
PROGRAMACIÓN CONCURRENTE Y DISTRIBUIDA

Práctica de Laboratorio (PECL)

Convocatoria Ordinaria – Mayo 2025

Publicado el 31 de marzo de 2025, versión 1.0.

Simulación de un apocalipsis zombi

Parte 1: Programación Concurrente

En el año 2025 se ha desencadenado un apocalipsis zombi, y hay un área que se ha establecido como un refugio seguro para los humanos. En la nueva normalidad que acompaña esta situación, se deben tener en cuenta las siguientes características generales:

- Existen tres zonas diferenciadas: una zona de refugio, en la que los humanos no corren peligro; una zona de riesgo, en la que los zombis merodean en busca de alimento; y unos túneles que conectan ambas zonas, en los que los humanos no corren peligro.
- La zona de riesgo cuenta con 4 áreas inseguras diferentes.
- Hay 4 TÚNELES para entrar y salir del refugio y que conectan con cada una de las áreas inseguras. Por cada uno de estos túneles solo puede circular un único humano a la vez. En caso de existir humanos en ambos lados del túnel, siempre tendrá prioridad el que intente acceder al refugio por seguridad (es decir, prioridad para entrar al refugio para estar a salvo).
- Para evitar aglomeraciones, los humanos deberán ser creados (instanciados) de forma escalonada (es decir, deben ser creados paulatinamente) en intervalos aleatorios de entre 0,5 y 2 segundos. Los humanos iniciarán su vida en el interior del refugio.
- El refugio cuenta con 3 zonas diferenciadas: ZONA DE DESCANSO, COMEDOR, y ZONA COMÚN.
- Además de los humanos, el sistema modelará el comportamiento de los zombis.
- El sistema generará 10.000 humanos y 1 zombi (el paciente cero).

ZOMBIS

Todos los zombies del sistema deben ser modelados obligatoriamente como hilos y se identificarán como “ZXXXX”, donde X es un número (id) único como, por ejemplo, Z0001, Z0023, Z1234, etc. El único zombi generado inicialmente (el paciente cero), tendrá como id “Z0000”.

Los zombies deambularán entre las distintas áreas inseguras en busca de comida, pasando de una a otra de forma aleatoria. Una vez acceden a una zona, comprobarán si hay algún humano a quien atacar. Si existen varios humanos, se seleccionará un objetivo aleatoriamente e intentará atacarle. En caso de que el ataque consiga herir al objetivo, se incrementará su contador de muertes. El ataque durará un tiempo aleatorio de entre 0,5 y 1,5 segundos. Si no encuentra a nadie a quien atacar, o si ya ha atacado a alguien en esa área insegura, permanecerá durante un tiempo aleatorio entre 2 y 3 segundos en esa área antes de cambiar a otra área.

Este comportamiento se **repite indefinidamente**.

HUMANOS

Los humanos deben ser modelados obligatoriamente como hilos y se identificarán como “HXXXX”, donde X es un número (id) único como, por ejemplo, H0001, H0023, H1234, etc.

Por seguridad, todos los humanos iniciarán su actividad en el refugio.

En el refugio, todos los humanos ayudarán en todas las tareas por igual, **repitiendo iterativamente** el siguiente comportamiento, una vez se ha unido al refugio:

- Accederán a la ZONA COMÚN, donde se prepararán, junto a sus compañeros, para adentrarse en el mundo exterior. Permanecerán en esa zona un tiempo aleatorio entre 1 y 2 segundos.
- Desde la ZONA COMÚN, seleccionarán un TÚNEL para salir al exterior (a la zona de riesgo) y esperarán hasta formar un grupo de 3 humanos que quieran acceder a ese TÚNEL concreto. Una vez formado el grupo, accederán al TÚNEL seleccionado y lo atravesarán (de uno en uno) accediendo a la zona exterior en búsqueda de comida. El tiempo estimado para atravesar el túnel es de 1 segundo. El grupo de paso es exclusivamente de 3 humanos (aunque luego se atravesase de uno en uno); si llegara simultáneamente un 4º humano, éste deberá esperar hasta formar otro grupo de 3 humanos.
- En la zona exterior estarán un tiempo aleatorio entre 3 y 5 segundos y cada uno recolectará 2 piezas de comida. Cuando un humano recoge la comida, y en el caso de que no sea atacado por un zombi, regresará al túnel para acceder al refugio sin esperar a sus compañeros.
- Si un humano es atacado por un zombi, intentará defenderse durante todo el ataque. La duración del ataque la marcará el zombi, por lo que el humano no podrá reanudar su marcha hasta que el ataque no finalice. Las probabilidades que el humano se defienda del ataque recibido son de 2/3. Si la defensa es exitosa, el humano quedará marcado y regresará inmediatamente al túnel al finalizar el ataque, sin recolectar la comida. En caso de no tener éxito en la defensa, el humano morirá y renacerá como zombi en la misma área donde fue atacado.

