

Sistemas de Tiempo Real

Informe Final

Simulación de la Estación Berazategui

Grupo de Desarrollo

Dehan, Lucas – 565/1

Di Marco, Gabriel – 1870/2

Kleinubing, Hernán – 1614/6

Palacio, Constantino – 1806/2

13/12/2023

Contenido

1. Introducción	3
2. Objetivos	3
3. Datos Utilizados	3
4. Herramientas de Simulación	4
5. Modelado de la Simulación.....	5
5.1. Agentes y Entorno.....	5
5.1.1. Plataforma	6
5.1.2. Reloj	7
5.1.3. Trenes.....	8
5.1.4. Personas.....	8
5.2. Salida Gráfica.....	10
5.3. Controles y Monitores	10
6. Ejecución de la Simulación	12
7. Conclusiones	14
8. Acceso al Código Fuente.....	15
9. Referencias.....	15

1. Introducción

El transporte público en Argentina es una parte fundamental de la vida diaria para una considerable porción de la población, especialmente en el Área Metropolitana de Buenos Aires.

La Línea Roca, que conecta la Capital Federal en la estación Plaza Constitución con las localidades del sur del Gran Buenos Aires y La Plata, despierta un interés particular debido a la gran cantidad de pasajeros que diariamente transitan por ella: en 2022, un total de 131.219.844 pasajeros viajó por alguno de los ramales de la Línea Roca.

En el proyecto anterior se abordó el estudio detallado de la estación Berazategui. Este lugar es frecuentemente transitado por aquellos que realizan transbordos entre los ramales de Bosques y La Plata. La simulación realizada tenía como objetivo contabilizar la cantidad de pasajeros en cada andén según la franja horaria, así como evaluar el uso del puente peatonal y analizar posibles situaciones de saturación.

En el proyecto actual, se ha decidido reimplementar la simulación haciendo uso de otra herramienta de trabajo buscando replicar resultados y comparar el rendimiento de ambos entornos de simulación.

2. Objetivos

La principal motivación detrás del presente proyecto es realizar una comparación de ambas herramientas de simulación utilizadas y determinar cuál de las dos resulta más adecuada para la resolución del problema presentado, considerando aspectos como:

- Impacto en el sistema: consumo de CPU/RAM
- Tiempo aproximado de simulación: medido desde que comienza la simulación hasta que termine de ejecutarse
- Complejidad del código resultante: legibilidad, modularización
- Sencillez de implementación: tanto el programa como los gráficos

Para realizar una comparación justa, se ejecutará la simulación usando ambas herramientas y se realizará un análisis de los resultados obtenidos, teniendo en cuenta los aspectos mencionados anteriormente.

3. Datos Utilizados

Los datos empleados fueron la cantidad de pasajeros por mes, adquiridos del Ministerio de Transporte, y se encuentran almacenados en un archivo Excel. Debido a la magnitud del conjunto de datos, se realizó un proceso de filtrado con el objetivo de seleccionar la información relevante para la simulación. Se han obtenido los siguientes datos para el año 2019, el utilizado en el modelado del problema:

E	F	M	A	M	J	J	A	S	O	N	D
212613	231473	263145	275146	285549	218172	197322	288444	308205	311335	308932	286682

Tabla 3-1. Datos filtrados para los meses del año 2019

Con el propósito de visualizar de manera efectiva los resultados, se incluye a continuación una gráfica que resume los meses seleccionados tras el proceso de filtrado.

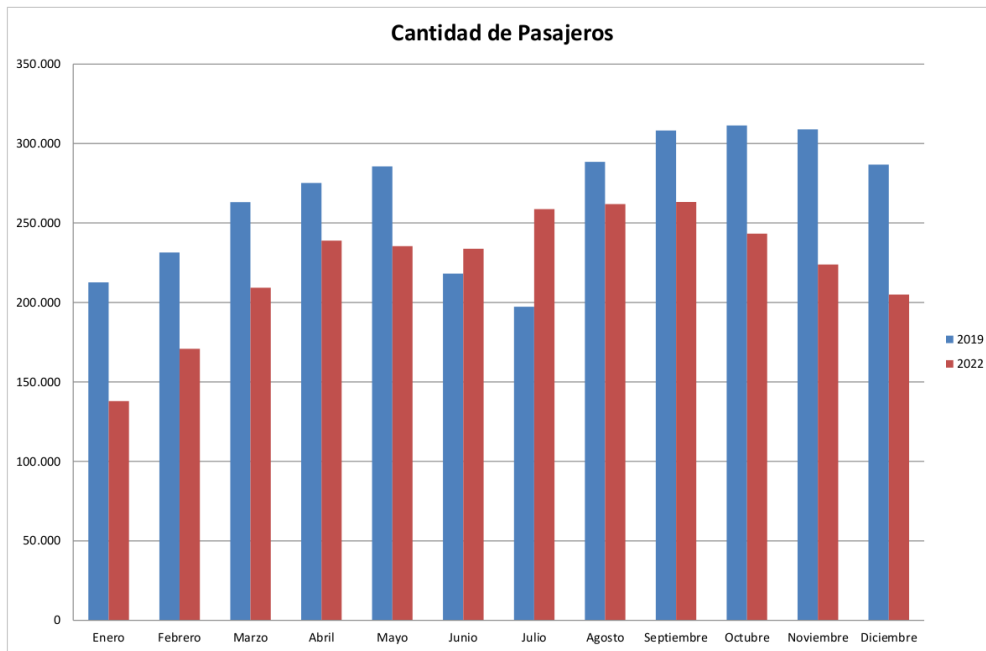


Figura 3-1. Cantidad de pasajeros mensuales en el año 2019 y el año 2022

La figura 3-1 presenta una comparativa de la cantidad de pasajeros que utilizaron la estación Berazategui en los años 2019 y 2022. Donde se observa el mes de octubre de 2019 y septiembre de 2022 que muestran el pico más alto en la afluencia de pasajeros en ambos años. Este análisis temporal revela patrones significativos en el comportamiento de la utilización de la estación a lo largo de estos dos años. Se seleccionó el 2022 como año de comparación porque el tráfico de pasajeros no fue el típico durante 2020 y 2021, y los datos de 2023 no están disponibles en su totalidad al no haber terminado el año.

