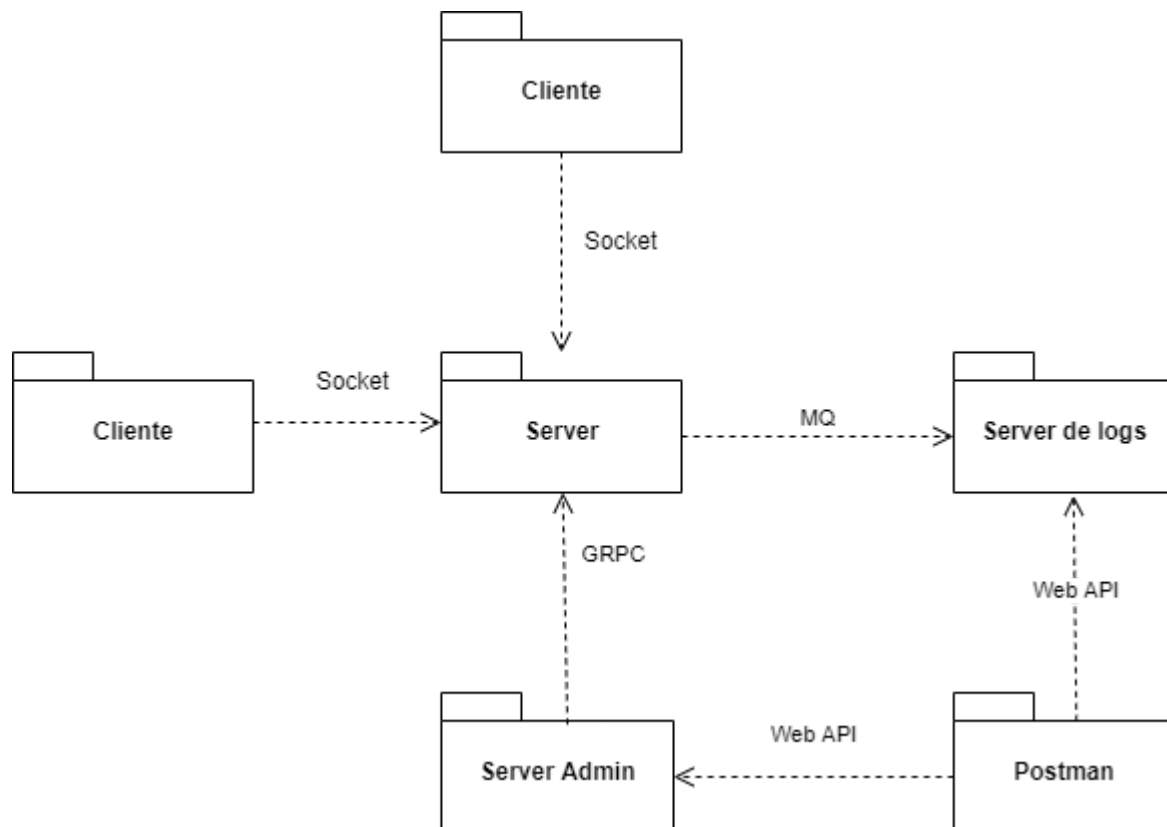
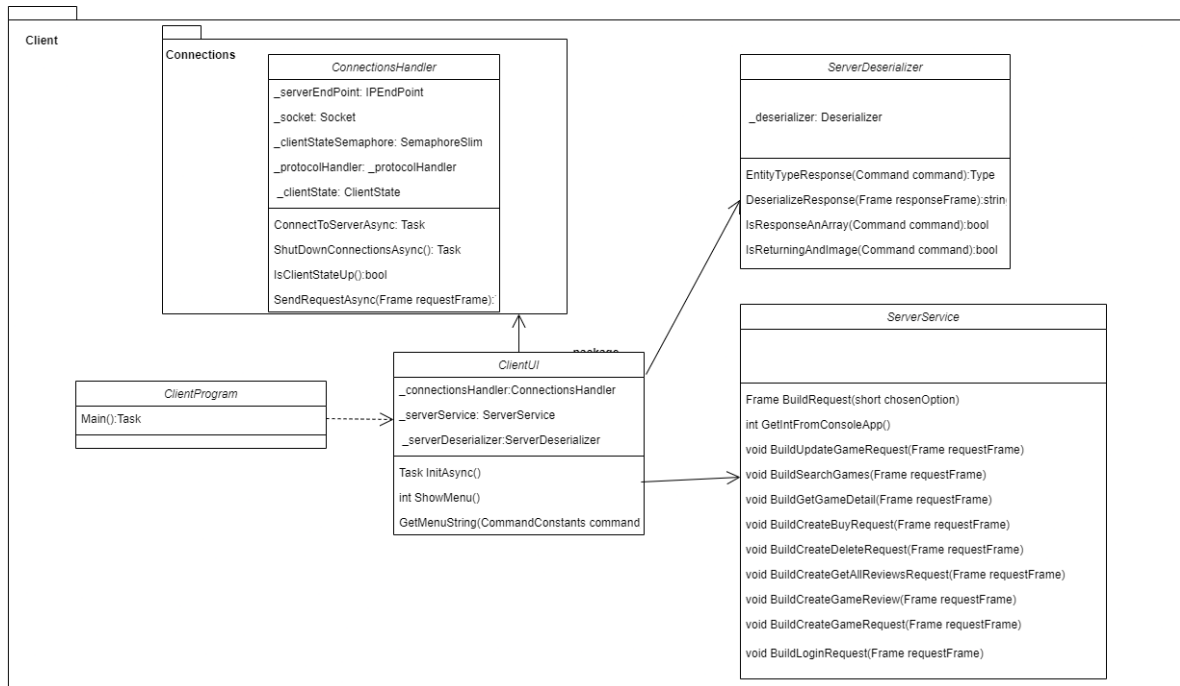
