

# Trabajo práctico especial

## Objetivo

Construir una simulación distribuida basada en agentes, con capacidad de **crecer dinámicamente** en computadoras utilizadas, y observar un comportamiento **tolerante a fallas**.

## Enunciado

Junto al enunciado, el alumno recibirá la implementación no distribuida de una simulación basada en agentes.

La simulación esta estructurada a través de un motor simple que permite registrar agentes en la simulación. Cada agente no es más que un programa que es ejecutado en su propio thread, que contribuye de manera independiente al fenómeno estudiado. El ciclo de vida de cada agente es controlado por el motor. Los agentes son independientes entre sí, aunque pueden enviar eventos al resto de los agentes, y obtener objetos “de entorno”, que proveen información de contexto para todos los agentes.

Como caso de prueba, se presenta una simulación simple de cadenas de productores / consumidores, **basada en tres tipos de agentes** y un entorno común de intercambio de recursos (el mercado).

Se pide construir implementaciones del motor de simulación y del entorno de los agentes, de tal manera que pueda coordinarse una simulación con agentes distribuidos en varias computadoras.

Los requisitos de la nueva simulación distribuida son:

Deben poder **agregarse y quitarse** agentes mientras la simulación esta corriendo. **La cantidad de agentes por nodo deberá mantenerse lo más pareja posible**. Es decir que la heurística utilizada para definir la afinidad de un agente a un nodo debe tender a que en estado permanente la diferencia de agentes entre todos los nodos no exceda la unidad. Los agentes deberan poder ser agregados y quitados desde cualquier nodo, sin importar donde esten corriendo.

Deben poder **agregarse nodos mientras la simulación está funcionando**. La aparición de un nuevo nodo implicará la redistribución de agentes existentes para **mantener balanceada la cantidad de agentes que se ejecutan por nodo**.

Deben poder **quitarse nodos mientras la simulación esta funcionando**. Los agentes que se encontraban corriendo en dicho nodo deberán ser migrados a otros nodos manteniendo el principio de **ecualizar la cantidad de agentes por nodo**. Si el nodo que se quita es el único presente, se finalizará la simulación.

**La simulación deberá tolerar la falla de hasta un nodo**. Al detectar la falla de un nodo, se deberá excluir a dicho nodo de la simulación. Los agentes que corrían en dicho nodo se darán por perdidos, y cualquier oferta no concretada a su nombre deberá ser eliminada del mercado. La falla de **cualquier** nodo no deberá causar la pérdida de ninguna información de mercado.

Deberá poder **consultarse información de la simulación o del mercado en su totalidad**, desde cualquier nodo del sistema.

Como requisito adicional, **NO se podrán modificar las interfases provistas** para la simulación y el mercado, así como tampoco la implementación de los agentes provistos.

## **Diseño del sistema**

Dentro del código fuente entregado por la cátedra, se destacan las siguientes interfases:

### **ar.edu.itba.pod.simul.simulation.SimulationManager**

Representa el punto de partida para gestionar una simulacion. Contiene metodos que permiten registrar y quitar agentes de la simulacion, comenzarla o finalizarla, registrar objetos de entorno que pueden ser consultados por los agentes, y obtener las interfases **Simulation** y **SimulationInfo**, que permiten a los agentes interactuar y obtener informacion global de la simulacion respectivamente.

Las implementaciones locales de dichas interfases pueden encontrarse en el paquete ar.edu.itba.pod.simul.simulation.local.

### **ar.edu.itba.pod.simul.market.MarketManager**

Representa el punto de partida para gestionar un mercado. Contiene metodos para inicializar y para finalizar un mercado, asi como acceder a las interfases operativa (Market), y de informacion global (MarketInfo).

Las implementaciones locales de dichas interfases pueden encontrarse en el paquete ar.edu.itba.pod.simul.simulation.local.

### **ar.edu.itba.pod.simul.comunicacion.\***

Interfaces que encapsulan la funcionalidad requerida para la implementacion distribuída de la simulación. Cada interfaz se encuentra debidamente documentada en cuanto a su propósito, las acciones realizadas por cada método, los parámetros de entrada, los parámetros de salida y las validaciones a realizar.

