Proyecto 1: Simulador Coronavirus
Int. Sistemas Distribuidos 2020-1

Juan Sebastián Prado Valero
Ingeniería de Sistemas
Pontificia Universidad Javeriana
Bogotá, Colombia
j_prado@javeriana.edu.co

Nicolás Camacho Plazas
Ingeniería de Sistemas
Pontificia Universidad Javeriana
Bogotá, Colombia
nicolas-camacho@javeriana.edu.co

Emanuel Álvarez Sánchez
Ingeniería de Sistemas
Pontificia Universidad Javeriana
Bogotá, Colombia
alvarez_emanuel@javeriana.edu.co

1. Definición del proyecto

El objetivo principal de este proyecto es crear un sisema que preste un servicio enfocado en una problemática de la sociedad actual usando para ello, paradigmas de comunicación distribuida y agentes móviles. El proyecto simulará el comportamiento de la pandemia producida por el COVID-19. Este contará con comunicación entre países y brokers.

2. Especificación de requerimientos y limitaciones

2.1. Sistema

El sistema debe implementar una aplicación distribuida que simule la propagación de un virus a nivel global (implementado en este caso a nivel de Centro y Sur América).

2.2. Balanceo de cargas y agentes móviles

Debe existir un esquema de *brokers* que permita mantener balanceado del sistema, esto con la intención de poder montar mas de un país(*agente*) en cada máquina y por lo tanto que este esquema tenga la capacidad de mover un agente a otra máquina de ser necesario

3. Diseño del sistema

El sistema fue implementado con Java usando *SocketServer* y *Sockets* el cual permitió la comunicación entre los brokers y agentes y que estos realizaran sus funciones principales.

3.1. Diagrama de clases

El diagrama de clases incluido en el “anexo1_diagrama_clases.pdf” contiene cada una de la clases usadas en el proyecto, las cuales contienen métodos que realizan tareas para llevar a cabo el funcionamiento del sistema.

3.1.1. Pais (Agente)

Clase encargada de contener los datos importantes de un país con los que se implementará un modelo básico de autómatas celulares y que este mismo usará para realizar la simulación. Además estará informando a los otros países de su estado actual y su nueva dirección de host si esté fue asignado a otra máquina (se lleva a cabo con ayuda se la clase *ConnectionP*)

3.1.2. Broker

Clase encargada de realizar el balanceo de cargas, el cual hace que el sistema distribuido no tenga una diferencia muy elevada de cargas entre las diferentes máquinas usadas.

Para realizar el balanceo de cargas (enviar un agente a otro host con la capacidad de procesarlo) este envía un DTO con el agente a transferir y el broker receptor se encarga de inicializar el agente en el host debido y agregarlo a su lista de agentes.

3.1.3. ConnectionB

Clase encargada de controlar la comunicación entre Brokers. Recibe las solicitudes que un Broker(cliente) le envía a otro Broker(Servidor) y lleva a cabo las operaciones correspondientes al tipo de mensaje, controlando el Broker(Servidor) y respondiendo al Broker(Cliente).

Esta clase extiende un hilo que se usa en el momento en que un broker le envia una solicitud a otro y luego de realizar la acción debida se finaliza el mismo.

3.1.4. ConnectionP

Clase encargada de controlar la comunicación entre Paises. Recibe las solicitudes que un País(cliente) le envía a otro País(Servidor) y lleva a cabo las operaciones correspondientes al tipo de mensaje, controlando el País(Servidor) y respondiendo al País(Cliente). Cumple la misma función que *ConnectionB* pero esta es para paises.

Esta clase extiende un hilo que se usa en el momento en que un país le envia una solicitud a otro y luego de realizar la labor correspondiente se finaliza el mismo.

3.1.5. DTOCambioHost

Clase usada como objeto para transportar la nueva dirección IP y el nombre de país que cambió de host. Es usada en la clase país para avisarle a los paises vecinos su nueva dirección de host.

3.1.6. Mensaje

Clase usada como contenedor de mensajes que serán intercambiados entre Brokers y Paises, el cual contiene un tipo de mensaje (identifica qué mensaje se enviará) y un objecto correspondiente al tipo de mensaje.

