



I115 SISTEMAS DE TIEMPO REAL

Trabajo de simulación

Simulación de la Línea C del Subte de Buenos Aires entre sus estaciones terminales

Grupo 3

Bejarano, Abril¹ (03339/5)
Majoros, Lorenzo¹ (03296/1)
Seery, Juan Martín^{1*} (03471/9)
Seijo, Gerónimo¹ (01859/7)

17/12/2025

¹{abril.bejarano,lorenzomajoros,juan.seery,seijo.geronimo}@alu.ing.unlp.edu.ar

* Autor responsable del informe

Resumen — La Ciudad Autónoma de Buenos Aires cuenta con uno de los sistemas de trenes subterráneos más importantes de América: el Subte de Buenos Aires. El mismo es utilizado por millones de pasajeros diariamente para movilizarse por la ciudad. En particular, la Línea C del Subte conecta dos estaciones cabeceras que permiten la interconexión con los ferrocarriles metropolitanos, resultando en un punto neurálgico para el transporte público. En este trabajo, se desarrolla un modelo basado en agentes para simular la dinámica entre los pasajeros que llegan desde los ferrocarriles metropolitanos y la frecuencia de los trenes subterráneos en la Línea C. El modelo permite analizar distintas configuraciones del sistema y optimizar la frecuencia de los trenes para mejorar el servicio ofrecido a los usuarios.

Índice

1. Introducción	2
2. Marco teórico	3
2.1. Modelo basado en agentes	3
3. Metodología	5
3.1. Descripción del modelo	5
3.1.1. Trenes	5
3.1.2. Pasajeros	6
3.2. Implementación	6
3.2.1. <i>Patches</i>	8
3.2.2. Trenes (<i>Turtle</i>)	9
3.2.3. Pasajeros (<i>Turtle</i>)	10
3.2.4. <i>Monitors</i>	11
3.3. Replicación de experimentos	12
4. Resultados	13
4.1. Optimizaciones	13
5. Conclusiones	14
Bibliografía	14
A. Simulaciones base	15
B. Simulaciones optimizadas	17

1. Introducción

El Subte de Buenos Aires es un sistema de trenes subterráneos ubicado debajo de las calles de la Ciudad Autónoma de Buenos Aires. El sistema resulta de gran importancia para los porteños, contabilizando poco más de 199 millones de pasajeros en 2024 [1].

En particular, la Línea C resulta de particular importancia al realizar un recorrido troncal a través la red, como se aprecia en la Figura 1. La misma registró al rededor de 31 millones de pasajeros en 2024 [1] y conecta dos estaciones de los ferrocarriles metropolitanos de Buenos Aires: Retiro (de donde parten las líneas Mitre, Belgrano Norte y San Martín hacia el norte) y Plaza Constitución (de donde parte la línea Roca hacia el sur). La coordinación entre estos sistemas de movilidad es fundamental para que los usuarios puedan trasladarse de manera consistente, predecible y veloz.

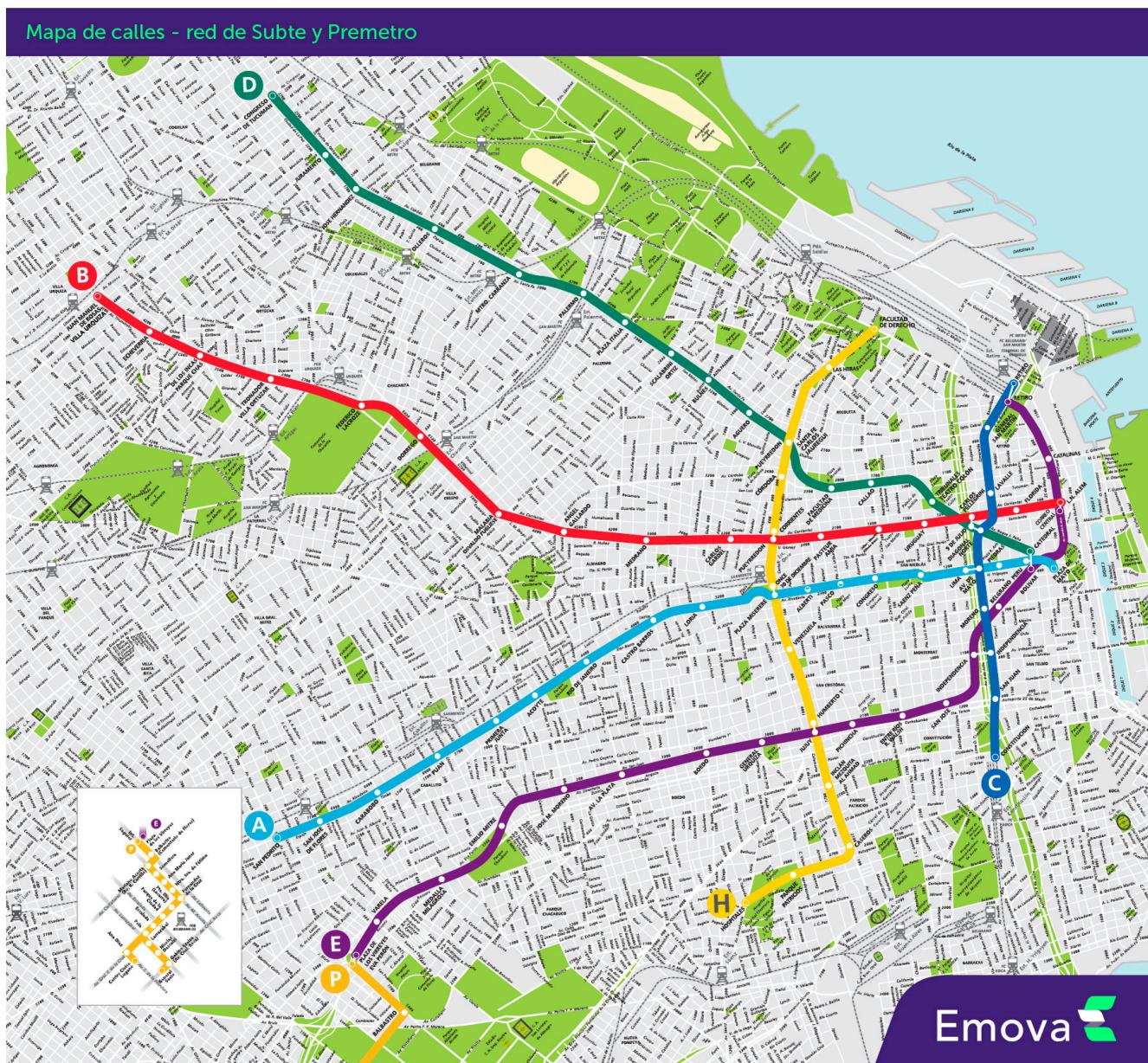


Figura 1. Mapa esquemáticos de las calles de red del Subte de Buenos Aires (obra de Emova).

En este trabajo, se estudia la Línea C entre sus estaciones terminales y se analizan las dinámicas entre los pasajeros que llegan desde los ferrocarriles metropolitanos y la frecuencia de los trenes subterráneos. Resulta importante este estudio porque permite optimizar los recursos del Subte para mejorar su servicio.

2. Marco teórico

Para que el estudio sea lo más verídico posible, se utilizaron datos de uso de los molinetes en el año 2024 [2]. Luego de analizar los datos en las estaciones de Retiro ([Figura 2](#)) y Constitución ([Figura 3](#)), se decidió centrarse únicamente en un uso “normal” de la red — esto es, entre los meses de marzo y noviembre inclusive para evitar fluctuaciones debido a los meses de calor y fiestas/vacaciones.

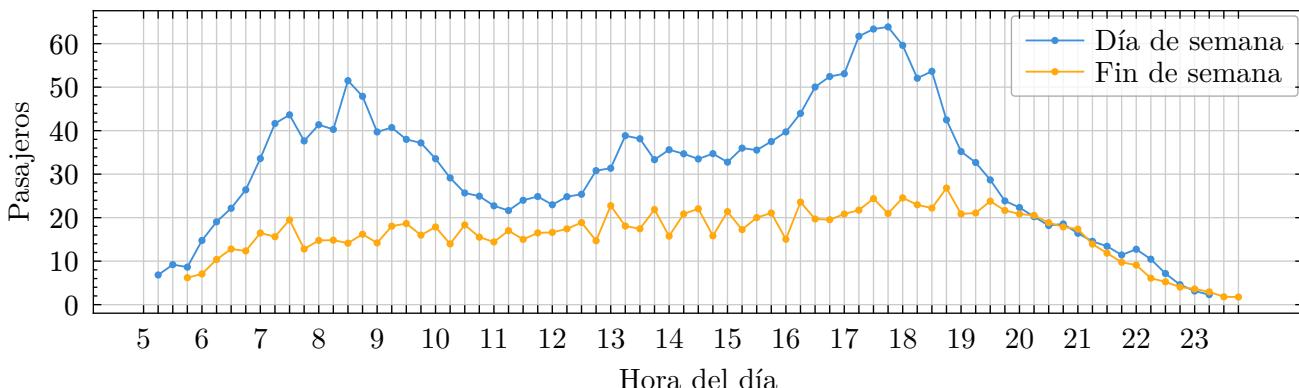


Figura 2. Histograma de pasajeros en la estación de Retiro. Cada punto representa el flujo de personas a lo largo de los 15 minutos siguientes. Limitado entre marzo y noviembre de 2024.