- El zombi que renace mantiene el mismo id numérico que tenía como humano. Por ejemplo, si muere el humano “H0001”, renace como “Z0001”.
- Una vez de vuelta en el refugio, depositarán automáticamente la comida recolectada y acudirán directamente a la ZONA DE DESCANSO, donde estarán un tiempo aleatorio entre 2 y 4 segundos.
- Tras el descanso, accede al COMEDOR. Intentará tomar una pieza de comida y se la comerá, tardando un tiempo aleatorio entre 3 y 5 segundos. Si no hay comida disponible, esperará de forma ordenada a que alguno de los exploradores retorne con comida.
- Después de comer, y en el caso de haber sido marcado por un zombi, necesitará reponerse de sus heridas, por lo que volverá a la ZONA DE DESCANSO para poder recuperarse del todo, tardando un tiempo de entre 3 y 5 segundos en recuperarse del todo. Al salir, no necesita volver a comer.
- Finalizadas estas actividades, regresa a la ZONA COMÚN para repetir el ciclo.

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.
- El funcionamiento de todos los humanos y los zombis no finaliza nunca.
- Todo el comportamiento del sistema se guardará en un fichero **log** (un fichero de texto llamado “apocalipsis.txt”), de forma que sea sencillo analizar lo sucedido. El log guardará los eventos que van teniendo lugar, por ejemplo: “El zombi Z0002 ataca al humano H1234 (número de muertes: 2)”, “El humano H4567 ha muerto”, “El humano H0001 accede a la zona de descanso”, 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.

Apocalipsis Zombi		
REFUGIO	TÚNELES	ZONA DE RIESGO
Descanso H5469, H8975, H5728, H5555	H1245 H8756 H0001 H19945	H1115 H3356 Z0001
Comedor H1745, H4825 Comida 53	H3322 H0002 H12345 H33221 H87456	H2345 H3221 H7456
Zona común H5826, H2233	H8856 H0081 H65345 H3333	H1456
	H1265 H3451 H3456 H0781 H1999	H1235 H2221 H8556

Nota importante: los IDs y las cantidades indicadas en la imagen no tienen por qué ser correctas, ni estar completas. Las imágenes se muestran con el único fin de proporcionar una idea sobre la interfaz final y la información que debe contener.

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 humanos dentro del refugio.
- Número de humanos en cada uno de los túneles.
- Número de humanos en cada área insegura.
- Número de zombis en cada área insegura.
- Ranking con los 3 zombis más letales y su número de muertes.
- Botón para detener/reanudar la ejecución del programa principal.

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:

Apocalipsis Zombi - Remoto

Número de humanos en el refugio

Número de humanos en los túneles

Número de humanos las zonas de riesgo

Número de zombis en las zonas de riesgo

Zombis más letales

Z0001 - 5 muertes

Z0002 - 2 muertes

Z0003 - 3 muertes

Detener ejecución

Condiciones de entrega

- Se deben desarrollar, en total, dos programas:
 - 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.
 - Un programa cliente que permita realizar las operaciones detalladas en la Parte 2, para consultar el estado del sistema de forma automática y poder pausar/reanudar de forma remota el programa.
- 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.
- La práctica se realizará **opcionalmente 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 o IntelliJ** completo, comprimido como **ZIP** (no utilizar extensión .rar).
- La entrega fuera del plazo indicado en el Aula Virtual supondrá una reducción en la calificación final, siendo del 25% si se entrega dentro de las 24 horas siguientes 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.
- El proyecto entregado deberá ser un **proyecto de NetBeans o IntelliJ**. 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 Matemáticas y Computación**
 - 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. Puede ser la memoria completa, o simplemente el código fuente. Para la defensa, no se permite el uso de dispositivos electrónicos (tampoco para consultar la memoria). Durante la defensa/examen no se puede compartir la memoria entre compañeros.
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 normativa vigente.