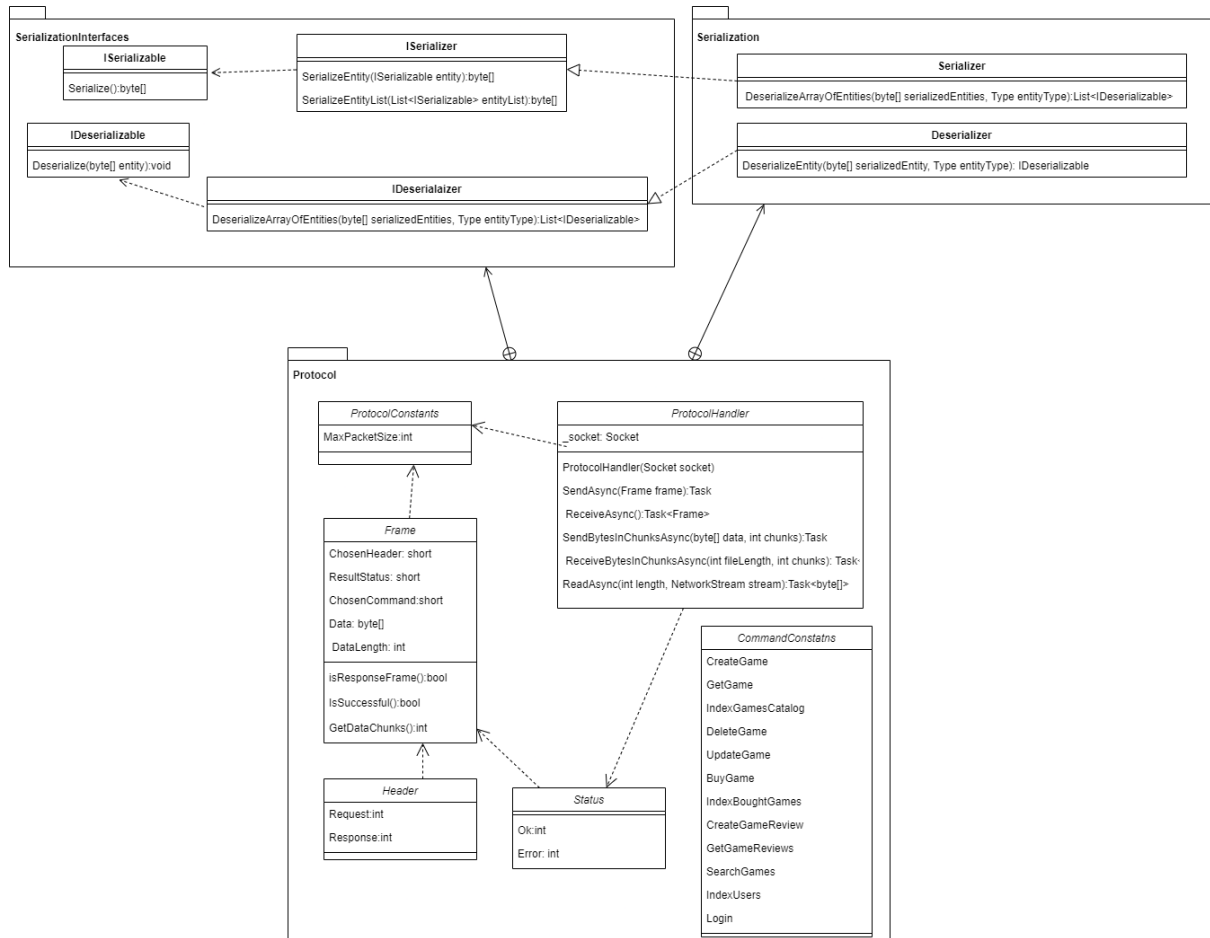