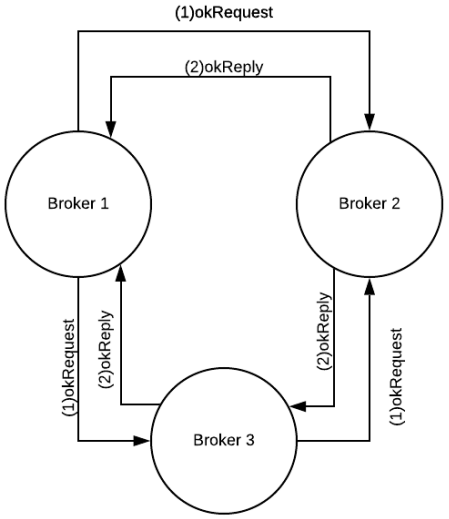
3.1.7. Tipo

Clase especial que tiene declarado los tipos mensajes que se podrán intercambiar entre Brokers y Paises.

3.2. Procesos e intercomunicación

3.2.1. ProtocoloOK

Protocolo que se inicia al comienzo de la ejecución del sistemas en el cual el Broker envía un mensaje a cada vecino leido del archivo de comunicación para asegurarse de que están activos y poder continuar con la ejecución del sistema.



Los **Brokres** intercambian mensajes entre ellos para sincronizarse

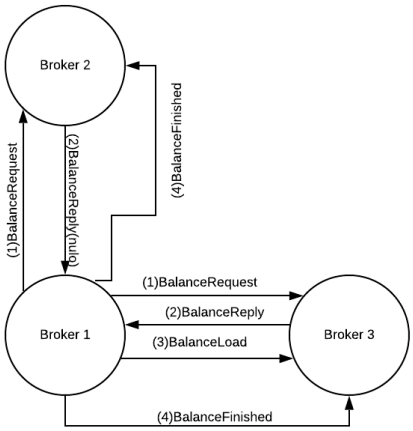
(1)okRequest	Broker1->Broker2	mensaje de comprobacion encendido	Broker2 envía Reply
(2)okReply	Broker2->Broker1	mensaje de respuesta	Broker1 recibe y sincroniza

Fig. 1

3.2.2. Protocolo de balanceo

Protocolo que se encarga de balancear cargas entre los diferentes Brokers (ejecutado cada 17 segundos), buscando entre los Brokers disponibles aquellos que posean una población total igual o menor a la mitad de la población del Broker solicitante.

Este protocolo empieza con un método en los Brokers que le solicita a los Broker vecinos su población total (usa el *protocolo obtener población*) y con ello se revisa si existe alguno que tenga una población significativamente menor al que está ejecutano el balanceo, de ser así, se procede a seleccionar el país para el cambio de host (aquel país que posea mayor carga/población). Luego se procede a ejecutar el intercambio de mensajes (*Balance*) con los Brokers que fueron seleccionados (con menor carga) hasta que se realice el traspaso del país **(Explicado a mayor detalle en la Figura N°2)** además antes de traspasar un país se detiene su simulación y se activa el protocolo *HostChange* para avisarle a los paises vecinos del país que se trasladará, la nueva dirección ip del nuevo host . Para finalizar se envía el país a través de *BalanceLoad()* y elimina del Broker que inició el balanceo



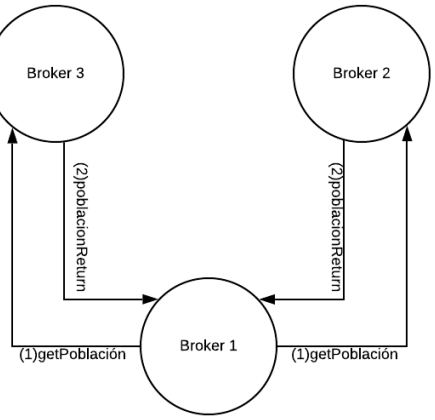
Broker1 realiza la petición de balanceo y envía el país al otro **Broker** que le acepte la peición
Broker1-2 reciben la solicitud y deciden si aceptarla o no

Tipo de Mensaje	Emisor/receptor	Mensaje emisor	Acción receptor
(1)BalanceRequest	Broker 1 -> Broker 2 Broker 1 -> Broker 3	Solicitud Balanceo	Broker 2 recibe solicitud y responde negativamente Broker 3 recibe y responde positivamente
(1)BalanceReply	Broker 2 -> Broker 1 Broker 3 -> Broker 1	respuesta del positiva/negativa	Broker 1 recibe respuestas y envía el país al broker 3 (positivo)
(3)BalanceLoad	Broker 1 -> Broker 3	DTO con el país	Broker 3 recibe el DTO y actualiza sus paises
(3)BalanceFiniched	Broker 1 -> Broker 2 Broker 1 -> Broker 3	mensaje de proeso terminado	Broker 2 termina proceso Broker 3 termina proceso

Fig. 2

3.2.3. Protocolo obtener población

Protocolo que se utiliza para lograr un balanceo de carga continuo y efectivo. En este un broker pide a cada uno de sus vecinos que envíen su población total actual. Cada broker al recibir esta solicitud, debe calcular la suma de la población actual de todos los países bajo su control, y poder responder adecuadamente al getPoblacion.



