
PROGRAMACIÓN AVANZADA

Práctica de Laboratorio (PECL)

Convocatoria Ordinaria – Mayo 2023

Publicado el 29 de marzo de 2023, versión 2.0.

Simulación del funcionamiento de una colonia de hormigas

Parte 1: Programación Concurrente

Se desea modelar el comportamiento de una colonia de hormigas que tiene las siguientes características:

- La colonia de hormigas cuenta con tres túneles, uno para entrar y dos para salir. Por cada uno de estos túneles solo puede circular una única hormiga a la vez y el tiempo requerido para cruzar los túneles es 0,1 segundos.
- Para evitar aglomeraciones en la colonia, las hormigas deberán ser creadas (instanciadas) de forma escalonada (es decir, deben ser creadas paulatinamente) en intervalos aleatorios de entre 0,8 y 3,5 segundos. Las hormigas nacerán en el exterior de la colonia y accederán a ésta a través del túnel de entrada ya explicado.
- La colonia cuenta con 3 tipos de hormigas: soldado, obreras y crías, y cada una de ellas tiene un comportamiento diferente.
- La colonia cuenta con cuatro zonas diferenciadas: ALMACÉN DE COMIDA (acceden únicamente las obreras), ZONA DE INSTRUCCIÓN (acceden únicamente las soldado), REFUGIO (acceden solamente las crías), ZONA PARA COMER (pueden acceder todos los tipos de hormiga) y ZONA DE DESCANSO (pueden acceder todos los tipos de hormiga).
- El sistema genera 10.000 hormigas en total: 6.000 obreras, 2.000 soldados y 2.000 crías.
 - Cada 3 hormigas obreras creadas, se creará 1 hormiga soldado y 1 hormiga cría.

Hormigas obreras

Las hormigas obreras deben ser modeladas como hilos y se identificarán como “HOXXXX”, donde X es un número (id) único como, por ejemplo, HO0001, HO0023, HO1234, etc.

Cada hormiga obrera cuyo identificador sea **impar** (sólo la parte numérica) **repetirá iterativamente** el siguiente comportamiento una vez se ha unido a la colonia:

- Acude a la zona exterior de la colonia a coger cinco elementos de comida y lo lleva al interior de la colonia, tardando 4 segundos. A continuación, lo deposita en el ALMACÉN DE COMIDA, tardando un tiempo aleatorio de entre 2 y 4 segundos. A este ALMACÉN DE COMIDA sólo pueden acceder simultáneamente 10 hormigas.

Cada hormiga obrera, con un identificador **par**, se encargará de ir sirviendo comida entre el ALMACÉN DE COMIDA y la ZONA PARA COMER. Así, una vez se haya unido a la colonia, **repetirá iterativamente** el siguiente comportamiento:

- Accede al ALMACÉN DE COMIDA para coger cinco elementos de comida requiriendo entre 1 y 2 segundos. Entonces, viaja hacia la ZONA PARA COMER, tardando un tiempo aleatorio de entre 1 y 3 segundos. A continuación, accede a la ZONA PARA COMER y deposita el elemento de comida, consumiendo entre 1 y 2 segundos. Desde este momento, elemento de comida está disponible para ser ingerido por cualquier hormiga de la colonia.

Otras consideraciones:

- Todas las hormigas obreras, pares o impares, tras realizar 10 iteraciones completas de su acción principal, pasan por la ZONA PARA COMER para reponer fuerzas y comer, consumiendo 1 unidad de alimento y tardando 3 segundos. A continuación, acceden a la ZONA DE DESCANSO, consumiendo allí 1 segundo, para, a continuación, volver a retomar su actividad habitual.

Hormigas soldado

Las hormigas soldado deben ser modeladas como hilos y se identificarán como “HSXXXX”, donde X es un número (id) único como, por ejemplo, HS0001, HS0023, HS1234, etc. Cada hormiga soldado, una vez se ha unido a la colonia, **repetirá iterativamente** el siguiente comportamiento:

- Hace instrucción en la ZONA DE INSTRUCCIÓN, tardando un tiempo aleatorio de entre 2 y 8 segundos en hacer esta operación. A continuación, descansa 2 segundos en la ZONA DE DESCANSO.