4. Herramientas de Simulación

La simulación original fue implementada en el entorno NetLogo, basado en agentes y programado mediante un derivado del lenguaje LOGO, derivado a su vez del LISP. Es una herramienta de utilización sencilla que permite crear modelos complejos en una cantidad relativamente baja de sentencias de programación. Su paradigma central es el uso de “turtle graphics”, en el que cada agente es una “tortuga” que se puede controlar individualmente o en grupos de agentes.

Para realizar la simulación se utilizó la plataforma GAMA, un entorno de desarrollo de simulación basado en agentes implementado utilizando el lenguaje Java. La cátedra ofreció otras alternativas, como Repast, pero se optó por utilizar GAMA por ser más sencillo y flexible en cuanto a gráficos. Además, la plataforma GAMA es open source y la documentación es muy extensa y detallada.

A diferencia del NetLogo, GAMA no utiliza el paradigma “turtle graphics” para realizar la visualización, sino que la simulación se implementa en un lenguaje llamado GAML (*GAMA Modeling Language*), inspirado en Smalltalk y Java. Este cambio de enfoque resulta más intuitivo desde el punto de vista del programador. GAMA permite la herencia entre agentes y crear varios agentes de diferentes tipos de forma sencilla, así como implementar máquinas de estados finitos y otros mecanismos de control.

Además, GAMA brinda una gran flexibilidad para la creación de visualizaciones y trazado de gráficas; estando todas estas funciones extensamente documentadas con ejemplos de uso

exhaustivamente comentados. Particularmente, permite utilizar imágenes cargadas desde archivos externos para representar a los diferentes agentes de la simulación, lo que permite generar vistas gráficas de mayor impacto visual que aquellas realizadas con figuras básicas.

5. Modelado de la Simulación

Como guía para la realización del modelo se tomaron las aproximaciones realizadas en el proyecto anterior. Se ha estimado que en un día circulan unos 80 trenes por la estación, donde la frecuencia es de 15 minutos en horario regular y 12 minutos en horas pico (entre 5AM y 12PM y entre 5PM y 9PM).

Tomando un mes como un periodo estándar de 30 días para mayor simplicidad y tomando al mes de octubre como el peor caso, se establece que en un día viajan 11044 pasajeros. Si se considera la estimación de 80 trenes diarios, la cantidad de pasajeros por cada tren puede calcularse de la siguiente manera:

$$\frac{\text{Pasajeros}}{\text{tren}} = \frac{\text{Pasajeros}}{\text{mes}} * \frac{1 \text{ mes}}{30 \text{ días}} * \frac{1 \text{ día}}{2 \text{ andenes} * 80 \frac{\text{trenes}}{\text{andén}}} = \frac{\text{Pasajeros} * \text{día}}{30 \text{ días} * 160 \text{ trenes}}$$

$$\frac{\text{Pasajeros}}{\text{tren}} = \frac{\text{Pasajeros por mes}}{4800 \text{ trenes por mes}}$$

Teniendo en cuenta las horas pico y suponiendo que en este horario viajan el doble de pasajeros que en el horario regular, se establece que 81.48% de los pasajeros viajan en hora pico, mientras que el 18.52% restante lo hace en trenes de frecuencia 15 min.

Redondeando estos valores a 80% y 20%, respectivamente, se obtienen las siguientes fórmulas para determinar la cantidad de pasajeros que bajan de un tren en cualquier momento:

$$\text{cant_bajan} = \begin{cases} \frac{\text{Pasajeros por mes} * 80\%}{30 \text{ días} * 110 * 2}, & \text{si es hora pico} \\ \frac{\text{Pasajeros por mes} * 20\%}{30 \text{ días} * 50 * 2}, & \text{si es horario regular} \end{cases}$$

En la ecuación anterior también se tiene en consideración que en las horas pico transita la mayor cantidad de trenes: 55 de los 80 diarios en cada andén, un 68.75%. Al ser dos andenes, el total de trenes diarios es 160. Trasladando el porcentaje, se obtiene que 110 trenes circulan en las horas pico y los 50 restantes lo hacen en horario regular.

Con los datos obtenidos del Ministerio de Transporte y estas estimaciones se procedió a modelar a los agentes que componen la simulación.

5.1. Agentes y Entorno

El entorno de simulación está modelado en torno a agentes, cuyo comportamiento se implementa mayoritariamente a partir de máquinas de estados finitos debido a que las funciones a realizar dependen de tanto las acciones realizadas por otros agentes o de ocurrencia de ciertos eventos o del paso del tiempo.

En la simulación existen dos tipos de agentes: aquellos que participan activamente de la simulación ya aquellos agentes que brindan soporte a la simulación. Las personas y los trenes forman parte de los agentes que participan de la simulación, mientras que el reloj cumple la

función de contar el paso del tiempo de la simulación para que las acciones a llevarse a cabo puedan sincronizarse entre sí.