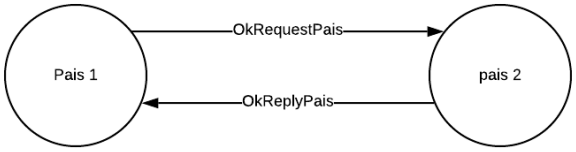
Broker1 solicita la población total a los otros vecinos
Broker2-3 reciben la solicitud y envian su cantidad de población

Tipo de Mensaje	Emisor/receptor	Mensaje emisor	Acción receptor
(1)getPoblacion	Broker 1 -> Broker 2 Broker 1 -> Broker 3	Solicitud de cantidad de poblaión	Broker 2-3 envian reply
(2)OkReplyPais	Broker 2 -> Broker 1 Broker 3 -> Broker 1	Cantidad total de población	Analiza la cantidad de población

Fig. 3

3.2.4. Protocolo de sincronización de paises

Protocolo para sincronizarse con los países vecinos y asegurarse de que están en línea. La ejecución del Protocolo empieza enviando el *OkRequestPais()* a los vecinos aereos y luego a los terrestres y cuando estos respondan con *OkreplyPais()* extrae la información del estado de cada país. La simulación se inicia cuando se completa el protocolo de sincronización de paises



Pais 1 realiza la petición de sincronización a **Pais 2**
Pais 2 envía respuesta a petición a **Pais 1**

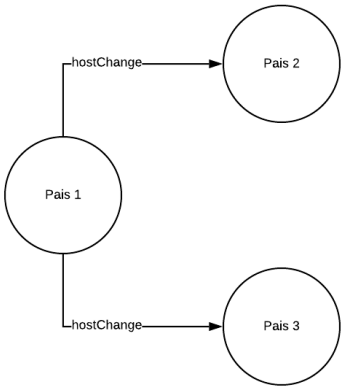
Tipo de Mensaje	Emisor/receptor	Mensaje emisor	Acción receptor
(1)OkRequestPais	Pais 1 -> Pais 2	sincronizar paises	Pais 2 acepta y envía reply
(2)OkReplyPais	Pais 2 -> Pais 1	confirmación de sincronización	Termina sincronización

Fig. 4

3.2.5. Protocolo HostChange

Protocolo llevado a cabo por un país que se inicia cuando el país va a ser cambiado de host, se envía un mensaje a los países vecinos para notificarles que ha cambiado de Host, incluye su nueva dirección IP.

Para informar el cambio, recorre primero los vecinos aereos y luego los terrestres y se envía un mensaje que contiene un mensaje de *hostChange* y un objeto *DTOCambioHost* el cual contiene la ip del nuevo Host y el nombre del país que será cambiado (su propio nombre)



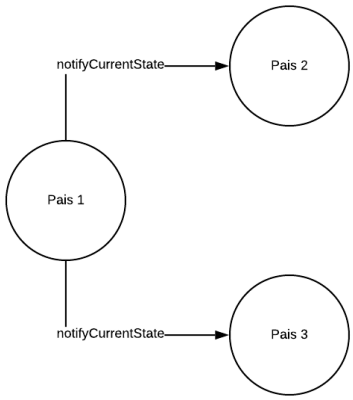
Pais 1 envía su nueva dirección ip a **Pais 2-3**

Tipo de Mensaje	Emisor/receptor	Mensaje emisor	Acción receptor
(1)hostChange	Pais 1 -> Pais 2 Pais 1 -> Pais 3	nueva dirección ip	Pais 2 -3 acepta y modifican la dirección ip del vecino

Fig. 5

3.2.6. Protocolo NotifyState

Mensaje que envía un país para notificarle a sus paises vecinos el estado actual con respecto al virus. Primero almaceno los datos que le pueden interesar a los países vecinos y luego se le envía un mensaje que indica notificación de estado (*notifyCurrentState*) y un objeto tipo *País*, primero a lo vecinos aereos y luego a los terrestres. Estos mensajes son recibidos por la clase ConnectionP. Este protocolo se ejecuta periodicamente cada cierto número de iteraciones para no sobrecargar la red