El alumno deberá respetar estrictamente cada interfaz debido a que parte de la evaluación se realizará ejecutando casos de prueba contra las mismas.

Todas las implementaciones del alumno deben ubicarse en el siguiente paquete:

### **ar.edu.itba.pod.XXX**

Donde XXX es el número de legajo del alumno. Cualquier clase, interfaz o archivo fuera del paquete será ignorado.

La implementación del alumno **debe instanciar el RMI Registry de manera embebida**. Es decir, no debe hacer falta levantar el mismo desde afuera de la aplicación.

## Modalidad y Calificación

El trabajo será desarrollado individualmente. Cualquier **plagio** detectado será penalizado con una calificación de 0 - no recuperable – reservándose la cátedra el derecho de tomar medidas administrativas adicionales.

La fecha límite de entrega será **22/11/2010 a las 19:00 horas**. Quienen no entreguen en dicha fecha dispondrán de una fecha tardía, que vence el **29/11/2010 a las 19:00 horas**, obteniendo una **penalización de 1 punto** en la calificación final obtenida por entregar tarde.

Quienes entreguen el trabajo en la primera fecha, y obtengan una calificación que implique desaprobar el trabajo, tendrán una oportunidad de mejorarlo y volver a entregarlo. La corrección recibirá una **penalización de 1.5 puntos**.

La entrega será calificada con una nota entre 0 y 10. **La nota mínima para considerar aprobado el trabajo práctico será de 7 puntos**.

## Entregables

Se deberá entregar los siguientes ítems:

Código fuente completo del trabajo práctico. El mismo debe ser enviado por mail a todos los docentes de la cátedra.

Manual para crear el jar ejecutable. El mismo puede ser utilizando **Maven** o **Ant**. En ambos casos, la ejecución **no debe contener error** (classpath seteados a mano dependientes de la ubicación, etc). Toda la ejecución de prueba se va a realizar por línea de comando. **No asumir la existencia de algun IDE** (Eclipse o similares).

Informe impreso.

La entrega se considera realizada cuando estén todos los ítems. **No se acepta la entrega parcial** de los mismos (por ejemplo, entregar el informe en otro momento).

## Informe

El informe debe contener el analisis de las siguientes situaciones:

Carga en un nodo

Construir una simulacion de una cadena de produccion con al menos 5 tipos de recursos. Simular a una velocidad tal que se produzcan aprox. 1 transacciones por segundo real utilizando 5 agentes.

Simular utilizando un solo nodo, incrementando la cantidad de agentes en 1 cada 10 segundos, midiendo la tasa de transacciones. Determinar el valor máximo hasta donde puede crecer el tamaño de la simulación (transacciones por segundo simuladas) y explicar que recurso está causando el problema.

#### Carga distribuída

Repetir la simulación anterior, en la situación límite de carga, utilizando dos nodos. Luego tres y finalmente cuatro nodos. Aumentar la cantidad de agentes hasta llegar al punto donde el valor máximo de transacciones por segundo simuladas llegue a su límite.

Analizar el límite, determinar las razones del mismo, y compararlo con el límite utilizando un solo nodo.

### **Distribución de la nota / criterios de calificación**

A la hora de calificar el trabajo práctico se tendrán en cuestión los siguientes aspectos:

¿La simulación funciona correctamente en un solo nodo? (1 punto)

¿La simulación funciona correctamente en varios nodos? (1 punto)

¿Se pueden agregar nodos cuando la simulación corre? (1 punto)

¿El informe está completo/correcto? (1 punto)

¿Se pueden quitar nodos cuando la simulación corre? (2 puntos)

¿Sobrevive la simulación a la falla de un nodo? (2 puntos)

¿Pueden los nodos interactuar con los nodos de referencia implementados por la cátedra? (2 puntos)

**La cátedra se reserva el derecho de otorgar puntos adicionales**, así como también de otorgar puntuación parcial de acuerdo a su entendimiento de cómo se han cumplido los requisitos de la simulación en cada caso.