Diagrama de clases:

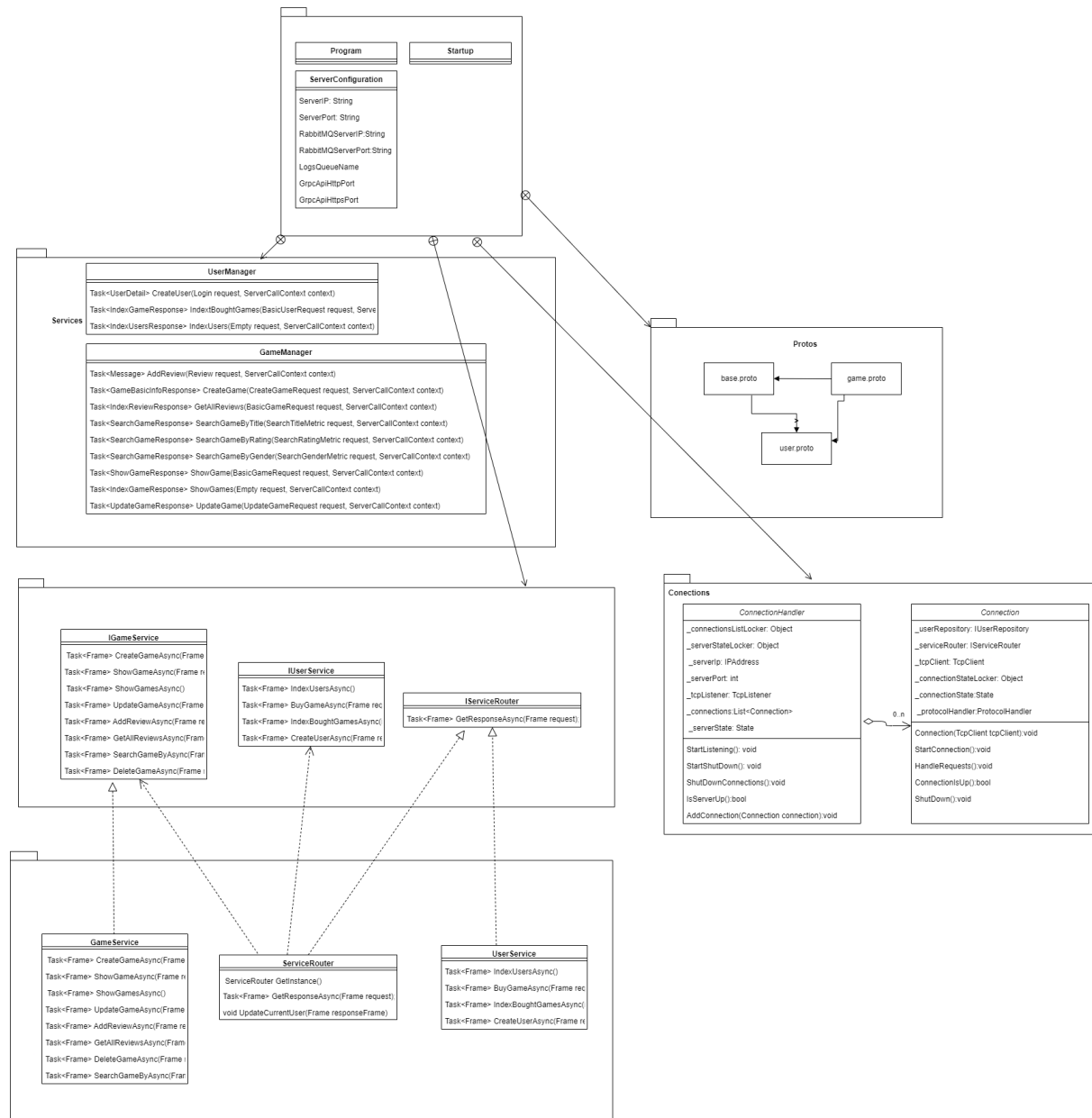
Cliente:

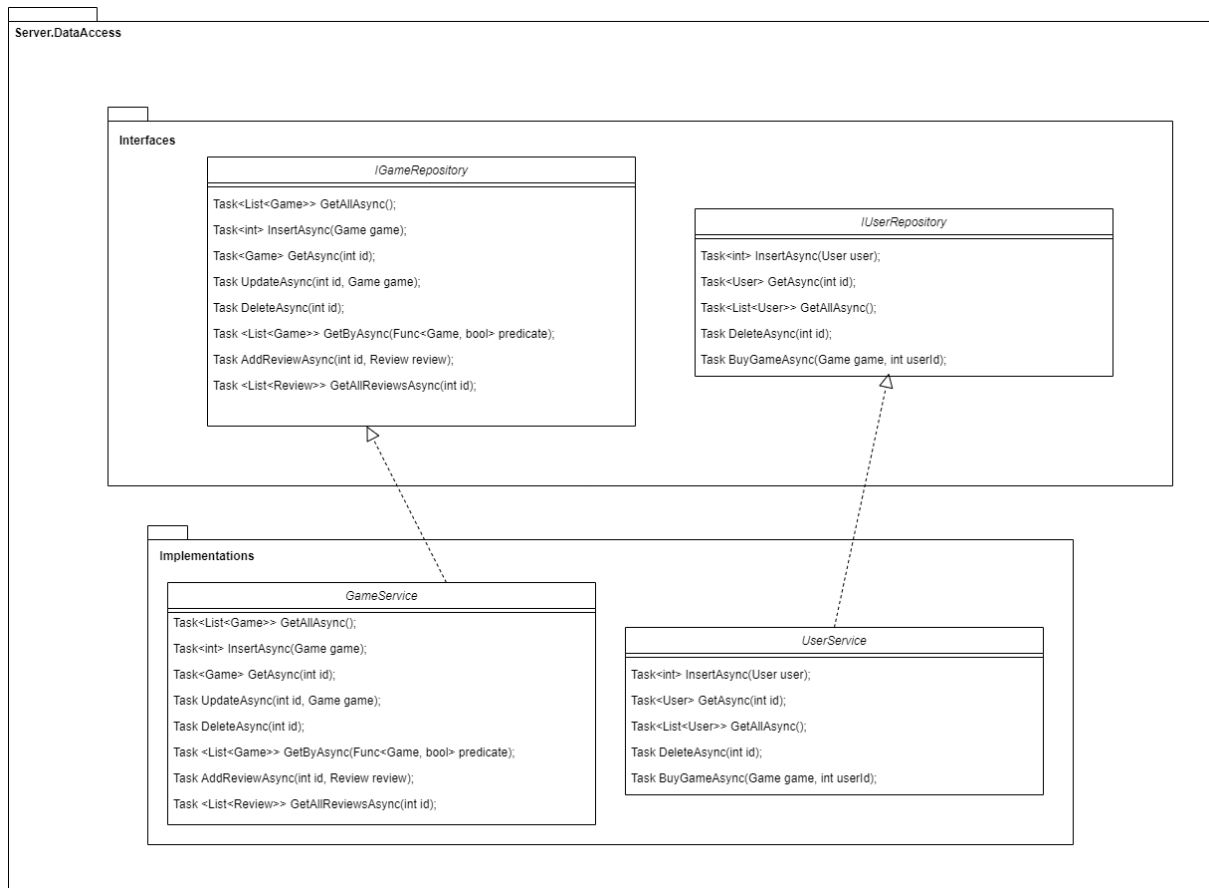
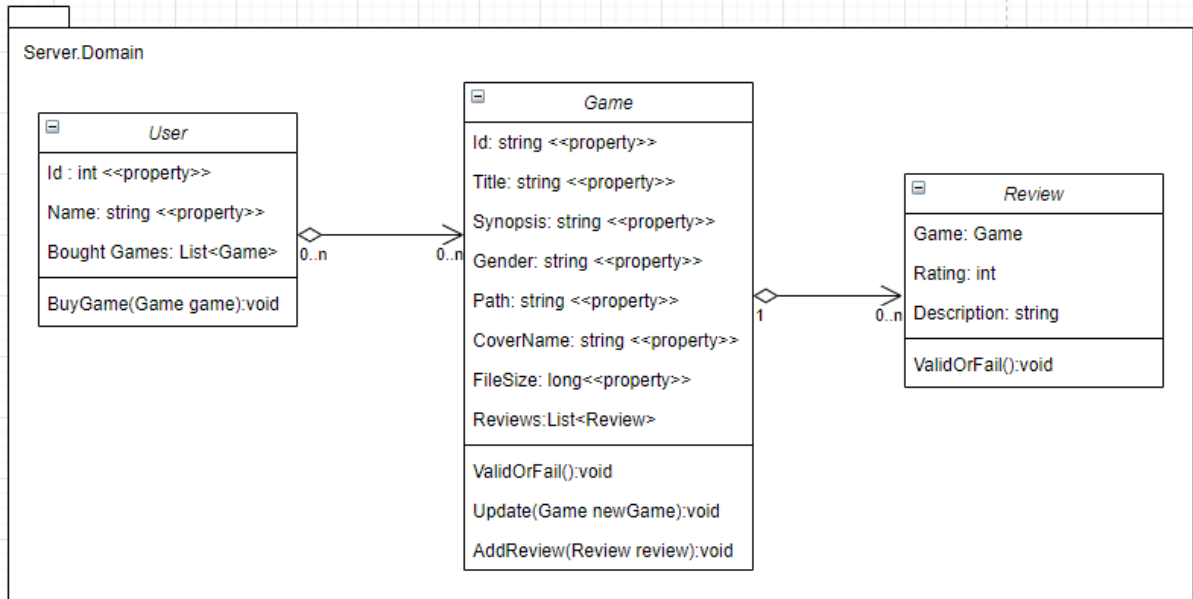