Otros agentes que brindan soporte a la simulación son los que componen el fondo de la pantalla de visualización gráfica (plataforma). Su principal función es crear la sensación de profundidad en la vista gráfica, es decir, solapar los elementos de tal forma que, por ejemplo, se vea a las personas caminar sobre el puente bajo el cual pasan los trenes. Estos agentes también sirven de puntos de anclaje para los desplazamientos de los pasajeros en la estación: los extremos del puente y las coordenadas de los andenes son utilizados para determinar cómo van a desplazarse las personas ante determinados eventos.

5.1.1. Plataforma

La plataforma es el más sencillo de los componentes de la simulación. Se compone de una grilla de 31 celdas de ancho por 31 de alto, donde cada elemento es un valor numérico a partir del cual se determina el elemento gráfico a dibujar en la pantalla. Este componente fue diseñado en una hoja de cálculo, tal como muestra la siguiente figura:

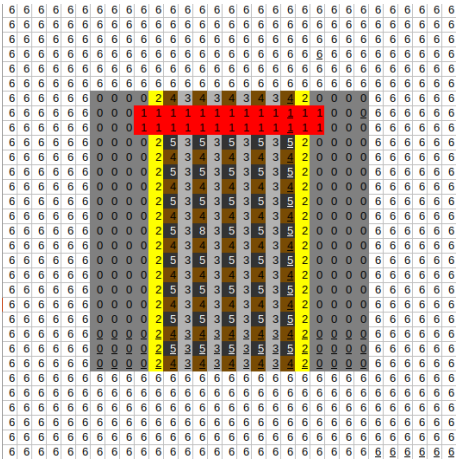


Figura 5.1.1-1. Matriz que representa la plataforma

La matriz se almacena como archivo de valores separados por coma (CSV) y es cargada por el sistema al iniciar la simulación. De acuerdo con los valores leídos se coloca en cada posición de la matriz uno de los siguientes elementos gráficos:

<i>Valor</i>	<i>Elemento colocado</i>
0	Piso del andén
1	Puente
2	Borde de andén
3	Vía (beta)
4	Durmiente (beta)
5	Fondo (beta)
6	Blanco
8	Vía

Tabla 5.1.1-1. Tabla de conversión entre matriz CSV y ventana gráfica

Los elementos de la tabla 5.1.1-1 marcados como “beta” son elementos que ya no se interpretan al cargarse desde el archivo de diseño. El elemento 8 es la imagen de la vía que, escalada, reemplaza a los elementos 3, 4 y 5. El fondo por defecto de la simulación es un color sólido.

El propósito del elemento 6 (blanco) es cubrir parte de la pantalla gráfica para que al eliminar a los agentes que ya no son necesarios, éstos no desaparezcan sin más, sino que parezca que dejan

la estación de tren. También se usa para disimular que los trenes permanecen detenidos hasta que la frecuencia les indique que deben dirigirse a la estación.

La siguiente figura muestra la vista gráfica de la simulación una vez cargados los elementos gráficos usando la tabla de diseño:

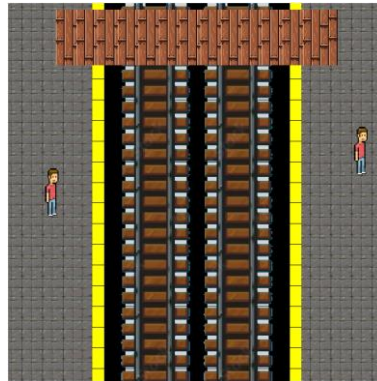


Figura 5.1.1-2. Vista gráfica de la simulación

Los elementos gráficos son solapados de forma tal de dar la sensación de profundidad, es decir, que los pasajeros caminen por encima del puente, mientras que el tren pasa por debajo de él. En el modelo en NetLogo, este efecto se logra ajustando el plano sobre el cual se ubicaba cada agente, mientras que en GAMA no existe otra forma que la aquí descrita.

5.1.2. Reloj

Este agente es el encargado de mantener un registro del paso del tiempo en la simulación, basándose en una escala de tiempo donde 100 pasos de simulación equivalen a un minuto en tiempo real. Esta escala es la misma utilizada en la simulación implementada en NetLogo.

En base a estos parámetros, se puede estimar que una hora real equivale a 6000 pasos en la simulación, por lo que la simulación completa (de 5AM a 10PM) demoraría unos 108000 pasos.

El agente reloj actualiza un contador de minutos cada 100 pasos de acuerdo con la base temporal elegida. Al contarse 60 minutos se actualiza un contador de horas, el cual es utilizado para determinar la cantidad de pasajeros que bajarán del tren durante las horas pico o si ingresan nuevas personas o no. Este valor es también usado para determinar el momento en que la simulación debe detenerse. Ambos contadores son compartidos por todos los agentes.

El comportamiento del reloj se modela como una máquina de estados de un solo estado que se ejecuta indefinidamente. Dicho estado se implementa según el siguiente diagrama de flujos:

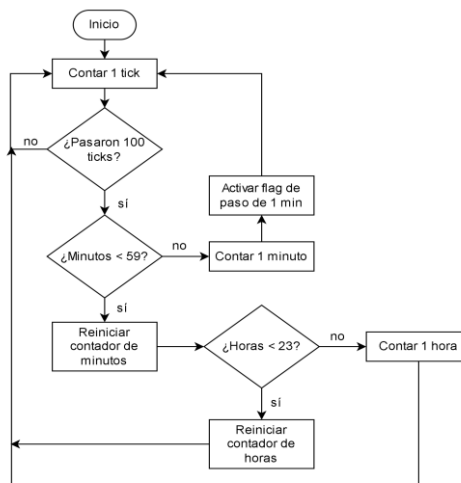


Figura 5.1.2-1. Diagrama de flujos del reloj

Tener en cuenta que al iniciar la simulación se inicializan los contadores de ticks, minutos y horas al valor 0. El flag de paso de un minuto presente en el diagrama de flujos se utiliza como marca temporal para los demás agentes.