Pais 1 envía suestado actual a **Pais 2-3**

Tipo de Mensaje	Emisor/receptor	Mensaje emisor	Acción receptor
(1)notifyCurrentState	Pais 1 -> Pais 2 Pais 1 -> Pais 3	Estado actual del país	Pais 2 -3 analiza la información

Fig. 6

4. Modelo de propagación

El sistema implementa un modelo básico de de propagación de un virus, el cual tiene como datos iniciales de cada país:

1. **Cantidad de poblacion**
2. **Porcentaje de aislamiento**
3. **Porcentaje población aislada**
4. **Porcentaje población infectada**
5. **Porcentaje poblacion vulnerable**
6. **Datos de países vecinos**

Luego con estos datos se calculan lo siguiente:

7. **Poblacion en aislamiento** = (Cantidad de poblacion * Porcentaje de aislamiento)
8. **Poblacion infectada** = (Cantidad de poblacion * Porcentaje población infectada)
9. **Población vulnerable** = (Cantidad de población * Porcentaje poblacion vulnerable)
10. **Poblacion infectada vulnerable** = 0
11. **Poblacion sana** = (Cantidad de población * (1 – Porcentaje poblacion infectada) – Población vulnerable)

Adicionalmente existen las siguiente tasas para población interna:

12. **Tasa de mortalidad**
13. **Tasa de mortalidad vulnerables**
14. **Tasa de transmisión**
15. **Tasa de tranmisión de vulnerables**

Y tasas de trasmisión a través de vecinos

16. **Tasa de transmisión aérea**
17. **Tasa de transmision aérea vulnerable**
18. **Tasa de transmisión terrestre**
19. **Tasa de transmisión terrestre vulnerable**

4.1. Funcionamiento del modelo

Con los datos que se posee, se pasa a calcular la nueva cantidad personas muertas:

Nuevos muertos = (Población infectada * Tasa de mortalidad)

Nuevos muertos vulnerables = (Población infectada vulnerable * Tasa de mortalidad vulnerables)

Luego con estas cantidades se actualizan todas las canidades de población

Población vulnerable = Población vulnerable(ant) **-** Nuevos muertos vulnerables

Cantidad de población **=** Poblacion(ant) – nuevos muerto – nuevos muertos vulnerables

Población infectada = Población infectada(ant) – Nuevos muertos

Población infectada vulnerable = Población infectada vulnerable – nuevos muertos vulnerables

Luego se verifica si la población sana (vulnerable y no vulnerable) es mayor que la cantidad de población en aislamiento, de ser así se pueden producir nuevos contagios por lo que habría que calcular nuevas propagaciones:

Nuevos infectados = (población infectada – Tasa de transmisión**) + (**población infectada vulnerable * Tasa de transmisión**)**

Nuevos infectados vulnerables = (Población infectada vulnerable * Tasa de transmisión vulnerables) + (Población infectada * Tasa de transmisión vulnerables)

Y se realiza la misma acción con los datos que los vecinos nos han entregado ya que ese país tiene conacto con todos sus vecinos aéreos o terrestres (La población usada en los siguientes fórmulas son la de los países vecinos, ya que esta es la cantidad de personas que ellos pueden llegar a infectar en nuestro país)

Vecinos aereos:

Nuevos infectados = (población infectada + población infectada vulnerable) * Tasa de transmisión aerea

Nuevos infectados vulnerables = (Población infectada vulnerable + Población infectada) * Tasa de transmisión aerea vulnerables

Vecinos terrestres:

Nuevos infectados = (población infectada + población infectada vulnerable) * Tasa de transmisión terrestre

Nuevos infectados vulnerables = (Población infectada vulnerable + Población infectada) * Tasa de transmisión terrestre vulnerables

Además si la nueva cantidad de población sana (vulnerables y no vulnerables) es menor que la poblacion en asilamiento se producirán nuevas infecciones

Nuevos infectados vulnerables = (Población sana vulnerable + poblacion sana) – Población en aislamiento

Nuevos infetados = (población sana vulnerable + poblacion sana**) –** Población en aislamiento

Y se actualizan los datos de la poblacion

Población sana vulnerable = Población sana vulnerable – Nuevos infectados vulnerables

Población infectada vulnerable = Poblacion infectada vulnerable + nuevo infectados vulnerables

Especificaciones del modelo: Si el porcentaje de infectados iniciales es muy bajo, estos puede morir muy pronto

Cuando la poblacion actual sea igual a la aislada el la simulación del modelo de detiene porque no se generarán nuevas infecciones. Se considera terminada la simulación.