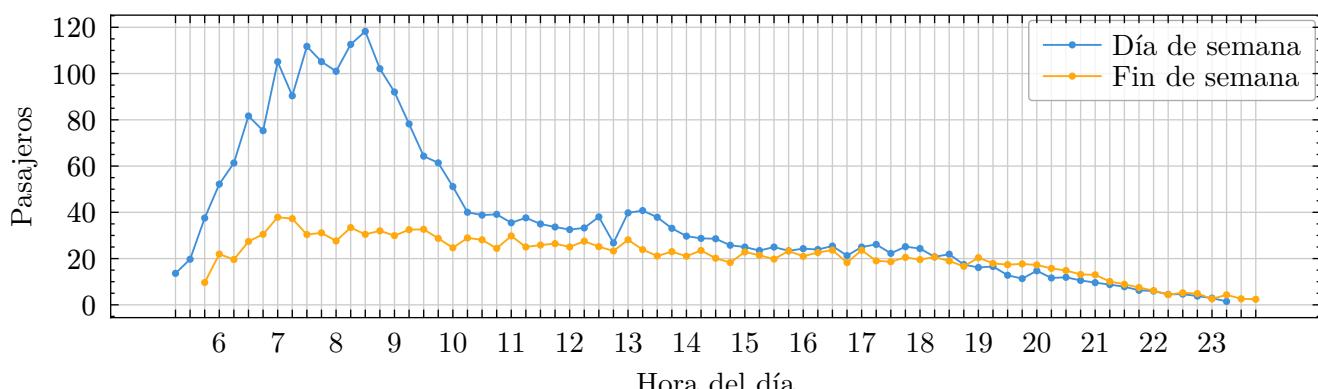


Figura 3. Histograma de pasajeros en la estación de Constitución. Cada punto representa el flujo de personas a lo largo de los 15 minutos siguientes. Limitado entre marzo y noviembre de 2024.

Dentro del horario “normal”, se agruparon los datos en dos curvas. Por un lado, los “días de semana”, lunes a viernes, caracterizado por el uso de los trabajadores que se mueven desde el Gran Buenos Aires hacia la ciudad a la mañana y vuelven a sus hogares a la tarde. Por otro lado, el “fin de semana”, caracterizado por la recreación.

2.1. Modelo basado en agentes

Con los datos obtenidos, se desarrolló un modelo de la Línea C del Subte de Buenos Aires. Particularmente, se sintetizó un modelo basado en agentes.

Los modelos tradicionales se basan en una descripción matemática de la realidad. Estos son limitados por las herramientas matemáticas disponibles, como las ecuaciones diferenciales. Cuando el sistema es muy complejo, las ecuaciones diferenciales pueden resultar limitantes o requerir mucho cálculo para resolverlas. Por ello, los sistemas suelen simplificarse en sistemas lineales para ser útiles.

En cambio, los modelos computacionales aprovechan que las computadoras pueden realizar numerosas operaciones matemáticas simples en muy poco tiempo, habilitando obtener resultados valiosos de modelos no lineales. Un tipo de modelo computacional son los **modelos basados en agentes** (ABM, por sus siglas en inglés, *Agent-based model*) [3, p. 10]. Lo particular de este tipo de modelos es que no se modela el sistema sí, sino que se pone énfasis en cada componente individual (cada *agente*). Cada agente cuenta con sus propias variables internas y un comportamiento predefinido con cada otro agente y con el

entorno. Cada agente debe ser descrito según sus **propiedades** (variables internas) y sus **acciones** (o comportamiento con otros agentes y su entorno) [4, p. 205].

Este marco teórico resulta conveniente, ya que permite modelar a cada agente de manera individual sin saber el comportamiento del sistema como un todo. Para el caso del Subte, donde se desconocen estas dinámicas de antemano, se justifica el uso de este tipo de modelado.

3. Metodología

Para estudiar la Línea C se desarrolló un ABM que permite aprovechar las capacidades computacionales para simular distintos escenarios.

3.1. Descripción del modelo

El modelo consta de dos agentes móviles: **persona**, que describe a un usuario del sistema y **tren** que describe una formación de vagones del Subte. Los mismos existen en la Línea C del Subte (su entorno), graficado en la [Figura 4](#).

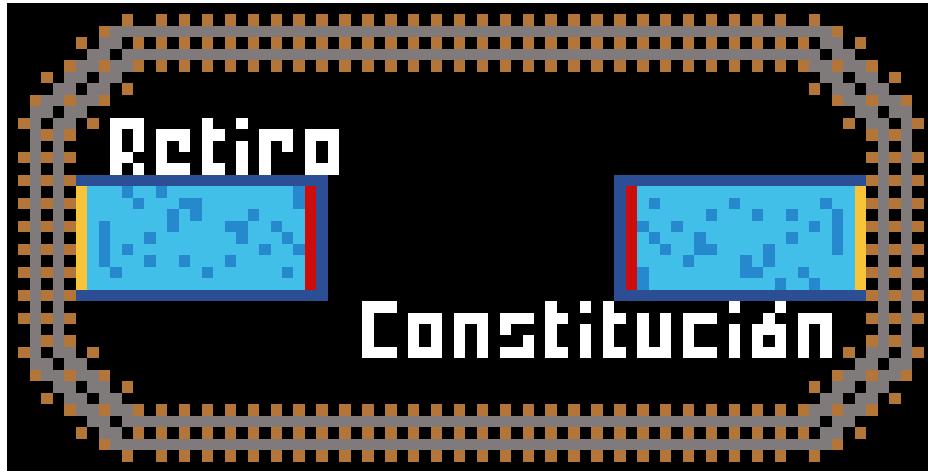


Figura 4. Representación gráfica del entorno del modelo de la Línea C.

3.1.1. Trenes

Los trenes circulan por las vías en sentido horario y se detienen en las estaciones. Para este modelo, se asumen trenes uniformes (misma capacidad de pasajeros y velocidad máxima) y que los mismo no deben “ingresarse” en circulación, solo aparecen en las vías cuando sea la hora de apertura. Su comportamiento puede describirse según la máquina de estados finita de Moore de la [Figura 5](#).

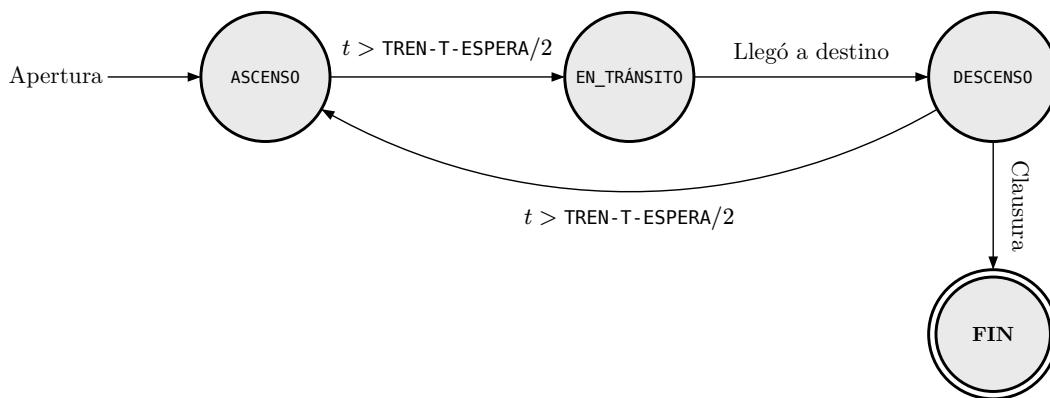


Figura 5. Máquina de estados de los trenes. Los lazos de espera no se graficaron para simplificar.

Cada tren puede encontrarse en uno de tres estados.

- **ASCENSO**: el tren se encuentra detenido en una estación por un tiempo fijo ($TREN-T-ESPERA/2$), luego pasa al estado **EN_TRÁNSITO**. Durante este estado, se admite la subida de pasajeros.
- **EN_TRÁNSITO**: el tren se mueve entre dos estaciones. Comienza acelerando hasta su velocidad máxima para luego frenar y parar en su destino. Al llegar al destino, pasa al estado **DESCENSO**.
- **DESCENSO**: el tren se encuentra detenido en una estación por un tiempo fijo ($TREN-T-ESPERA/2$). Durante este estado, se admite el descenso de pasajeros. Si el Subte sigue en funcionamiento, luego pasa al estado **ASCENSO**; de lo contrario, el tren sale del sistema (**FIN**).