Los valores de los contadores del reloj son globales, en principio, para permitir que los agentes realicen comprobaciones de horario para ajustar los parámetros a la hora pico o para determinar si deben crearse pasajeros nuevos o si debe detenerse la simulación.

5.1.3. Trenes

Los trenes se implementan como dos instancias de un mismo agente, que a su vez es modelado como una máquina de estados finitos compuesta por tres estados. La siguiente figura resume su estructura en forma de diagrama de estados.

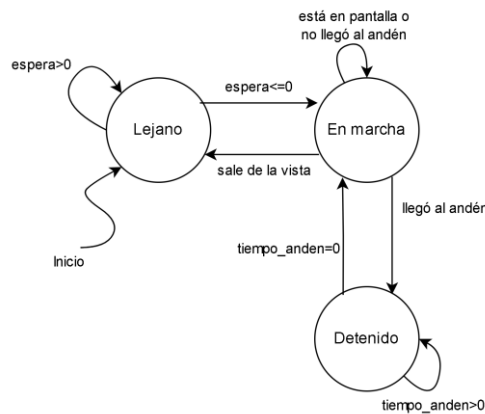


Figura 5.1.3-1. Diagrama de estados del agente `tren`

En el estado `lejano`, el tren se encuentra fuera de la vista (oculto, recordando el funcionamiento de la vista gráfica), simulando estar en otra parte del recorrido, lejos de la estación. Este es el estado inicial de ambos vehículos. Pasado un tiempo de espera, funcional a la frecuencia de cada tren, se pasa al estado `en_marcha`, en el cual el tren se vuelve visible y se muestra acercándose hasta el andén. En este estado, el tren se sigue moviendo hasta que llega a la mitad de la plataforma, momento en el cual se detiene por completo y pasa al estado `detenido`. Un tren detenido simplemente espera a que se agote el tiempo de espera (tiempo *de andén* o de *tolerancia*), para luego volver al estado `en_marcha`.

Es cuando el tren se detiene que se realiza el “intercambio” de pasajeros: aquellos que bajan se acomodan en el andén, mientras que los que suben se trasladan hacia la formación detenida.

Al regresar al estado `en_marcha`, el tren comienza a moverse hacia adelante, hasta desaparecer de la vista. Cuando alcanza el extremo de la ventana gráfica, se oculta el agente y se pasa al estado `lejano`, repitiéndose el ciclo.

Los parámetros que controlan el funcionamiento de los agentes se pueden configurar estáticamente mediante los controles o dinámicamente mediante la opción de ajuste de hora pico. En ambas situaciones existe un elemento de aleatoriedad: el tren puede retrasarse hasta 3 minutos o adelantarse hasta 1 minuto. Esto da realismo a la simulación, pues es una situación que ocurre bastante a menudo en las estaciones de tren de la vida real.

Ambos trenes son creados al momento de iniciarse la simulación, a través del constructor global, y los mismos no son destruidos al terminar la ejecución.

5.1.4. Personas

El comportamiento de las personas se modela de forma similar al de los trenes. Se crean múltiples instancias de un agente de tipo `persona`, el cual posee un comportamiento estandarizado basado en una máquina de estados finitos compuesta por cinco estados principales

y dos adicionales para realizar acciones de control. La siguiente figura muestra el diagrama de estados para un agente de tipo *persona*.

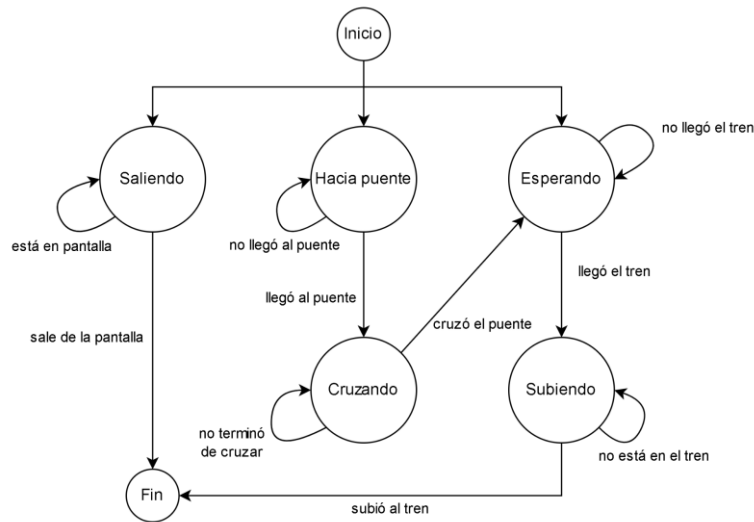


Figura 5.1.4-1. Diagrama de estados para el agente *persona*

Una persona puede empezar en cualquiera de tres estados. Como GAMA no permite especificar múltiples estados iniciales en una máquina de estados, se requiere un estado *inicio*, el cual determina en base a un valor numérico el estado en que debe arrancar el agente, que puede ser *saliendo*, *hacia_puente* o *esperando*.

En el estado *saliendo*, una persona se dirige hacia los molinetes de la estación (modelados como alguno de los extremos inferiores de la vista gráfica). Cuando sale de la pantalla, el agente se elimina, simulando que la persona deja la estación.

En el estado *hacia_puente*, una persona se dirige hacia el puente peatonal (*cruzando*), desde la posición actual y determinando automáticamente si debe ir al extremo izquierdo o derecho basándose en su posición relativa a la mitad de la pantalla. Una persona que cruza el puente se dirige de un extremo del puente al otro, para luego acomodarse en alguna posición del andén en espera del tren, simbolizado mediante la transición al estado *esperando*.