Otras consideraciones:

- Todas las hormigas soldado, tras realizar 6 iteraciones completas de su acción principal, pasan por la ZONA PARA COMER a reponer fuerzas y comer, consumiendo 1 unidad de alimento. Esta operación tarda 3 segundos en realizarse.
- Las hormigas soldado deben estar atentas, porque, en caso de que un insecto invasor amenace la colonia, deben dejar lo que estén haciendo en ese momento y salir inmediatamente de la colonia a repeler al insecto invasor. Cuando todas las hormigas soldado de la colonia se encuentren en el exterior (únicamente aquellas que estaban creadas de forma previa a la generación de la amenaza. Las hormigas soldado creadas durante el intento de invasión realizarán la instrucción normalmente), estarán listas para repeler al insecto. A partir de dicho momento se producirá una lucha entre las hormigas y el insecto, consumiendo 20 segundos. El evento finaliza con la huida del insecto sin causar bajas de ninguna hormiga soldado. A continuación, continúan con el comportamiento previamente definido (ya sea ir a la ZONA DE INSTRUCCIÓN o la ZONA PARA COMER), es decir, no se altera el contador de iteraciones al realizar esta acción.

- Las hormigas obreras no alterarán su comportamiento con la aparición de un insecto invasor.

Hormigas crías

Las hormigas crías deben ser modeladas como hilos y se identificarán como “HCXXXX”, donde X es un número (id) único como, por ejemplo, HC0001, HC0023, HC1234, etc. Cada hormiga cría **repetirá iterativamente** el siguiente comportamiento una vez se ha unido a la colonia:

- Accede a la ZONA PARA COMER, tardando un tiempo aleatorio de entre 3 y 5 segundos en alimentarse. A continuación, descansan 4 segundos en la ZONA DE DESCANSO.

Otras consideraciones:

- Dado que las hormigas cría son muy débiles, en caso de que exista la amenaza de un insecto invasor, todas ellas (tanto las creadas de forma previa a la amenaza como las generadas durante la amenaza) deben dejar lo que estén haciendo en ese momento y acudir rápidamente a la zona de REFUGIO, hasta que la amenaza haya desaparecido, volviendo entonces a su comportamiento habitual (iniciando su alimentación en la ZONA PARA COMER).

Otras consideraciones adicionales

- Todo el comportamiento del sistema se mostrará gráficamente por pantalla mediante una interfaz de usuario de Java. Además, se deberá incluir un botón para pausar/reanudar el sistema, de forma que sea sencillo hacer el seguimiento de la ejecución del programa. Un posible ejemplo de interfaz del sistema se muestra más abajo.
- Se debe implementar un botón en la interfaz de usuario que genere un insecto invasor que amenace a la colonia, de tal manera que se genere una amenaza de insecto invasor, en el que las hormigas soldado deberán salir de la colonia a repeler la amenaza. Esta amenaza de insecto invasor durará hasta que sea repelido por las hormigas soldado. Una vez una amenaza esté activa no será posible generar una nueva hasta que ésta haya finalizado.
- El funcionamiento de todas las hormigas no finaliza nunca.
- El comportamiento histórico del sistema se guardará en un fichero **log** (un fichero de texto llamado “evolucionColonia.txt”), de forma que sea sencillo analizar lo sucedido. El log guardará todos los eventos que van teniendo lugar, por ejemplo: “La hormiga obrera HO0002 sale a buscar comida”, “La hormiga soldado HS0010 comienza a hacer instrucción”, etc. En cada línea de dicho log deberá constar la marca de tiempo (**fecha y hora**, incluyendo el segundo determinado en el que tuvo lugar el evento) **y el evento** en sí. El sistema de log deberá implementarse como un recurso compartido a utilizar por todo el sistema concurrente y deberá protegerse adecuadamente.

Exterior de la colonia:

Hormigas buscando comida: HO0005, HO0007, HO0009

Hormigas repeliendo un insecto invasor:

Interior de la colonia:

Hormigas en el ALMACÉN DE COMIDA: HO0001, HO0003

Hormigas llevando comida a la ZONA PARA COMER: HO0002, HO0004

Hormigas haciendo INSTRUCCIÓN: HS0001, HS0002, H0004

Hormigas descansando: HO0008

Unidades de Comida (ALMACÉN) 5

Unidades de Comida (ZONA PARA COMER) 2

ZONA PARA COMER HO0006, HC0001, HC0002, HC0003, HS0003

REFUGIO

Pausar Reanudar Generar Amenaza Insecto Invasor

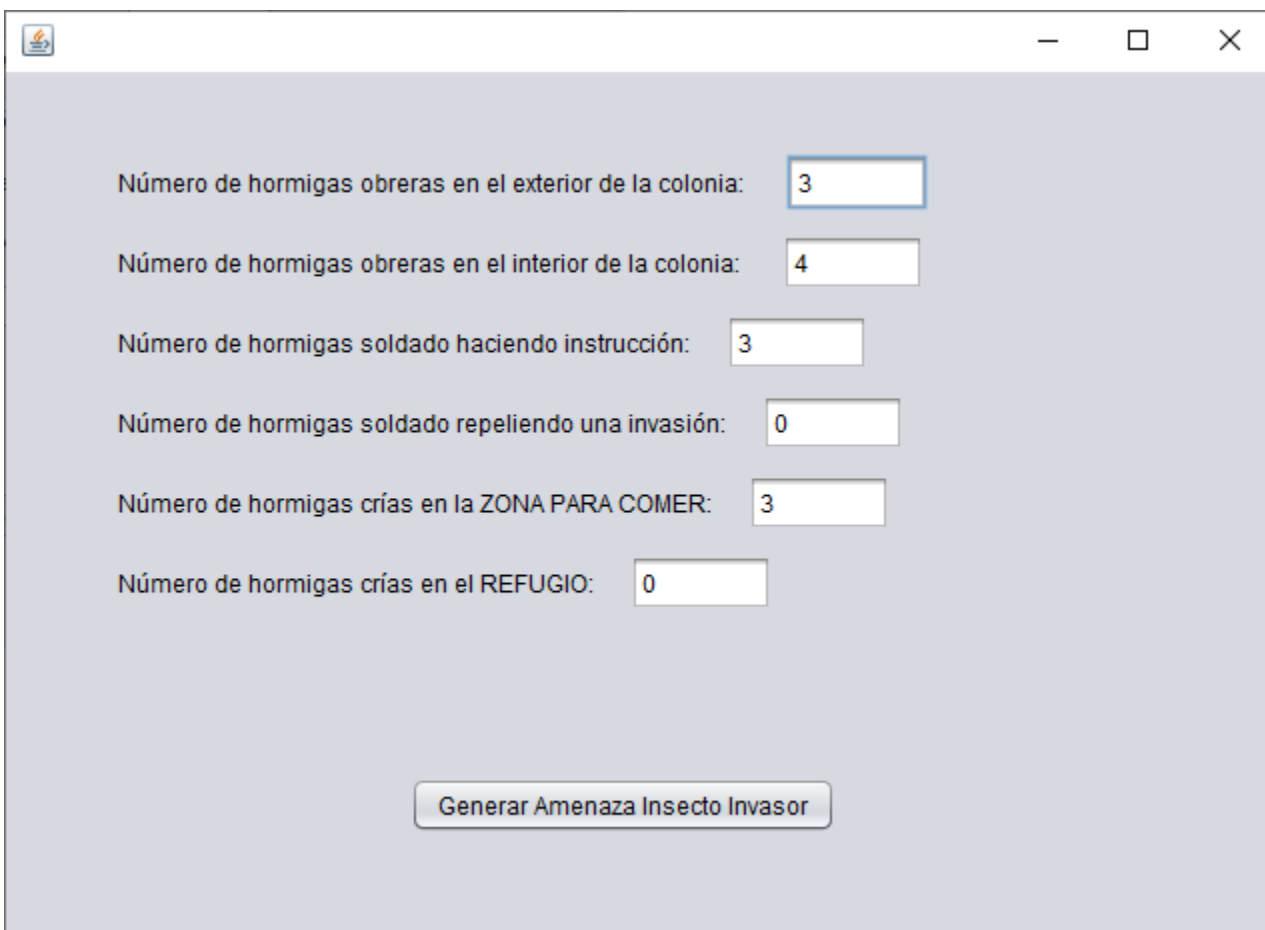
Parte 2: Programación Distribuida

Basándose en la Parte 1 anterior, incluir un nuevo módulo de acceso remoto implementado mediante RMI o Sockets, que ofrecerá la siguiente información y operaciones:

- Número de hormigas obreras en el exterior de la colonia.
- Número de hormigas obreras en el interior de la colonia.
- Número de hormigas soldado haciendo instrucción.
- Número de hormigas soldado repeliendo una invasión.
- Número de hormigas crías en la ZONA PARA COMER.
- Número de hormigas crías en el REFUGIO.
- Iniciar de forma remota, mediante un botón, la generación de un insecto invasor.

La consulta de estos datos debe ser en tiempo real y debe mostrarse de forma automática, es decir, **sin que el usuario deba pulsar ningún botón ni interactuar de forma alguna** para actualizar la información.

Un posible ejemplo de interfaz gráfica es el que se muestra en la siguiente figura.



The image shows a graphical user interface (GUI) window with a light gray background. The window has a standard title bar with a minimize button, a maximize button, and a close button. The main content area contains six text labels followed by input fields, each containing a numerical value. The labels are in Spanish and describe different ant colony parameters. At the bottom center, there is a button labeled "Generar Amenaza Insecto Invasor".

