



## Bienvenidos al mundo real...

Pues la historia comienza con un secuestro. Éramos programadores en la nave *Nebuchadnezzar*, tan buenos, tan buenos, que las máquinas nos han secuestrado. Necesitan que un programador humano les haga una simulación ya que sólo un humano es capaz de captar pequeñas sutilezas que las máquinas son incapaces de percibir.

Los cacharros estos están algo preocupadillos por la evolución del virus **Smith**, la influencia de **Neo** en el sistema y la

tendencia que tienen el resto de los personajes a morir inesperadamente.

Nos cuentan que, en **Matrix**, los personajes tienen un *id*, *nombre*, una *localización* (compuesta de *latitud* y *longitud* de localización principal y *nombre de la ciudad de nacimiento*) y una *edad*. También tendrán una *fecha de creación en el sistema*. Sin embargo, tendremos al agente **Smith** que además tendrá un *número - porcentaje* que indicará su *capacidad de infectar*, **Neo** que tendrá un atributo para indicar si se cree que es el *elegido* o no y los **personajes genéricos** que tendrán, además, un *número - porcentaje* con la *probabilidad de morir* que tienen.

Obligatoriamente todos los personajes deben implementar los métodos *generar*, *pedir* y *mostrar* (*pedir* y *mostrar* como siempre; *generar* ya veremos lo que hace), sin embargo, debemos permitir la creación de personajes genéricos. También debemos llevar la cuenta del *total de personajes* creados en Matrix desde el inicio de la simulación.

Inicialmente generaremos 200 personajes y se agrupan (ni **Neo** ni **Smith** estarán en esta estructura). Debemos tener en cuenta que el primero en generarse será el primero en ir a Matrix. Una vez generado, lo usaremos en el programa principal rellenándola con 200 personajes. Para evitar que pregunte los nombres de los personajes y de las ciudades para cada personaje, se eligen al azar de una lista de los siguientes: *Pepe, Juan, Ana, Sonia, Pedro, Chiquito y Elena*, para los nombres de los personajes y *Nueva York, Pekín, Roma, París, Londres y Caracuel*, para los nombres de las ciudades.

La simulación de Matrix que nos piden debe ser una **matriz de 5x5 (por defecto, aunque debe ser parametrizable y pedida desde el método principal al comenzar)**, que ellos observarán. En el tablero habrá, inicialmente, un **Neo**, un **Smith** y el resto serán personajes normales, que se irán sacando de la estructura generada. Todos se colocarán Matrix y se actualizarán sus valores de coordenadas. Una vez rellenada la matriz inicial, comienza la simulación que **parará a los 5 minutos o cuando nos quedemos sin personajes vivos, es decir, todos hayan sido infectados por Smith**. El ecosistema se basa en lo siguiente:

- Cada segundo se evalúa el porcentaje de morir de cada personaje de la matriz, si es inferior a 30% el personaje casca y es sustituido por otro; en caso contrario se decrementa dicha probabilidad en un 10%.
- Cada dos segundos el agente **Smith** entra en acción. Genera aleatoriamente una capacidad de infectar, que irá entre 1 y su capacidad máxima de infectar (Debes pensar cuanto sería). Según ese número infectará (convertirá en agentes Smith) a esa cantidad de personajes adyacentes excepto: cuando todos los que le rodean son ya Smith que no hace nada y si está **Neo** no lo infecta.

- **Neo actúa cada 5 segundos.** Decide al 50% si se cree que es el elegido o no (está algo indeciso el hombre y se lo cuestiona), si se lo cree, se carga a todos los agentes *Smith* que tenga alrededor, en caso contrario no hace nada. Tras actuar (o no) siempre lo que hace es cambiar de posición aleatoria: si en la nueva posición hay un agente *Smith*, se lo carga; si hay un personaje, intercambia su posición. **Todos los agentes Smith eliminados se almacenarán en un depósito de virus que siempre está ordenado por fecha de creación ascendente.**
- **Cada 5 segundos imprimimos el estado de Matrix, mostrando las casillas y marcando como o si no está ocupada, si hay un agente Smith, si está Neo y si es un personaje.** No olvidemos que en cada actuación de personaje debemos mostrar un mensaje, para ver lo que pasa y ayudarnos a saber que nuestro programa funciona bien.
- **Cada 10 segundos aparecen 5 personajes en Matrix de nuestro almacén si hay hueco disponible.**
- **Cada 30 segundos podemos introducir un nuevo personaje genérico, pidiéndole los datos por consola.**

Al final de la simulación a parte del estado del sistema: Matrix, posición de Neo, sacaremos la lista de personajes generados por el sistema y ordenados por id, y la lista de virus Smith ordenada por fecha de manera descendente.

Las máquinas son muy organizadas y nos sugieren amablemente, retorciéndonos un poco un brazo, que todas las funciones auxiliares cómo podrían ser leer un entero, leer un real o generar un número al azar estén organizadas en una **clase auxiliar** (yo les haría caso).

Además, también nos piden, de la misma forma amable y simpática, que nuestro programa sea robusto; es decir, que se controlen posibles errores, que hagamos el diagrama de clases y un gráfico que lo muestre cómo funciona todo el tinglado este que se trae la simulación con las máquinas, para eso tienes los folios extras.

Como recompensa, nos darán a elegir las famosas pastillitas. Si elegimos la azul..., bueno, eso será otra historia. Aunque lo mismo prefieres un buen aprobado 📖

“Programar sin una arquitectura o diseño en mente es como explorar una gruta sólo con una linterna: no sabes dónde estás, dónde has estado ni hacia dónde vas”  
— *Danny Thorpe*