En el estado *esperando*, una persona permanece en su sitio hasta que el tren se detiene en la estación. Cuando esto ocurre, la persona se mueve hacia la formación detenida (estado *subiendo*) y el agente es eliminado al llegar al tren, nuevamente para simular que la persona se subió y dejó la estación.

Ambas menciones de la eliminación de un agente implican la transición a un estado de control llamado *fin*. En este estado simplemente se borra al agente de la simulación.

El movimiento de las personas se implementa mediante las habilidades (o *skills*) que provee el entorno GAMA. En esencia, permite mover los agentes a partir de un par de coordenadas y un valor de velocidad. Por ejemplo, para que una persona se mueva desde los molinetes hasta el extremo del puente peatonal, basta con especificar al agente el objetivo (*target*), es decir, las coordenadas del extremo del puente, y una velocidad (por ejemplo, 0.8). Este mecanismo se utiliza para todo movimiento que realicen los agentes, no solo las personas.

Las personas son creadas de varias formas. Una de ellas es cuando un tren se detiene en la estación. Las personas que bajan del tren son nuevos agentes instanciados en el andén, tanto en el estado *hacia_puente* como *saliendo*, dependiendo del valor asignado a la probabilidad de cruce de las personas. Tener presente que solamente van a cruzar aquellos pasajeros que bajen del andén izquierdo, pues solamente en el andén derecho se pueden realizar transbordos. Los pasajeros que bajen en el andén derecho se limitarán a bajarse del tren e irse.

La otra forma de crear agentes es al azar: cada 100 ticks se crea un nuevo agente persona en cualquiera de los dos andenes en el estado `hacia_puente` o `esperando`. Esto simula que una persona ingresa a la estación desde los molinetes y realiza un viaje en uno u otro sentido. La creación espontánea de pasajeros ocurre hasta las 9PM, simulando que a esa hora están llegando los últimos trenes a la estación y puede desactivarse completamente a partir de los controles de la simulación.

5.2. Salida Gráfica

Como salida gráfica se realiza una gráfica comparable con la del proyecto original: se trazan las cantidades de pasajeros en ambos andenes y en el puente en diferentes colores. Se respeta la convención del informe previo: en azul se representan los pasajeros en el andén izquierdo, en verde los pasajeros en el andén derecho y en rojo aquellos que están en el puente. La gráfica se actualiza cada 10 ticks.

La siguiente figura muestra la gráfica resultante para una determinada configuración de los parámetros de entrada de la simulación, a modo de ejemplo:

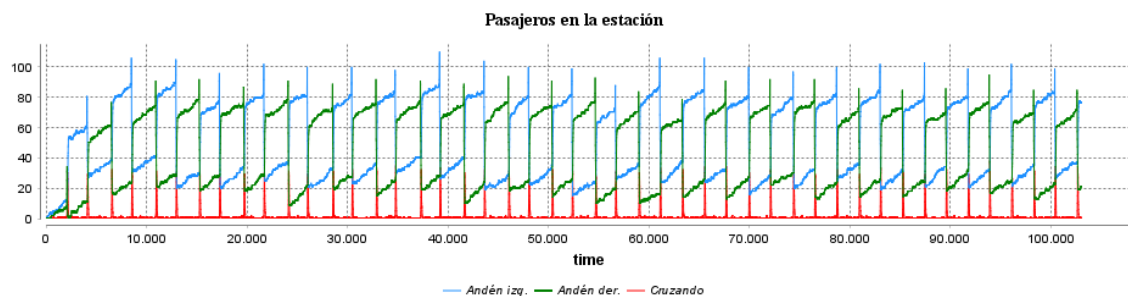


Figura 5.2-1. Salida gráfica de una simulación completa

Cabe destacar que se observó que la presencia de esta gráfica ha prolongado significativamente el tiempo de la simulación. Durante aproximadamente los primeros 20000 ticks, la simulación se ejecuta a una velocidad rápida, comparable con la velocidad de la simulación en NetLogo. A partir de ese momento, la velocidad de simulación desciende a una tasa aparentemente constante hasta llegar al punto de una velocidad aproximada de 1 tick/minuto real.

El consumo de memoria nunca llega al límite por defecto de GAMA (4GiB), por lo que se desconoce la causa de este problema. Aun así, si se cierra la ventana de la gráfica, la simulación se ejecuta normalmente a velocidad razonable. El valor de tiempo de ejecución medido para la simulación completa **sin** el trazado de la gráfica es de aproximadamente 2 minutos, por lo que se concluye que esta funcionalidad es la causa de la caída del rendimiento del modelo.

5.3. Controles y Monitores

A la simulación se han agregado los siguientes controles para la configuración de los parámetros iniciales. Se respeta la convención de nombres usada en el proyecto original:

- `hora_inicial`: selecciona la hora a la que arranca la simulación. Por defecto son las 5AM, pero permite elegir entre las 5AM y las 10PM con una resolución de 1 hora.
- `mes`: selecciona entre los datos de los diferentes meses, es decir, la cantidad de pasajeros que van a bajar del tren. Por defecto es "Enero".
- `tolerancia_anden`: tiempo (en ticks) que el tren espera a que suban/bajen los pasajeros, por defecto 200. Permite seleccionar entre 100 y 500, con paso 10 ticks.
- `frecuencia_tren`: tiempo que un tren tarda en llegar a la estación, en ticks. Por defecto, 4000. Se puede seleccionar entre 1000 y 4000 ticks, en pasos de 100. Si la opción `ajuste` se encuentra activada, esta opción no tiene efecto en la simulación.