Además, los trenes son parametrizados, permitiendo variar sus propiedades para simular distintos escenarios.

- **max-pasajeros**: pasajeros máximos que soporta la formación.
- **TREN-V-MAX [m/s]**: velocidad máxima de los trenes.
- **TREN-ACC [m/s²]**: aceleración del tren.

Se estimaron parámetros base para realizar comparaciones. Se simula que la distancia entre ambas estaciones es de 4,5 km (aproximado según los mapas de Emova). Además, también según estimaciones realizadas por los autores, el recorrido entre cabeceras tarda aproximadamente 13 min. Esto resulta en una velocidad media de los trenes de 20 km/h.

3.1.2. Pasajeros

El otro agente móvil es el pasajero. Se asume que todo pasajero usuario de la Línea ingresa por las cabeceras, acorde al estudio deseado de la sincronización con los ferrocarriles metropolitanos. Las estaciones intermedias no resultan de interés porque se quiere ver si los trenes pueden recoger a la mayor cantidad de pasajeros desde las cabeceras, para luego bajar donde sea necesario — además, al ser cabeceras, no importa si el tren llega lleno porque necesariamente se vacía antes del ascenso de nuevos pasajeros.

De este modo, su comportamiento puede describirse según la máquina de estados finita de Moore de la [Figura 6](#). Cada pasajero idealmente espera a un tren, se sube y viaja; pero se le agregó un parámetro de *paciencia* que le permite desistir y movilizarse de otra manera si espera demasiado.

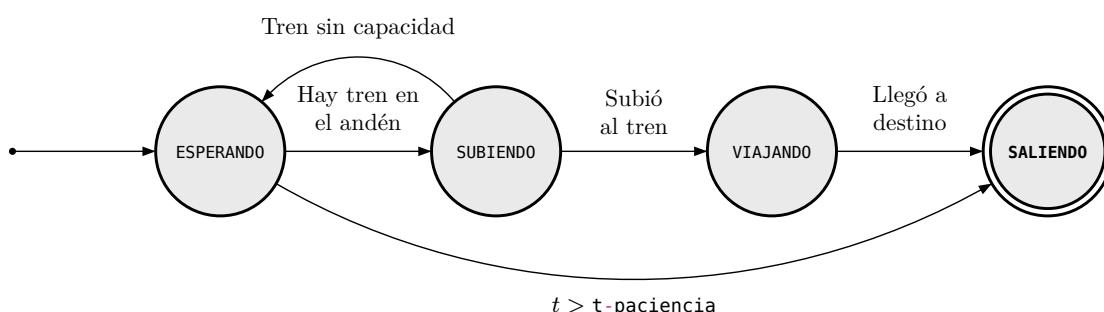


Figura 6. Máquina de estados de los pasajeros. Los lazos de espera no se graficaron para simplificar.

Cada pasajero puede encontrarse en uno de cuatro estados.

- **ESPERANDO**: el pasajero se encuentra en la estación esperando a que llegue el próximo tren. En este estado, incrementa su variable interna *t-esperando*. Si se supera su umbral de paciencia, el mismo pasa al estado **SALIENDO**. Si llega un tren en estado de **ASCENSO**, el pasajero pasa al estado **SUBIENDO**.
- **SUBIENDO**: el pasajero se dirige hacia el tren. Al llegar a las puertas, verifica si el tren tiene espacio disponible. Si lo tiene, se sube y pasa al estado **VIAJANDO**; de lo contrario, pasa nuevamente al estado **ESPERANDO**.
- **VIAJANDO**: el pasajero espera sereno a que el tren llegue a su destino. Al pasar el tren a su estado de **DESCENSO**, el pasajero pasa a su estado de **SALIENDO**.
- **SALIENDO**: desde su posición actual, el pasajero se dirige hacia la salida más cercana para luego desaparecer de la simulación.

Cada pasajero puede ser parametrizado con dos opciones.

- **destino**: hacia dónde se dirige.
- **t-paciencia [s]**: el tiempo máximo dispuesto a esperar en el andén.

Los mismos no aparecen de forma azarosa en las estaciones. Deben aparecer siguiendo los histogramas de molinetes ([Figura 2](#) y [Figura 3](#)).

3.2. Implementación

Este modelo se implementó en NetLogo 7.0.0, una herramienta diseñada para implementar ABM [\[5\]](#), que permite definir agentes, sus propiedades y comportamientos de manera sencilla. La [Figura 7](#) muestra la interfaz de la simulación.

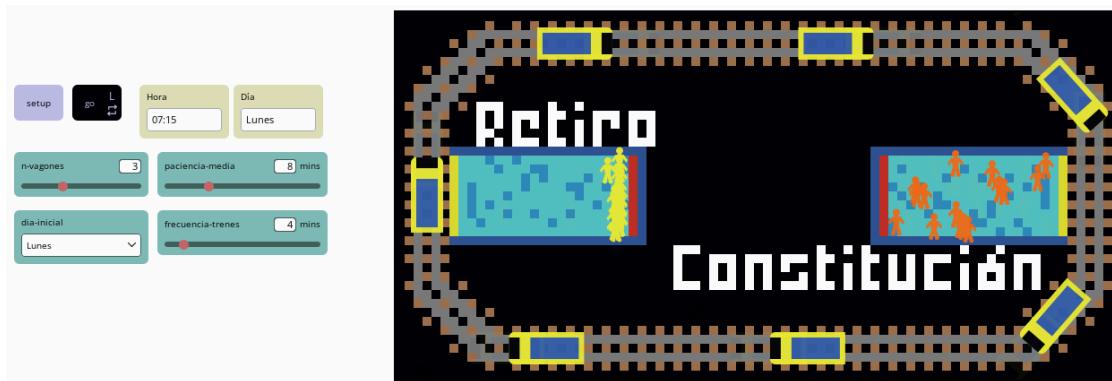


Figura 7. Interfaz de la simulación en NetLogo. A la izquierda, se observan los controles de la simulación. A la derecha, se observan los trenes y los pasajeros circulando por la Línea C del Subte de Buenos Aires.

La simulación se ejecuta en un entorno gráfico con coordenadas entre $(40, 20)$ y $(-40, -20)$, donde se representan las vías del Subte, trenes y pasajeros circulando por las mismas. La simulación cuenta con controles para iniciar, pausar y reiniciar la simulación, así como para ajustar parámetros como la cantidad de trenes en circulación y la velocidad de los mismos. Los dos botones más importantes son el de **setup** y el de **go**, que inicializan y ejecutan la simulación, respectivamente.

En el [Código 1](#) se observan las variables globales definidas en NetLogo. Con las constantes se parametriza el modelo. Estas son:

- **SEG-A-TICKS**: cantidad de segundos reales que representa un *tick* de la simulación, fijado en 1 tick cada segundo.
- **DIAS-HORA-INICIO** y **DIAS-HORA-FIN**: horarios de inicio y fin de operación del Subte, diferenciando entre días de semana y fines de semana [6].
- **TREN-V-MAX**: velocidad máxima de los trenes.
- **TREN-ACC**: aceleración de los trenes.
- **TREN-T-ESPERA**: tiempo que el tren permanece detenido en cada estación, fijado en un estimado de 60 s. Este tiempo se divide en dos partes iguales para el ascenso y descenso de pasajeros.
- **N-TRENES**: cantidad de formaciones en circulación.
- **ESTACIONES-DISTANCIA**: distancia entre estaciones, estimada en 4,5 km entre cada estación.
- **ESTACIONES-DEMANDA**: la frecuencia de pasajeros en cada estación, primero por día y luego por la cantidad de pasajeros cada 15 minutos. Estos datos se obtuvieron a partir del análisis de los histogramas de molinetes ([Figura 2](#) y [Figura 3](#)).

El resto de valores son inferidos (como **ESCALA**, que convierte las distancias reales en distancias gráficas) o son variables de estado de la simulación (como **dia-actual**, que indica el día actual de la semana).

Aparte de estas entradas, se miden distintas variables de salida para analizar el desempeño del sistema (obtenidas a partir de los *monitores* de NetLogo, como se ve en la [Figura 8](#)). Se mide la ocupación de cada estación y la cantidad de pasajeros cuya paciencia se agotó.