5. Archivos usados

En los archivos, los datos relacionados a cada título descrito a continuación van debajo del mismo. Para entenderlo mejor se recomienda ver los archivos de prueba en ArchivosBroker1.zip y en ArchivosBroker2.zip.

5.1. Broker "*brokerFile.txt*"

Archivo de configuración del Broker, archivo de texto en el cual se incluyen los datos para iniciar cada Broker

- puertoB: puerto a través del cual de van a comunicar los brokers (debe ser el mismo en todos los brokers)
- vecinosB: direcciones IP de lo Brokers vecinos
- Paises: nombres de los archivos de configuración de los paises a los cuales ese Broker administrará actualmente

5.2. País "*paisFile.txt*"

Archivo de configuración para un país (agente). El nombre del archivo cambia dependiendo de cada país, por ejemplo, para colombia sería "*colombiaFile.txt*"

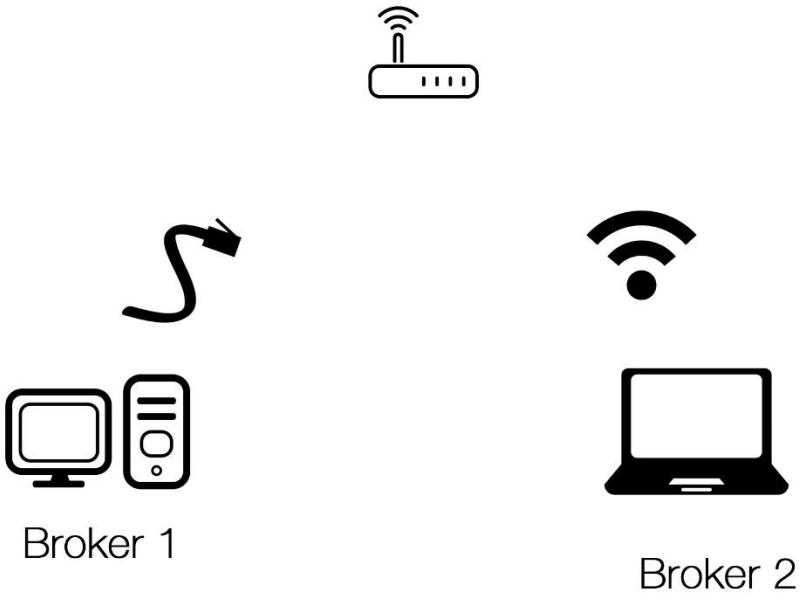
- nombre: indica el nombre del país
- poblaión: Cantidad de la población del país
- porcentajeaislamiento: porcentaje de población aislada del país
- porcentajepoblacioninfectada: porcentaje de población infectada del país
- porcentajepoblacionvulnerable: porccenaje de poblacion vulnerable del país
- puertopaises: Socket por el cual se comunicará con los paises vecinos
- vecinosaereo: vecinos aereos del país, presentados de la forma "nombre del País;IP;número de puerto"
- vecinosterrestres: vecinos terrestres del país, presentados de la forma "nombre del País;IP;número de puerto"

Además se debe estar seguro que las direcciones IP de los países y los Puertos de los sockets sean consistetes ya que de lo contrario nunca se llegará a sincronizar, por ende no se podrá ejecutar el sistema

6. Escenario de pruebas

El escenario en el que se probó el sitema consistió en dos computadoras conectadas a una red local de hogar, se extrajo sus direcciones ip y se agregó estas a los archivos de configuración de los Brokers y Paises

Cada computadora ejecutaba un Broker y cierta cantidad de paises inicialmente como se muestra en la siguiente figura



Broker 1 administraba a:

Colombia, Chile, Venezuela

Sitemas operativo: Windows 10

Broker 2 adminisraba a:

Brasil

Sistemas operativos: Arch Linux

7.GUI

La interfaz que se ejecuta en cada computador, muestra información sobre los países que se están simulando al control de cada Broker. Además muestra su población actual, la cantidad de personas aisladas, los sanos comunes, las personas vulnerables sanas, los infectados comunes y los infectados vulnerables de cada país. Además existe una consola que muestra el estado y funcionamiento actual del progama.