Protocol:

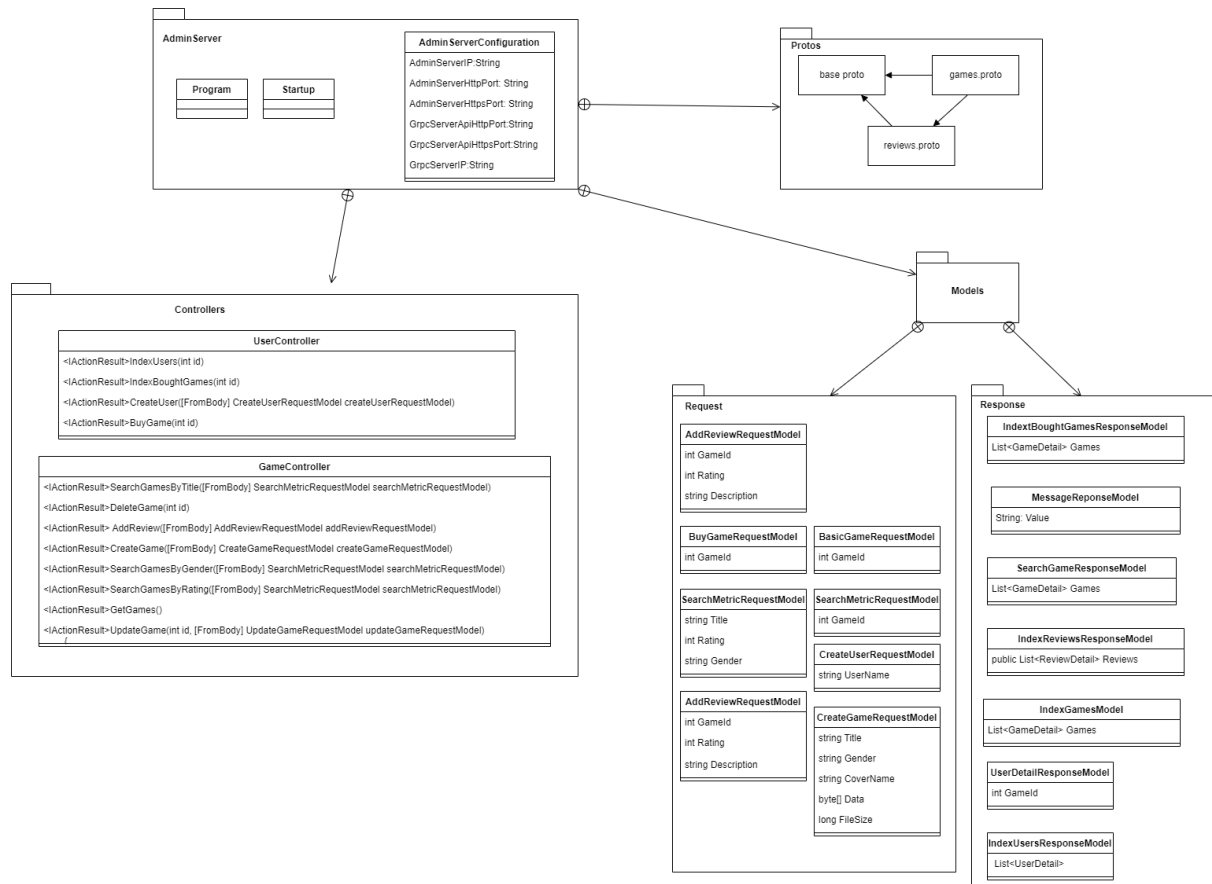


ServerGRPC:

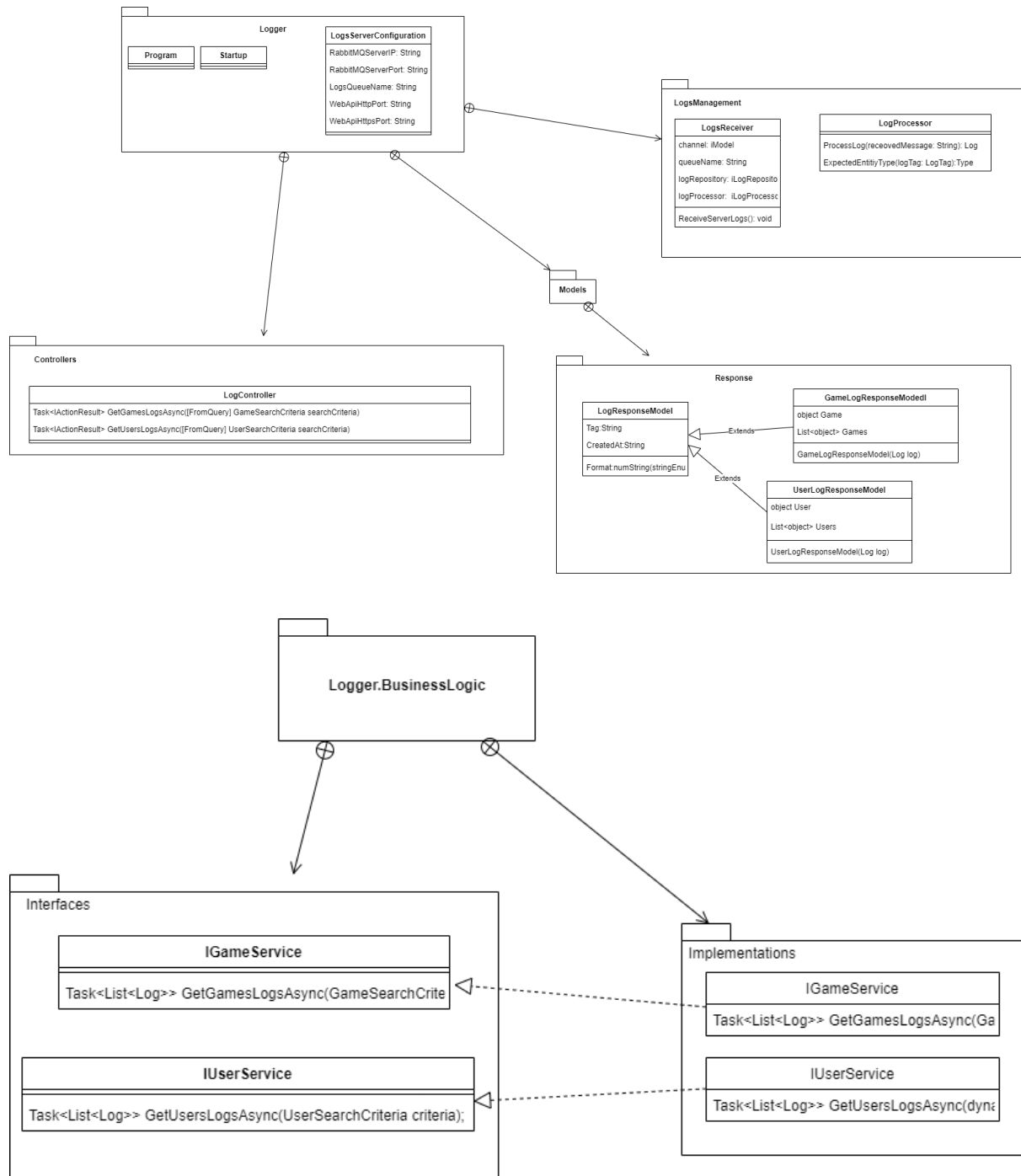


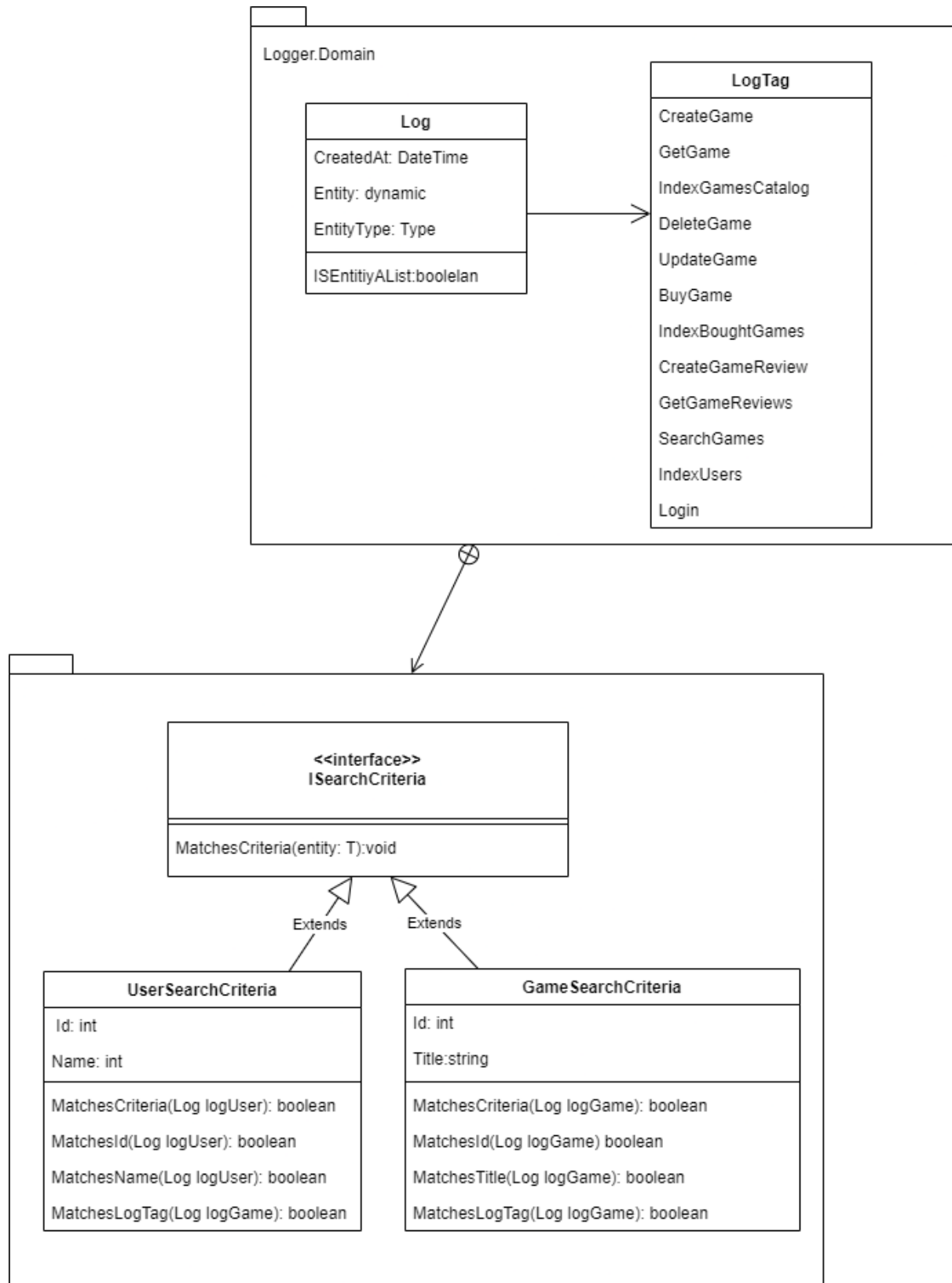