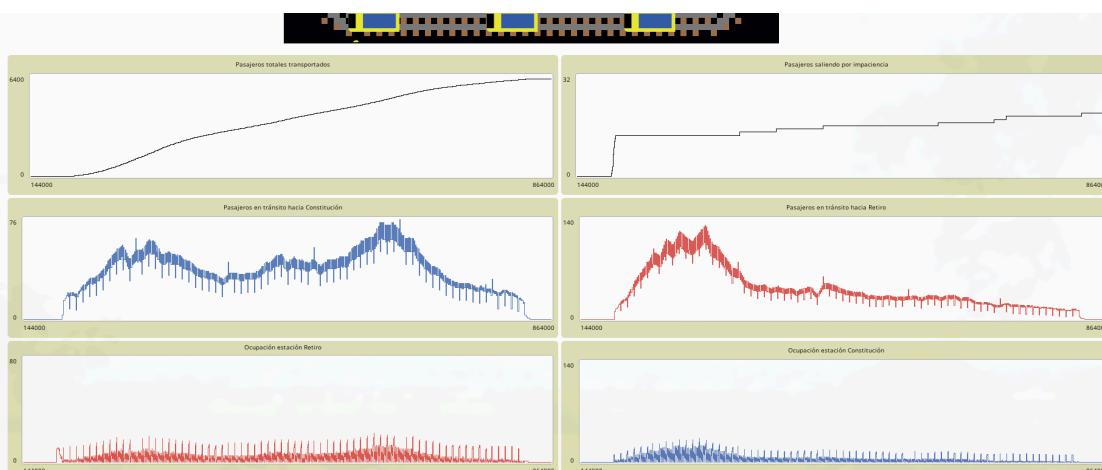


Figura 8. Monitores de la simulación en NetLogo.

simulacion.nlogox

```

3  globals [
4    ; Constantes
5    SEG-A-TICKS          ; Cantidad de ticks en un segundo
6    ESCALA                ; Cuántos metros mide un patch
7    DIAS-NOMBRES         ; Lista de nombres (strings) de los días de la semana
8    DIAS-HORA-INICIO     ; Hora de inicio (integer) del servicio de cada día de
  la semana
9    DIAS-HORA-FIN        ; Hora de fin (integer) del servicio de cada día de
  la semana
10   TREN-V-MAX           ; [m/s] Velocidad máxima de los trenes
11   TREN-ACC              ; [m/s2] Aceleración del tren
12   TREN-T-ESPERA        ; [s] Tiempo de espera en el andén. TREN-T-ESPERA/2 para
  descenso y TREN-T-ESPERA/2 para ascenso
13   TREN-T-VUELTA        ; [s] Tiempo total de un ciclo, desde que sale de un
  extremo hasta que vuelve.
14   N-TRENES              ; Cantidad de formaciones en circulación
15   ESTACIONES            ; Lista de estaciones (strings)
16   ESTACIONES-DISTANCIA  ; [m] Distancia entre estaciones
17   ESTACIONES-RECORRIDOS ; Tabla de puntos del recorrido hacia esa estación en
  cada estación
18   ESTACIONES-DEMANDAS   ; Tabla de matrices con la frecuencia de pasajeros en cada
  estación, primero por día y luego por la cantidad de pasajeros cada 15 minutos
19
20   ; Variables globales
21   dia-actual             ; Día actual de la semana, como índice de los arreglos
  DIAS-*
22   hora-actual            ; Entero entre 0 y 23
23   min-actual             ; Entero entre 0 y 59
24   seg-actual             ; Entero entre 0 y 59
25   n-dispatched          ; Tabla con la cantidad de pasajeros dispatched en cada
  estación en los últimos 15 min
...
30 ]

```

Código 1. Declaración de las variables globales en NetLogo. Se separan en constantes (parámetros del modelo) y variables (estado de la simulación).

3.2.1. Patches

En NetLogo, los *patches* son las celdas que componen el entorno gráfico. En este modelo, la mayoría de celdas son inocuas, salvo algunas que son utilizadas por las personas para ubicarse en las estaciones. Así, se definieron parámetros para cada *patch* en el [Código 2](#) que permiten identificar las paredes de las estaciones, los andenes y las salidas.

simulacion.nlogox

```

70  patches-own [
71    estacion            ; "RETIRO", "CONSTITUCIÓN" o "" (vacío)
72    tipo                ; "PARED", "SALIDA", "ANDÉN" o "" (vacío)
73 ]

```

Código 2. Declaración de variables de los *patches* de NetLogo.

3.2.2. Trenes (*Turtle*)

En NetLogo, los *turtles* son los agentes móviles. En este modelo, se definieron dos tipos de *turtles*: los trenes y las personas. Por un lado, los trenes tienen sus parámetros previamente descritos, así como variables internas para manejar su estado (como `estado` y `n-pasajeros`), definidas en el [Código 3](#).

`simulacion.nlogox`

```

41  trenes-own [
42    ; Parámetros fijos
43    max-pasajeros           ; Pasajeros máximos que soporta la formación
44    ; Variables de estado
45    estado                 ; Estado del tren ("ASCENSO", "EN_TRÁNSITO" o "DESCENSO")
46    ticks-estado           ; Cantidad de ticks desde que el tren entró al
      estado actual
47    direccion               ; Hacia dónde se dirige ("RETIRO" o "CONSTITUCIÓN")
48    n-pasajeros             ; Cantidad de pasajeros en los vagones
49    ; Si se encuentra "EN_TRÁNSITO"
50    recorrido               ; Cola de puntos para recorrer
51    v-actual                ; [m/s] Velocidad actual de la formación
52    d-recorrida             ; [m] Distancia ya recorrida
53    d-total                 ; [m] Distancia total del recorrido
54 ]

```

Código 3. Declaración de variables de los trenes de NetLogo.

La `direccion` del tren indica hacia dónde se mueve (hacia Constitución o hacia Retiro). Con `n-pasajeros` se lleva una cuenta de cuántos pasajeros hay en ese momento en el tren. `recorrido` es una lista de coordenadas que permiten el movimiento fluido del tren entre estaciones, junto a `v-actual`, `d-recorrida` y `d-total`.

Luego, cuando comienza el día, el `trenes-dispatcher` se encarga de crear los trenes necesarios y ubicarlos en la estación inicial. Para ello, se utiliza el código del [Código 4](#). Luego, los mismos desaparecen al finalizar el día.

`simulacion.nlogox`

```

240  to trenes-dispatcher
241    let direccion-inicial "CONSTITUCIÓN"
242    if hora-actual >= hora-inicio and hora-actual < hora-fin and count trenes < N-TRENES [
243      ; Al empezar el día, se van ingresando los trenes equiespaciadamente
244      let intervalo-salidas (TREN-T-VUELTA * SEG-A-TICKS) / N-TRENES
245      let proxima-salida intervalo-salidas * count trenes
246      if (ticks-desde hora-inicio 0) >= proxima-salida [
247        create-trenes 1 [
...
252        let recorrido-inicial table:get ESTACIONES-RECORRIDOS direccion-inicial
253        setxy (item 0 (item 0 recorrido-inicial)) (item 1 (item 0 recorrido-inicial))
254        facexy (item 0 (item 1 recorrido-inicial)) (item 1 (item 1 recorrido-inicial))
255

```

Código 4. Rutina que crea los trenes al inicio del día en NetLogo.

Finalmente, la máquina de estados se implementa como múltiples funciones dentro de la rutina `go` que se ejecuta en bucle, como se muestra en el [Código 5](#).

```
simulacion.nlogox
178  to go
179    tick
...
190    trenes-dispatcher
...
193    ask trenes [
194      set ticks-estado ticks-estado + 1
195      (ifelse
196        estado = "ASCENSO" [ update-tren-ascenso ]
197        estado = "EN_TRÁNSITO" [ update-tren-en-transito ]
198        estado = "DESCENSO" [ update-tren-descenso ]
199        [ error (word "Estado de tren inválido: \""
200          estado "\"") ]
200      )
201    ]
...
212 end
```

Código 5. Avance de las máquinas de estados de los trenes en NetLogo.

3.2.3. Pasajeros (*Turtle*)

Similarmente, los pasajeros tienen sus parámetros previamente descritos, así como variables internas para manejar su estado, definidas en el Código 6.

```
simulacion.nlogox
57  pasajeros-own [
58    ; Parámetros fijos
59    destino           ; A dónde quiere llegar ("RETIRO" o "CONSTITUCIÓN")
60    t-paciencia       ; [s] Paciencia máxima del pasajero
61    ; Variables de estado
62    estado            ; Estado del pasajero ("ESPERANDO", "SUBIENDO", "VIAJANDO"
63    o "SALIENDO")
63    ticks-estado       ; Cantidad de ticks desde que el pasajero entró al
64    estado actual
64    t-esperando        ; [s] Cuánto tiempo estuvo esperando en la estación
65  ]
```

Código 6. Declaración de variables de los pasajeros de NetLogo.

El parámetro `t-paciencia` es aleatorio para cada pasajero, siguiendo una distribución normal con media definida por un *slider* (por defecto 8 min) y desviación estándar de 1,5 min. La `dirección` indica hacia dónde se dirige el pasajero (hacia Constitución o hacia Retiro). `t-esperando` lleva la cuenta del tiempo que el pasajero lleva esperando en el andén.

El comportamiento de estos es bastante más simple que el de los trenes, porque se mueven sin rumbo por las estaciones hasta que llegue algún tren. Luego, intentan subirse. Si logran subirse, se crea un *link* entre el pasajero y el tren para que el pasajero pueda “saber” cuándo llegó a su destino. Finalmente, descienden y se dirigen a la salida y se elimina el *link*.