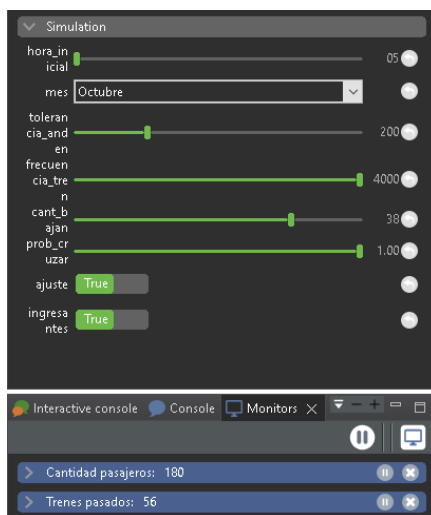
- `cant_bajan`: cantidad de pasajeros que bajan cada vez que el tren se detiene en la estación. Su valor por defecto es 38. Puede tomar valores entre 0 y 50, con paso 1. No surte efecto en la simulación si `ajuste` está activada.
- `prob_cruzar`: equivalente a `porcentaje_cruzan` del proyecto original. Indica la probabilidad de que un pasajero realice un transbordo al bajar del tren. El valor por defecto es 100% (1.0). Puede tomar valores entre 0% y 100% en pasos de 1%.
- `ajuste`: permite activar o desactivar la opción de ajuste automático de cantidades. Si se activa el ajuste, se regula la cantidad de pasajeros que bajan de acuerdo con la definición de hora pico y la frecuencia de los trenes.
- `ingresantes`: permite activar o desactivar la creación espontánea de pasajeros en los molinetes de la estación. Si está activada, se creará un pasajero nuevo en cualquiera de los dos andenes (con igual probabilidad) cada cierto tiempo.

Para la visualización de algunos parámetros de la simulación en el transcurso de la misma se provee un conjunto de monitores. Estos no son los mismos utilizados en la simulación del año anterior. Agregar exactamente los mismos monitores resultaría en un modelo mucho más complejo, debido a las diferencias de implementación entre NetLogo y GAMA. Los monitores agregados al modelo son los siguientes:

- Cantidad de pasajeros: cuenta el total de pasajeros en la estación, tanto en los andenes como en el puente peatonal
- Trenes pasados: cuenta la cantidad de trenes que se han detenido en la estación en el transcurso de la simulación.

Un monitor adicional solapado sobre la vista gráfica es el reloj, que permite ver la hora actual de la simulación. Este monitor se actualiza cada 100 ticks (base temporal).

Los controles y monitores descritos anteriormente pueden verse en las figuras 5.3-1a y b. La figura 5.3-1a permite observar los controles y monitores “de texto”, mientras que la figura 5.3-1b muestra el reloj solapado a la vista gráfica, junto a los agentes.



(a)



(b)

Figura 5.3-1. (a) Controles y monitores. (b) Vista gráfica con reloj.

6. Ejecución de la Simulación

A efectos de comparación de resultados con la implementación previa se ejecuta la simulación con los siguientes parámetros de entrada:

- Hora inicial: 5AM
- Mes: octubre 2019
- Tolerancia andén: 200 ticks
- Probabilidad de cruce: 100%
- Ingresantes: sí

Se activa el ajuste automático de hora pico, por lo que los valores iniciales de los parámetros `frecuencia_tren` y `cant_bajan` no son relevantes. Los resultados obtenidos se sintetizan en la siguiente figura:

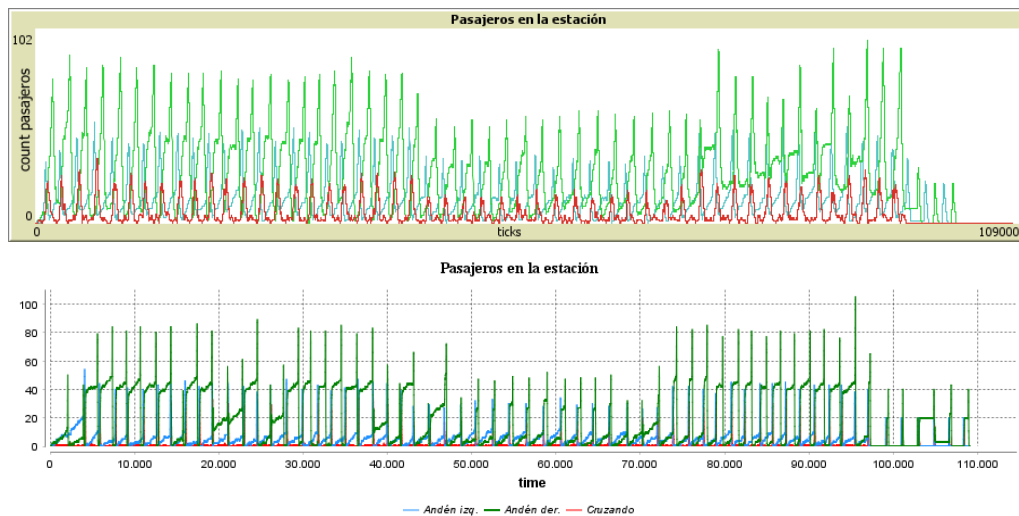


Figura 6-1. Resultados de la simulación en NetLogo (arriba) y GAMA (abajo)

En la figura 6-1 se puede observar no solo que los resultados con ambas herramientas son comparables entre sí, sino que, al haber ejecutado la simulación con la opción de ajuste activada, se ven claramente las franjas horarias de mayor flujo de pasajeros: entre aproximadamente las 5AM y las 12PM se ve una concentración de un promedio de 80 pasajeros en el andén derecho y alrededor de 50 en el izquierdo. Esta diferencia se debe a que sólo se realizan transbordos en el andén derecho. La demanda del transporte cae hasta aproximadamente las 4:30PM, donde se empiezan a repetir los resultados obtenidos en la mañana hasta cerrar la estación a las 10PM.

