****

**Instituto Tecnológico Autónomo de México**

**¡PÉGALE AL MONSTRUO!**

**Proyecto Alpha**

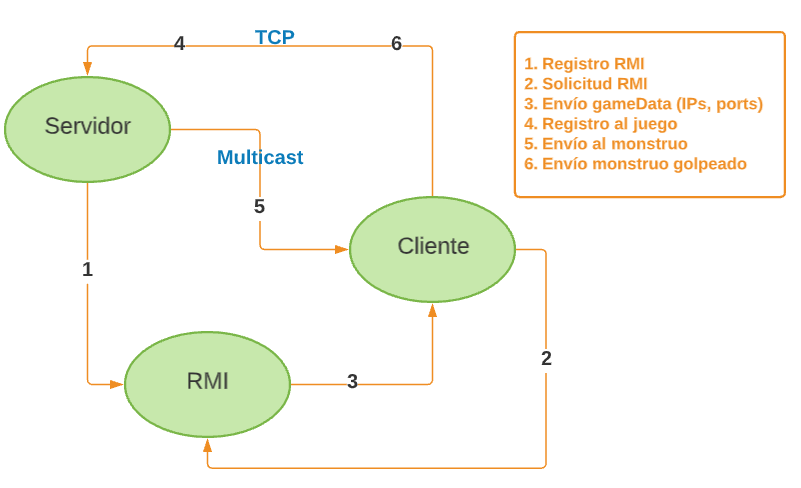
**17 de marzo de 2021**

|  |  |
| --- | --- |
| **Preparado para:** | **Preparada por:** |
| **Profesor: José Octavio Gutiérrez García**  **Materia: Sistemas Distribuidos**  **Periodo: Primavera 2021** | **Integrantes del equipo:**  **Claudia Denisse Campos González 143495**  **Luis Gerardo Godfrey Castañeda 142232**  **Luciano Montes de Oca Villa 173670** |

* **Introducción**

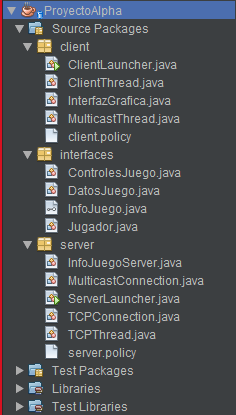
Los estresadores nos permiten observar el tiempo de respuesta de los servidores dependiendo de los clientes que acceden y las peticiones que se le hacen.

Con los estresadores podemos ir cambiando las características y configuraciones para saber la carga de trabajo que puede soportar el programa.

* **Desarrollo**

Para el desarrollo del juego, se armo primeramente el diagrama de secuencia, con el que podemos darnos una idea más general problema.

Con esto dividimos el proyecto en 3 packages que nos permiten diferenciar correctamente a las clases involucradas: client, server e interfaces.



Para el package client tenemos las clases que se encargan de instanciar al cliente con su respectiva interfaz gráfica y hacen la petición de los datos del juego.

Para el package interfaces tenemos los objetos adicionales para complementar el funcionamiento del sistema.

Y para el package server tenemos la implementación de la información del juego, la que se encarga del manejo de mensajes Multicast, la que registra el servicio RMI y crea los hilos TCP para enviar y recibir los mensajes de los jugadores.

Tanto en client como en server tenemos un archivo tipo policy que contiene los permisos necesarios.

Para el estresamiento se quitó el uso de interfaces gráficas (GUI) y se realizó cambios en el código de 4 clases:

1. ***ClientLauncher***: se cambió la implementación de un solo cliente con el uso de un for para la creación de 50, 100, 150, … hasta 500 clientes. También se agrega el número de peticiones como parámetro del hilo del cliente.
2. ***GameControl***: Aquí se guardan los tiempos de ejecución en un archivo txt para cada número de clientes (En total tenemos 10 archivos).
3. ***ServerLauncher***: Se dejo que hubiera una sola ejecución del programa para poder guardar los datos de medición más fácilmente.
4. ***MulticastThread***: Se modificaron los tiempos de envío de mensajes TCP y Multicast, siendo estos aleatorios.

Con los cambios realizados, corrimos el estresamiento para el número de clientes que correspondían (50, 100, 150, … hasta 500 clientes) en 2 máquinas diferentes:

1. Windows 10 de 64 bits con Intel Core i7, 2.20GHz, 16 GB de RAM, 1TB HDD y 256 GB SSD
2. Windows 10 de 64 bits con AMD Ryzen 7, 8 GB de RAM 2.20 GHz, 1 TB HDD y 256 GB SSD

Se tomaron los tiempos de registro RMI y de respuesta en el juego.

* **Resultados**
* **Conclusiones**