Los pasajeros aparecen en las estaciones según la demanda de pasajeros, implementada en el Código 7. La misma lee los histogramas de molinetes y crea pasajeros acorde a la frecuencia de llegada de los mismos.

Finalmente, la máquina de estados se implementa como múltiples funciones dentro de la rutina `go` que se ejecuta en bucle, como se muestra en el Código 8.

```

simulacion.nlogox

351  to pasajeros-dispatcher
352    if min-actual mod 15 = 0 and seg-actual = 0 [
353      foreach ESTACIONES [ est -> table:put n-dispatched est 0 ]
354    ]
355    if hora-actual >= hora-inicio and hora-actual < hora-fin [
356      let min-intervalo int (min-actual / 15)
357      let intervalo-idx (hora-actual - hora-inicio) * 4 + min-intervalo
358      foreach ESTACIONES [ est ->
359        let curva table:get ESTACIONES-DEMANDAS est
360        let p-totales array:item (array:item curva dia-actual) intervalo-idx
361        let p-actuales table:get n-dispatched est
362        if p-totales > 0 [
363          let proxima-llegada (15 * 60 * SEG-A-TICKS) * (p-actuales / p-totales)
364          if (ticks-desde (hora-actual) (min-intervalo * 15)) >= proxima-llegada [
365            create-pasajeros 1 [
366              ...
367            ...
368            table:put n-dispatched est p-actuales + 1
369          ]
370        ]
371      ]
372    ]
373  ]
374 end

```

Código 7. Rutina que simula la demanda de pasajeros en NetLogo.

```

simulacion.nlogox

178  to go
179    tick
...
191    pasajeros-dispatcher
...
202    ask pasajeros [
203      set ticks-estado ticks-estado + 1
204      (ifelse
205        estado = "ESPERANDO" [ update-pasajero Esperando ]
206        estado = "SUBIENDO" [ update-pasajero Subiendo ]
207        estado = "VIAJANDO" [ update-pasajero Viajando ]
208        estado = "SALIENDO" [ update-pasajero Saliendo ]
209        [ error (word "Estado de pasajero inválido: \""
210          estado "\"") ]
211      )
212    end

```

Código 8. Avance de las máquinas de estados de los pasajeros en NetLogo.

3.2.4. Monitors

Para variar los parámetros del modelo, se utilizan *sliders* y *choosers* en la interfaz gráfica de NetLogo, como se ve en la Figura 9. Estos permiten modificar las constantes definidas en el Código 1.

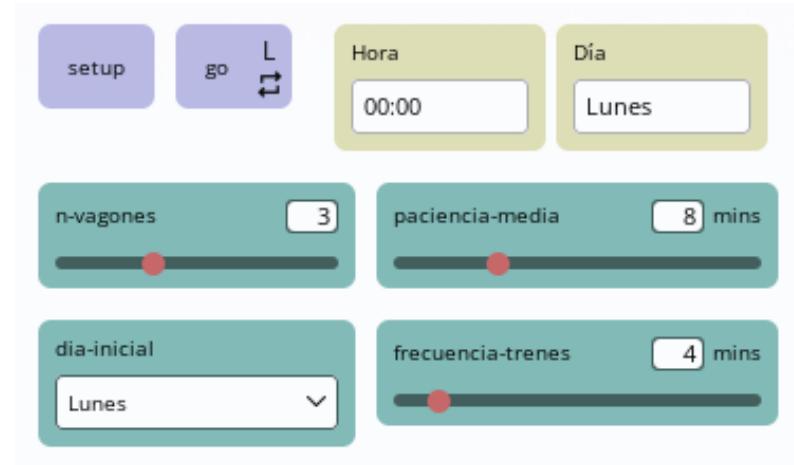


Figura 9. Controles y monitores de la simulación.

Con **n-vagones** se cambia la cantidad de vagones por formación — cada vagón cuenta con una capacidad máxima de 20 pasajeros. Con **paciencia-media** se cambia la paciencia media de los pasajeros — la desviación estándar se mantiene fija en 1,5 min. Con **dia-inicial** se puede elegir el día de la semana en que comienza la simulación. Finalmente, con **frecuencia-trenes** se puede elegir la frecuencia de los trenes en minutos, internamente se calcula la cantidad de trenes necesarios para mantener esa frecuencia, como se ve en el [Código 9](#).

simulacion.nlogo

```
98  set TREN-T-VUELTA 2 * (t-tren-recorrido + TREN-T-ESPERA) ; [s]
99  set N-TRENES round(TREN-T-VUELTA / (frecuencia-trenes * 60))
```

Código 9. Cómputo de la cantidad de trenes necesarios según la frecuencia deseada en NetLogo. La duración del recorrido del tren se calcula considerando una aceleración y desaceleración uniforme.

Luego, se cuenta con dos monitores que muestran la hora actual de la simulación y el día de la semana actual. Más abajo se encuentran *plots* que grafican distintas variables de la simulación ([Figura 8](#)). Para cada estación, se grafica la ocupación de la misma en función del tiempo. Además, se grafica la cantidad de pasajeros viajando en algún tren según su destino. Finalmente, se grafican dos funciones acumuladoras: una que cuenta la cantidad de pasajeros que lograron llegar a su destino y otra que cuenta la cantidad de pasajeros que desistieron de esperar.

3.3. Replicación de experimentos

El código fuente de la simulación se encuentra disponible en el repositorio de GitHub <https://github.com/JuanM04/str-sim>. Para ejecutar la simulación, basta con descargar el repositorio y abrir el archivo **simulacion.nlogo** con NetLogo 7.0.0 o superior. Nótese que es necesario descargar todo el repositorio, ya que la simulación depende de archivos externos (los histogramas de molinetes). En ese repositorio también se encuentran archivos externos a la simulación en sí, como

- el código fuente de este informe en **informe/**,
- los datasets procesados en **datasets/** junto a el script que los generó
- y las simulaciones realizadas en **simulaciones/**.

4. Resultados

Sobre la simulación desarrollada, se realizaron múltiples experimentos variando los parámetros del modelo. En particular, se variaron la cantidad de vagones por formación y la frecuencia de los trenes. Para estos experimentos, se fijó la paciencia media de los pasajeros en el doble de la frecuencia de los trenes, para simular que los pasajeros están dispuestos a esperar entre uno y dos trenes antes de desistir.

Como base, se tomaron frecuencias de 3, 5 y 7 minutos para los días de semana, sábados y domingos, respectivamente [6]. Sobre estas frecuencias, se variaron la cantidad de vagones por formación entre 1 y 3 vagones (cada uno con capacidad para 20 pasajeros). Estos resultados se resumen en la [Tabla 1](#).

Tabla 1. Configuraciones base de la simulación con frecuencias estándar del Subte.

Día	Frecuencia	Trenes	Vagones	Pas. transportados	Pas. que desistieron	Simulación
Lunes	3 min	10	1	4870	360	Figura 10
			2	4891	14	Figura 11
			3	4893	12	Figura 12
Sábado	5 min	6	1	2711	22	Figura 13
			2	2717	16	Figura 14
Domingo	7 min	4	1	1918	0	Figura 15

Puede apreciarse que para los días de semana, con coches de un solo vagón se quedan cortos en capacidad, ya que hay una cantidad considerable de pasajeros que desisten de esperar. Al agregar un vagón más, la cantidad de pasajeros transportados aumenta ligeramente, pero la cantidad de pasajeros que desisten disminuye drásticamente. Finalmente, al agregar un tercer vagón, la cantidad de pasajeros transportados y que desisten se mantiene casi constante.

Un análisis semejante se realizó con los días sábados y domingos, llegando a conclusiones similares. Para los sábados, basta con formaciones de un vagón porque formaciones de dos vagones no presentan una mejora. Es interesante el caso de los domingos, con formaciones de un solo vagón son suficientes para transportar a todos los pasajeros sin que nadie desista.

4.1. Optimizaciones

A partir de los resultados base, se investigó optimizar el sistema. Comenzando por un día de semana, se probó con reducir la frecuencia de los trenes a 5 minutos (un minuto menos de la paciencia media estimada). Dejando la cantidad de vagones en 2, se obtuvieron los resultados de la [Figura 16](#). Al disminuir la frecuencia, la cantidad de pasajeros transportados disminuyó considerablemente ($4891 \rightarrow 3037$) y los pasajeros que desistieron aumentaron ($14 \rightarrow 1868$). Para solventar eso, se agregó un vagón más por formación, obteniendo los resultados de la [Figura 17](#). Con esta configuración, se obtuvieron solo 143 pasajeros que desistieron, transportando un total de 4762 pasajeros.

Esto resulta en una optimización considerable, ya que se logró transportar casi la misma cantidad de pasajeros que con la configuración base (frecuencia de 3 minutos y formaciones de 2 vagones) pero con menos trenes en circulación (6 en lugar de 10). Esto implica un ahorro considerable en costos operativos, ya que se utilizan menos recursos para brindar un servicio similar.