Con el objetivo de comparar el rendimiento de ambas herramientas, se decidió probar con el peor caso bajando la tolerancia del andén a 20 ticks. Es de esperar que el sistema se sature, debido a la cantidad de pasajeros acumulados en los andenes por no tener tiempo suficiente para subir al tren.

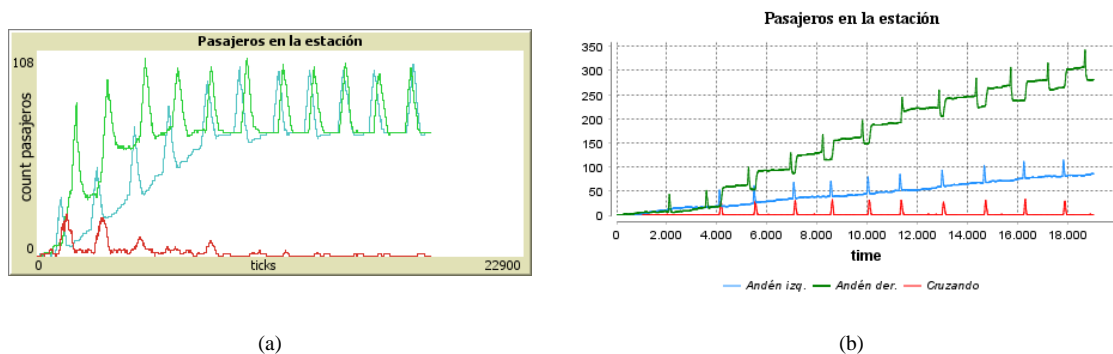


Figura 6-2. Caso de saturación del transporte: (a) NetLogo y (b) GAMA.

En la figura 6-2 se puede observar que la estación nunca se vacía, demostrando que el sistema se satura con un tiempo de tolerancia muy pequeño. Al no tener suficiente tiempo para subir al tren, los pasajeros quedan aglomerados en los andenes indefinidamente, produciéndose los resultados mostrados en la figura 6-2 (b).

La diferencia principal con la implementación en NetLogo es que la cantidad de pasajeros oscila entre un rango relativamente estable de valores, mientras que en la implementación en GAMA estos números crecen indefinidamente. También cabe destacar que el consumo de recursos del sistema aumenta de manera apreciable entre una ejecución normal (con los valores por defecto) y el caso extremo de saturación del modelo. El límite de GAMA de 4GiB es alcanzado varias veces en el transcurso de esta prueba. Cuando esto ocurre, se detiene la simulación momentáneamente hasta que el *garbage collector* de la plataforma logra liberar memoria suficiente para seguir trabajando.

A partir de la ejecución anterior se alteran los parámetros de inicio de la siguiente manera. Si se desactiva la opción de ajuste, dejando la tolerancia al valor por defecto (200 ticks), y aumentando la frecuencia a 4000 ticks (40 minutos), se observan los siguientes resultados:

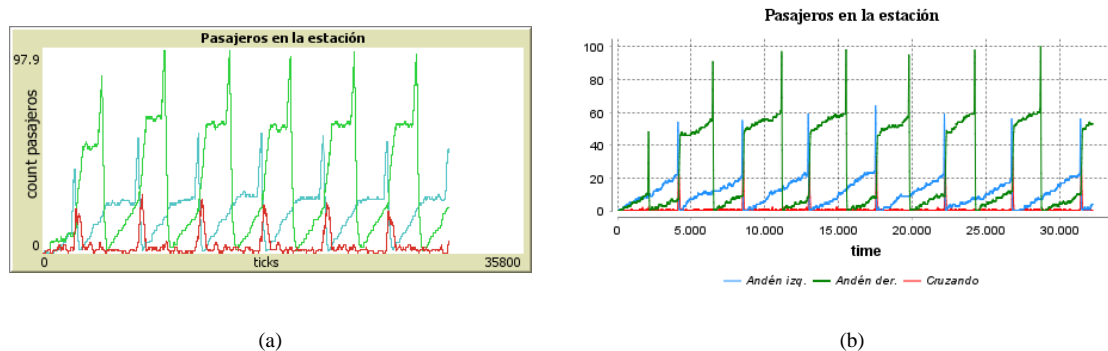


Figura 6-3. Caso extremo de frecuencia baja: (a) NetLogo y (b) GAMA

La figura 6-3 muestra que, al bajar la frecuencia a 40 minutos, ocurre un desbalance considerable en la cantidad de pasajeros en ambos andenes. Este desbalance también ocurre cuando la frecuencia de los trenes es muy alta (10 minutos). Al ser muy próximos entre sí los arribos de trenes, se incrementa notablemente el flujo de pasajeros en la estación, habiendo más de 100 personas en el andén derecho en todo momento de la simulación, tal como se muestra en la figura 6-4.