Label	Value
Número de hormigas obreras en el exterior de la colonia:	3
Número de hormigas obreras en el interior de la colonia:	4
Número de hormigas soldado haciendo instrucción:	3
Número de hormigas soldado repeliendo una invasión:	0
Número de hormigas crías en la ZONA PARA COMER:	3
Número de hormigas crías en el REFUGIO:	0

Generar Amenaza Insecto Invasor

Condiciones de entrega

1. Se deben desarrollar, en total, dos programas:
 - a. Un servidor, cuyo código base será el programa desarrollado en la Parte 1, ampliado con la funcionalidad correspondiente para dar soporte al módulo de programación distribuida.
 - b. Un programa cliente que permita realizar las operaciones detalladas en la Parte 2, para consultar el estado del sistema de forma automática y generar la amenaza del insecto invasor de forma remota.
2. Se podrán emplear todos los mecanismos utilizados en clase para resolver todos los problemas de comunicación y sincronización que se plantean en este enunciado. No obstante, se deben utilizar los mecanismos de sincronización y comunicación que resuelvan el problema **de la forma más eficiente y óptima** posible.
3. La práctica se realizará **principalmente por parejas** (también puede llevarse a cabo individualmente) y deberá ser entregada **antes** de la fecha indicada en el Aula Virtual, a través de la tarea correspondiente, mediante la subida de dos archivos: la **memoria** de la práctica en formato PDF y el **proyecto Netbeans** completo, comprimido como **ZIP** (no utilizar extensión .rar).
4. La entrega fuera del plazo indicado en el Aula Virtual supondrá una reducción en la calificación final, siendo del 25% si se entrega el día siguiente a la fecha límite, o del 50% si se entrega dentro de los dos días siguientes. La entrega más allá de esos dos días no será admitida bajo ninguna circunstancia.
5. El proyecto entregado deberá ser un **proyecto de NetBeans**. No se admitirán proyectos realizados con otros entornos de desarrollo.
6. Para aprobar, es condición necesaria que todos los programas funcionen correctamente y de acuerdo a las especificaciones indicadas en los enunciados.
7. Se debe desarrollar la solución haciendo uso de **buenas prácticas de programación**. Por ejemplo, es necesario que todos los nombres de las clases comiencen por una letra mayúscula y todos los nombres de atributos y métodos comiencen por una letra minúscula; los atributos deberán ser privados, y sólo se podrá acceder a ellos mediante métodos getter y setter.
8. **Ambas partes** (concurrente y distribuida) de la práctica de laboratorio **se deberán entregar juntas** (es decir, en un único proyecto y una única memoria), ya que la parte de programación distribuida se construye sobre la parte concurrente.
9. Si la práctica es realizada por una pareja, **sólo uno de los integrantes deberá subirla** al aula virtual, indicando el nombre de ambos alumnos en la portada de la memoria.
10. En la portada de la memoria deberán figurar los datos siguientes:
 - a. **Grado en Ingeniería [Informática / de Computadores]**
 - b. **Curso 20xx/20xx – Convocatoria Ordinaria**

c. DNI – Apellidos, Nombre

- 11. La memoria deberá incluir, como anexo, el código fuente del programa. Si esto no fuera así, la práctica no podrá ser aprobada.**
- 12. La memoria explicativa de la práctica realizada deberá incluir, en el orden siguiente:** 1) un análisis de alto nivel (descripción general del problema e identificación de sus principales actores); 2) diseño general del sistema y discusión de las herramientas de sincronización utilizadas; 3) descripción de las clases principales (atributos y métodos); 4) diagrama de clases; y 5) el código fuente, como anexo.
- 13. Dicha documentación, exceptuando el código, no deberá extenderse más de 20 páginas. La calidad de la documentación** – presentación, estructura, contenido, redacción – **será un elemento básico en la evaluación de la práctica.**
- 14. De cara a la defensa/examen de la práctica, el estudiante podrá llevar impresa una copia de la memoria entregada con la práctica.**
- 15. La resolución de la práctica debe ser genuina y realizada desde cero, es decir, no se podrá utilizar ningún tipo de código fuente de prácticas presentadas a esta convocatoria o en convocatorias anteriores, bien sean propias o de otros compañeros. En caso de detectarse esta situación, la práctica no será evaluada y tendrá una calificación de 0 Suspenso. Además, se ejecutarán los procedimientos correspondientes para comunicar dicho comportamiento fraudulento, con consecuencias académicas acorde a la legislación vigente.**