De manera similar se optimizaron los fines de semana, llegando a las conclusiones de la [Tabla 2](#).

Tabla 2. Configuraciones base y optimizadas para cada día.

Día	Frecuencia	Trenes	Vagones	Pas. transportados	Pas. que desistieron	Simulación
Lunes	3 min	10	2	4891	14	Figura 11
	5 min	6	3	4762	143	Figura 17
Sábado	5 min	6	1	2711	22	Figura 13
	9 min	3	2	2607	126	Figura 19
Domingo	7 min	4	1	1918	0	Figura 15
	11 min	3	1	1915	3	Figura 20

Nótese que para el caso de los domingos, se probó reducir la frecuencia a 13 minutos, pero con esta configuración no se logró reducir la cantidad de pasajeros que desistieron de 100. Esto no resultaba de un problema en los otros días, pero como los domingos viajan alrededor de 2000 pasajeros, esto resultaba en un porcentaje elevado de pasajeros que desistían (5%). Por ello, se redujo la frecuencia hasta lograr un resultado más razonable.

5. Conclusiones

El modelo desarrollado permitió simular el funcionamiento de la Línea C del Subte de Buenos Aires bajo distintos parámetros. El mismo fue propuesto para estudiar la sincronización entre los trenes del Subte y los ferrocarriles metropolitanos en las cabeceras de Retiro y Constitución. A partir de estos datos, se logró estimar la cantidad de trenes necesarios para satisfacer la demanda de pasajeros en distintos días de la semana. Además, se logró optimizar el sistema reduciendo la cantidad de trenes en circulación sin afectar significativamente la cantidad de pasajeros transportados.

Estos descubrimientos resultan valiosos para la planificación del servicio de la Línea C y pueden ser extendidos a otras líneas del Subte o a otros sistemas de transporte público. Futuras investigaciones podrían explorar la inclusión de estaciones intermedias, variaciones en la demanda a lo largo del año o la incorporación de eventos especiales que afecten el flujo de pasajeros.

Bibliografía

- [1] Instituto de Estadística y Censos de la Ciudad Autónoma de Buenos Aires, *Pasajeras y pasajeros pagos transportados por línea de subterráneo y premetro. Ciudad de Buenos Aires*. (2024). [En línea]. Disponible en: <https://www.estadisticaciudad.gob.ar/eyc/banco-datos/pasajeras-y-pasajeros-pagos-transportados-por-linea-de-subterraneo-y-premetro-ciudad-de-buenos-aires-enero-de-2010-junio-de-2024/>
- [2] Jefatura de Gabinete de Ministros. Secretaría de Transporte y Obras Públicas. Subterráneos Buenos Aires (SBASE), *Subte: Viajes Molinetes*. (2024). [En línea]. Disponible en: <https://data.buenosaires.gob.ar/es/dataset/subte-viajes-molinetes>
- [3] S. F. Railsback y V. Grimm, *Agent-Based and Individual-Based Modeling*. Princeton University Press, 2019.
- [4] U. Wilensky y W. Rand, *An Introduction to Agent-Based Modeling*. MIT Press, 2015.
- [5] U. Wilensky, «NetLogo», *Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL*, 1999, [En línea]. Disponible en: <https://ccl.northwestern.edu/netlogo/>
- [6] Emova, «Horarios del servicio de la red de Subte». Accedido: 17 de diciembre de 2025. [En línea]. Disponible en: <https://web.archive.org/web/20251217235559/https://emova.com.ar/index.php/horarios-del-servicio/>

A. Simulaciones base

En este apéndice se muestran los resultados completos de las simulaciones realizadas para distintos parámetros del modelo para lograr una configuración base.

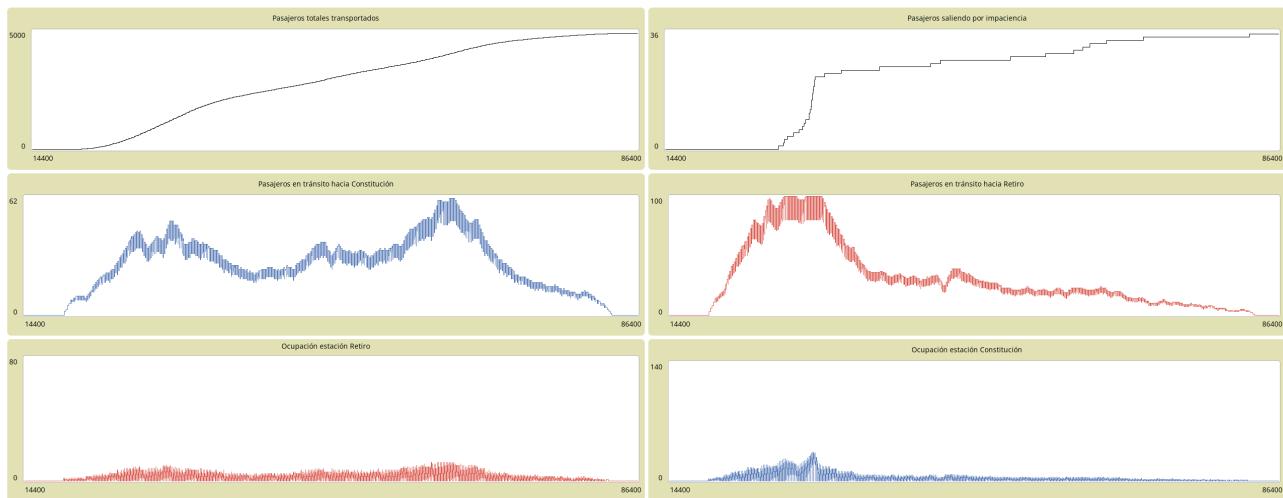


Figura 10. Resultados de la simulación un lunes con trenes cada 3 minutos, formaciones de 1 vagón y una paciencia media de 6 minutos.

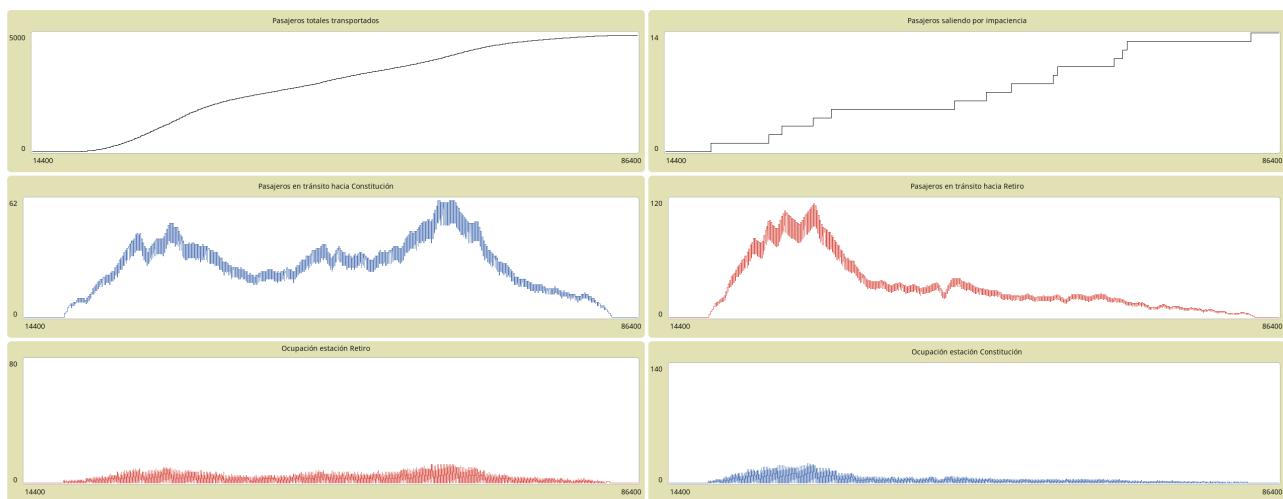


Figura 11. Resultados de la simulación un lunes con trenes cada 3 minutos, formaciones de 2 vagones y una paciencia media de 6 minutos.