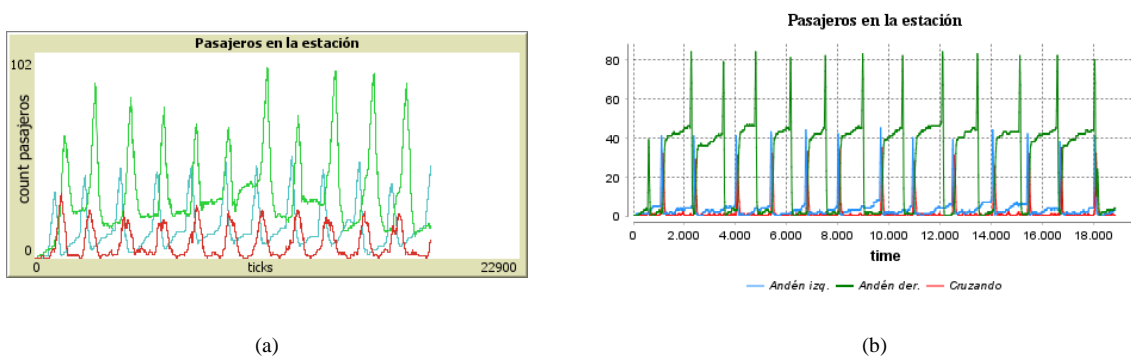


Figura 6-4. Caso extremo de frecuencia alta: (a) NetLogo y (b) GAMA

7. Conclusiones

A partir de los resultados de la simulación, se puede concluir que el rendimiento del modelo implementado en NetLogo es mejor, pues el tiempo de la simulación completa es mucho menor al obtenido con GAMA. Se pueden listar las siguientes ventajas de la utilización de NetLogo en vez de GAMA:

- La simulación se ejecuta a mucha mayor velocidad. GAMA, al estar basado en Java, inherentemente requiere una mayor capacidad de cómputo y una cantidad mayor de memoria, lo que influye en la velocidad de la simulación y, por tanto, en su duración.
- La manipulación de los elementos gráficos es más sencilla. Agregar monitores, modificar los controles y trazado de gráficas es prácticamente instantáneo en NetLogo, mientras que esta misma tarea requiere código dentro del modelo de GAMA.
- Aunque menos legible, el programa resultante es más corto y no se requiere de un entorno completo de desarrollo, pues el código fuente es un archivo suelto que no debe importarse como proyecto (como en GAMA).

Sin embargo, la plataforma GAMA es superior a NetLogo en los siguientes aspectos:

- La implementación resulta más sencilla en GAMA que en NetLogo. El lenguaje de programación GAML (basado en Java) resulta más intuitivo desde el punto de vista del programador que el derivado de LOGO que utiliza NetLogo.
- La delimitación entre los módulos del programa (agentes, comportamiento, salidas) es más clara que en NetLogo. GAMA permite realizar una división bien marcada entre el código de los diferentes agentes y las funciones que implementan su comportamiento.
- La plataforma GAMA posee una documentación extensa y muy detallada, con ejemplos de implementación y tutoriales en línea. El entorno de desarrollo incluye proyectos enteros con ejemplos de implementación y elementos de las vistas gráficas.

En base a lo anterior se determina que, al menos en el sistema de pruebas utilizado, el rendimiento de NetLogo es superior al de GAMA: el proyecto implementado en GAMA tiene un tiempo total de simulación que excede las 12 horas a una velocidad promedio comparable con 1 tick de simulación por cada minuto real, mientras que el proyecto de NetLogo tiene un tiempo de ejecución total de aproximadamente 1 minuto.

Simulaciones sencillas, como el ejemplo de *prey-predator* disponible en la página de tutoriales de GAMA, se ejecutan sin problemas a una velocidad razonable en el equipo de pruebas, por lo que se concluye en que la plataforma GAMA no resulta adecuada para el desarrollo de modelos complejos, particularmente en comparación con NetLogo y en la plataforma de hardware acotada con la que se cuenta.

Teniendo en cuenta lo mencionado en 5.2, si se lograra implementar un mecanismo de exportación de los datos en tiempo real a un archivo CSV, se evitaría el trazado del gráfico y así se aceleraría la simulación considerablemente. Aun así nunca se alcanzaría el nivel de performance que tiene el modelo en NetLogo y se agregaría la tarea adicional de utilizar un programa de procesamiento numérico como MATLAB para armar la gráfica en base al archivo resultante. Se reafirma entonces la conclusión de que la plataforma GAMA no resultaría adecuada para modelizar la situación presentada en este informe.

8. Acceso al Código Fuente

Todo el material relevante a la realización de este proyecto se incluye en el siguiente repositorio de GitHub: https://github.com/Constantino-Palacio/proyecto_tiempo_real.git. Es de acceso público y contiene:

- **Datos utilizados para la simulación:** copia del documento en formato de Microsoft Excel descargado desde la página del Ministerio de Transporte (actualizado en julio de 2023).
- **Proyecto 2022:** código fuente e informe final del proyecto de simulación original implementado en NetLogo, para referencia.
- **Proyecto 2023:** código fuente del presente proyecto (importar al entorno de trabajo de GAMA para su ejecución, el código fuente en sí mismo no alcanza)
- **Documentación:** un documento en formato PDF con el presente informe, incluyendo también las presentaciones de la propuesta y el avance del proyecto. Se incluye un video de aproximadamente un minuto de duración, mostrando el progreso a la fecha del avance.

9. Referencias

- [1] Sitio de descarga del software GAMA (disponible al 06/12/2023): <https://gama-platform.org/download>.
- [2] Sitio web del software NetLogo (disponible al 06/12/2023): <https://ccl.northwestern.edu/netlogo/>.
- [3] Documentación del entorno GAMA (disponible al 06/12/2023): <https://gama-platform.org/wiki/Home>.
- [4] Tutoriales de GAMA (disponible al 06/12/2023): <https://gama-platform.org/wiki/Tutorials>.
- [5] Blanco, V. et al. *Simulación de la Estación Berazategui*. 14/12/2022. Incluido como adjunto en el repositorio del proyecto.
- [6] Ministerio de Transporte. *Boletos Vendidos por Estación – Línea Roca*. Actualizado por última vez el 09/08/2023, incluido como adjunto en el repositorio del proyecto.