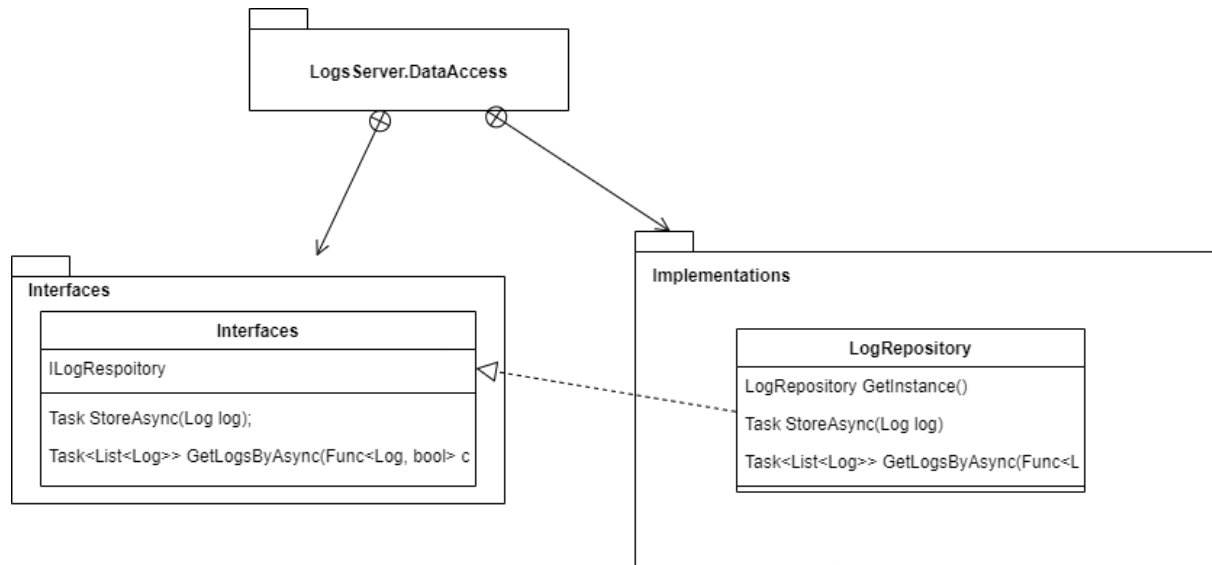
Admin Server:



Logger:







Mecanismos de comunicación de los componentes

Aplicación Cliente y aplicación Server

El protocolo utilizado para la comunicación es el siguiente:

Nombre del campo	Header	Command	DataLength	Data(optional)	Result
Valor	RES/REQ	0 - 11	Int	Variable	OK/ERROR
Largo	2	2	4	Variable	2

Se utilizaron los siguientes comandos para comunicar según la funcionalidad requerida:

Comando	Valor
Disconnect from server	0
Create a game	1
Get game information	2
Get all games	3
Delete a game	4
Update game info	5
Buy a game	6
Show my bought games	7
Give a review	8
Get game reviews	9
Search games by title, gender or rating	10
Get server users	11

En caso de que el usuario no esté autenticado los comandos son:

Comando	Valor
Disconnect from server	0

Login	1
-------	---

Aplicación ServerGrpc y aplicación Logger

Para la comunicación de estos dos componentes utilizamos RabbitMQ utilizando una cola de mensajes.

Cada vez que se realiza una acción en ServerGrpc a nivel de Services se envía un mensaje al Logger, el cual lo recibe, lo procesa y lo guarda.

Aplicación ServerGrpc y aplicación AdminServer

Para la comunicación de estos dos componentes utilizamos Grpc utilizando una cola de mensajes.

Los protos definidos análogamente entre ambos módulos define la forma en la cuál se comunican.

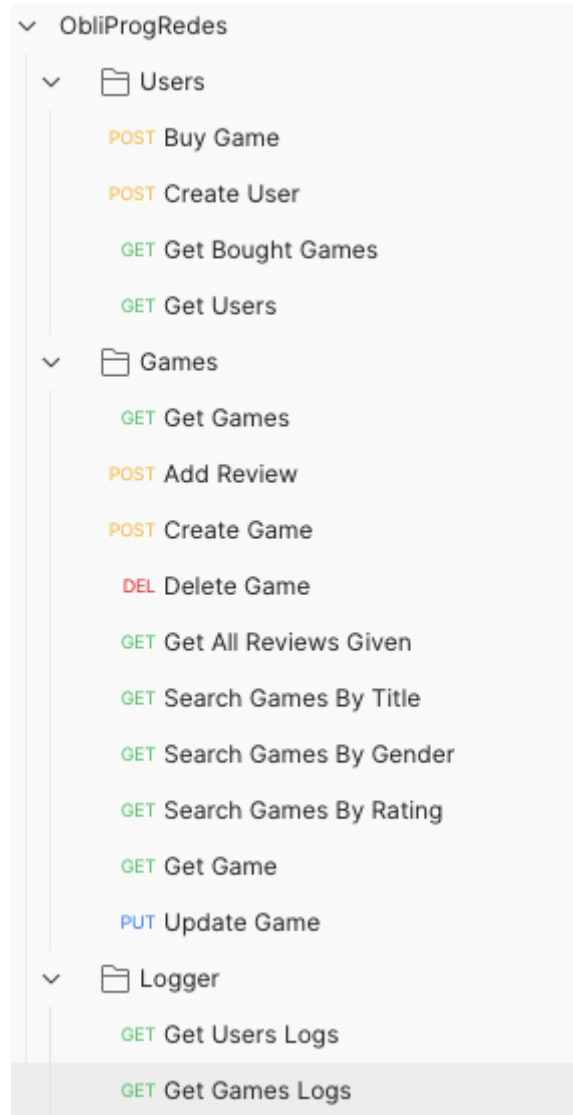
Tuvimos que realizar una configuración extra para soportar más de un proto a nivel de proyecto.

Aplicación AdminServer y Logger con el usuario

Se puede acceder a las apis expuestas por estos servidores desde la colección de Postman que se encuentra en el repositorio.

Datos de prueba

En el repositorio dentro de la carpeta Postman se encuentra una colección de postman la cual utilizamos para probar los distintos servidores.



Aclaraciones para ejecutar la aplicación

Para ejecutar la aplicación deberemos tener las siguientes consideraciones:

- Primero se debe ejecutar el ServerGrpc, luego el AdminServer y por último el Logger.
- Las aplicaciones compiladas en Release fueron generadas desde Mac y por lo que vimos no generó un .exe. Para correr la aplicación se deberá hacer mediante el comando dotnet utilizando las siguientes dlls:
 - Server.dll dentro de la carpeta Server.
 - Client.dll dentro de la carpeta Client.
- En caso de que sea necesario ya sea el caso se pueden configurar los puertos e Ips utilizados en el appsettings.json.
- El proyecto Client sigue configurado como en la entrega anterior, por defecto los App.config vienen configurados utilizando la ServerIP 127.0.0.1, ServerPort 6000 y ClientIP 127.0.0.1. En caso de querer cambiarlo deberán abrir el App.config de cada proyecto y configurarlo.