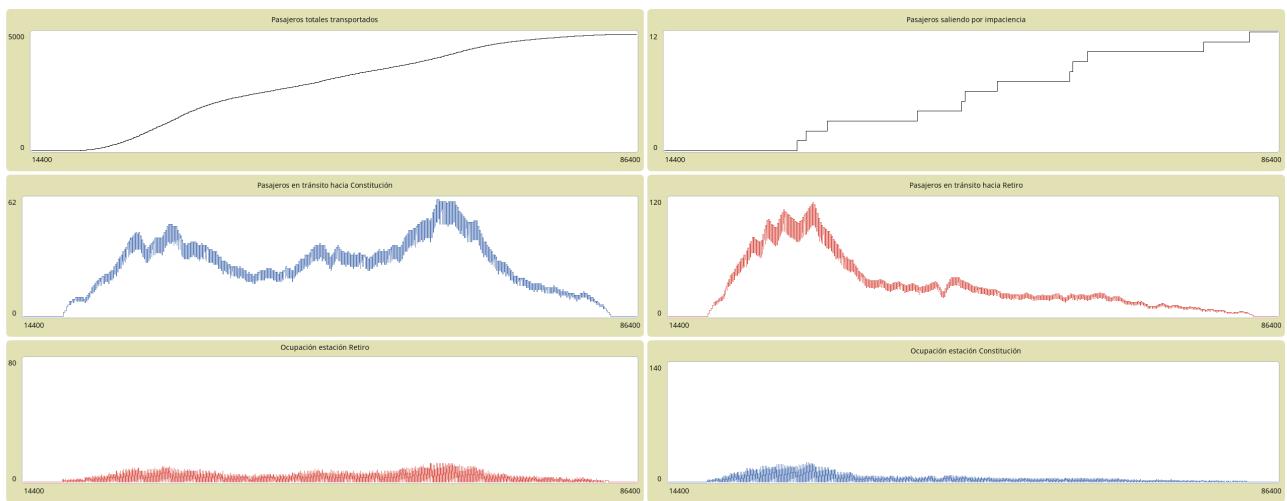


Figura 12. Resultados de la simulación un lunes con trenes cada 3 minutos, formaciones de 3 vagones y una paciencia media de 6 minutos.

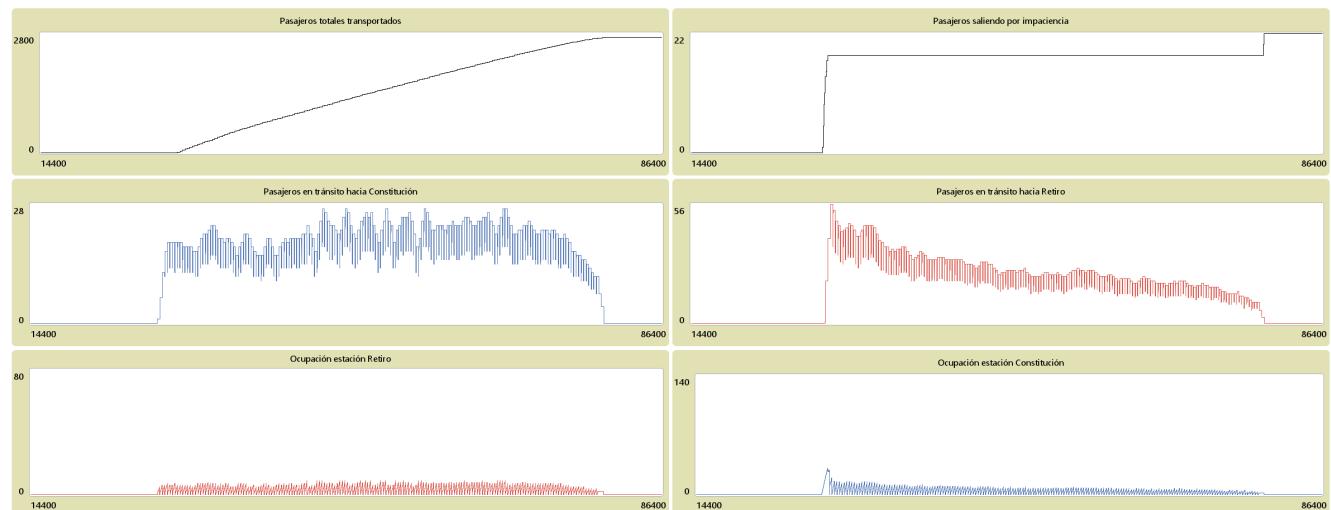


Figura 13. Resultados de la simulación un sábado con trenes cada 5 minutos, formaciones de 1 vagón y una paciencia media de 10 minutos.

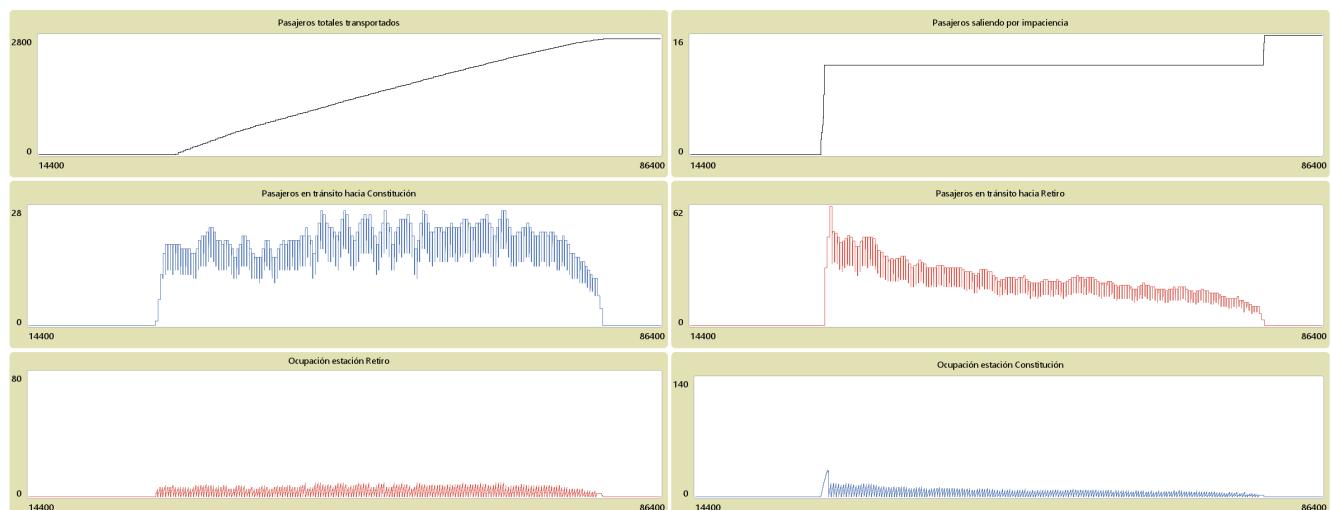


Figura 14. Resultados de la simulación un sábado con trenes cada 5 minutos, formaciones de 2 vagones y una paciencia media de 10 minutos.

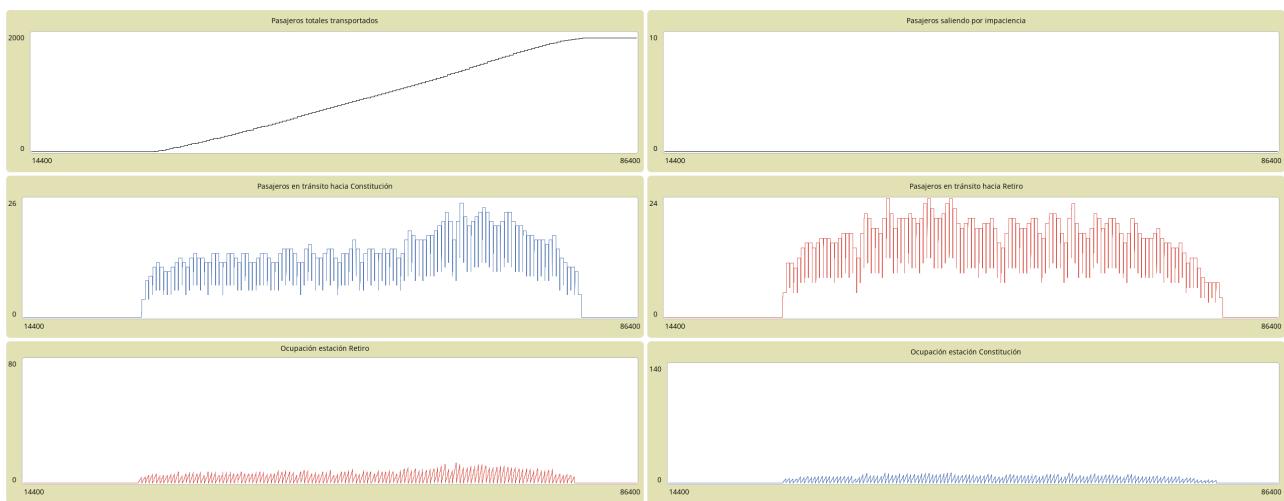


Figura 15. Resultados de la simulación un domingo con trenes cada 7 minutos, formaciones de 1 vagón y una paciencia media de 14 minutos.

B. Simulaciones optimizadas

En este apéndice se muestran los resultados completos de las simulaciones realizadas para distintos parámetros del modelo para lograr una configuración optimizada.

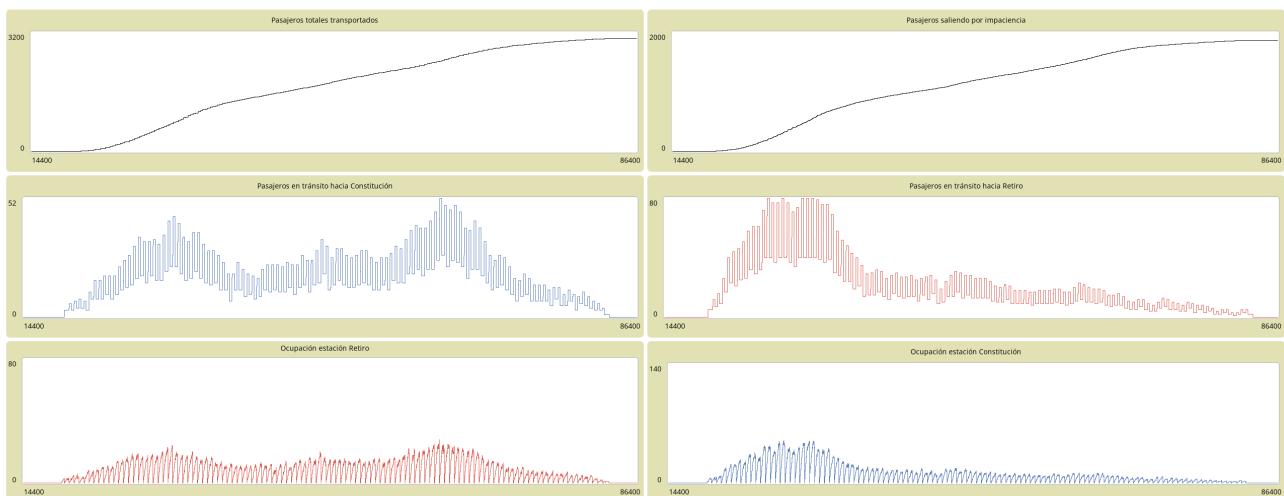


Figura 16. Resultados de la simulación un lunes con trenes cada 5 minutos, formaciones de 2 vagones y una paciencia media de 6 minutos.

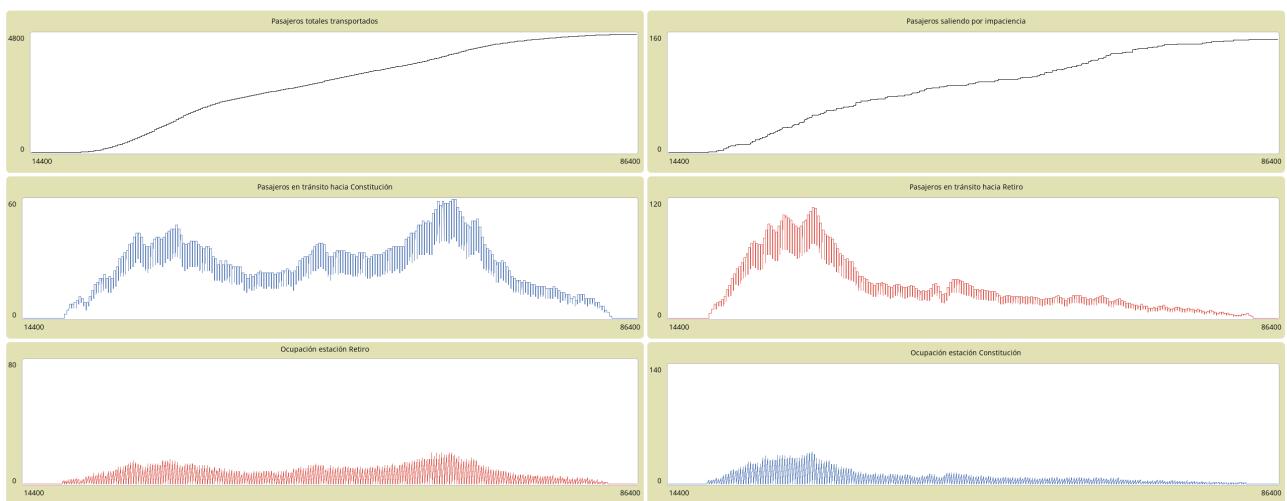


Figura 17. Resultados de la simulación un lunes con trenes cada 5 minutos, formaciones de 3 vagones y una paciencia media de 6 minutos.

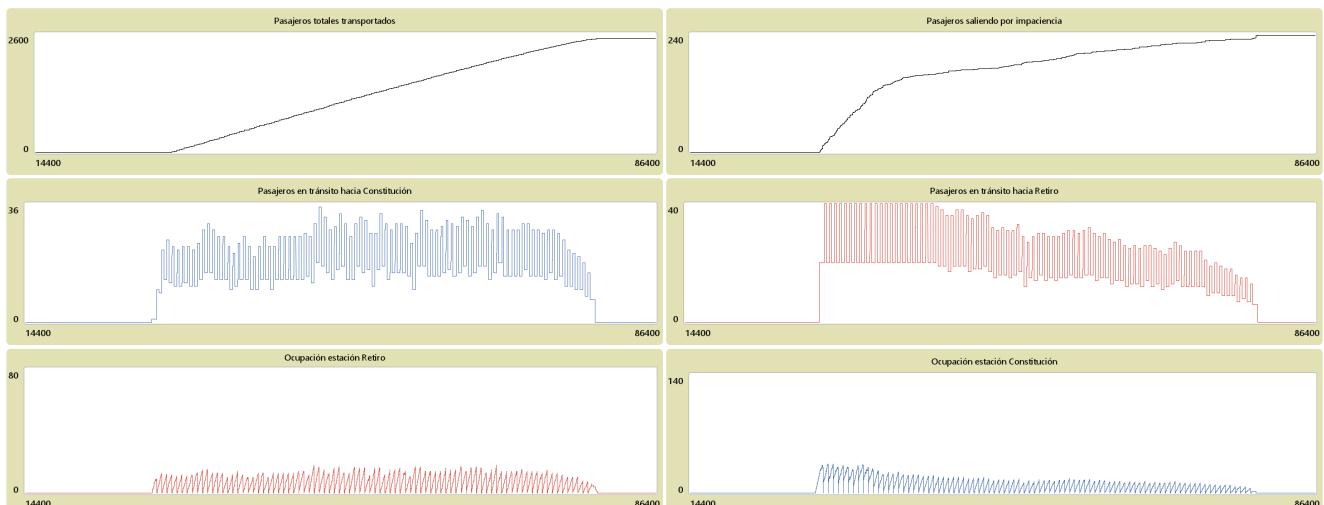


Figura 18. Resultados de la simulación un sábado con trenes cada 9 minutos, formaciones de 1 vagón y una paciencia media de 10 minutos.

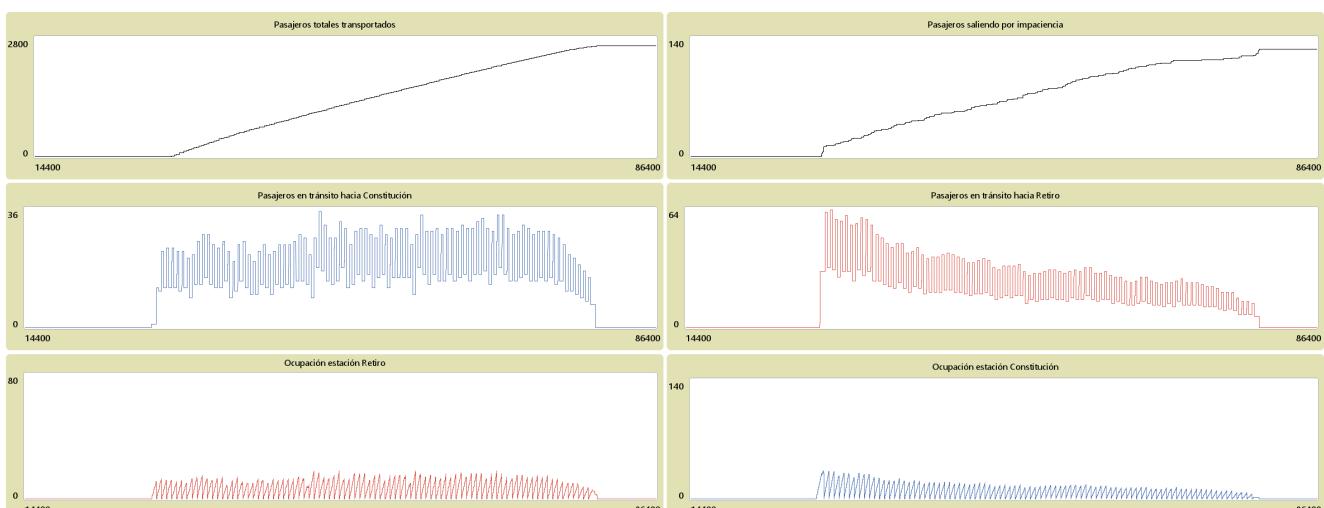


Figura 19. Resultados de la simulación un sábado con trenes cada 9 minutos, formaciones de 2 vagones y una paciencia media de 10 minutos.

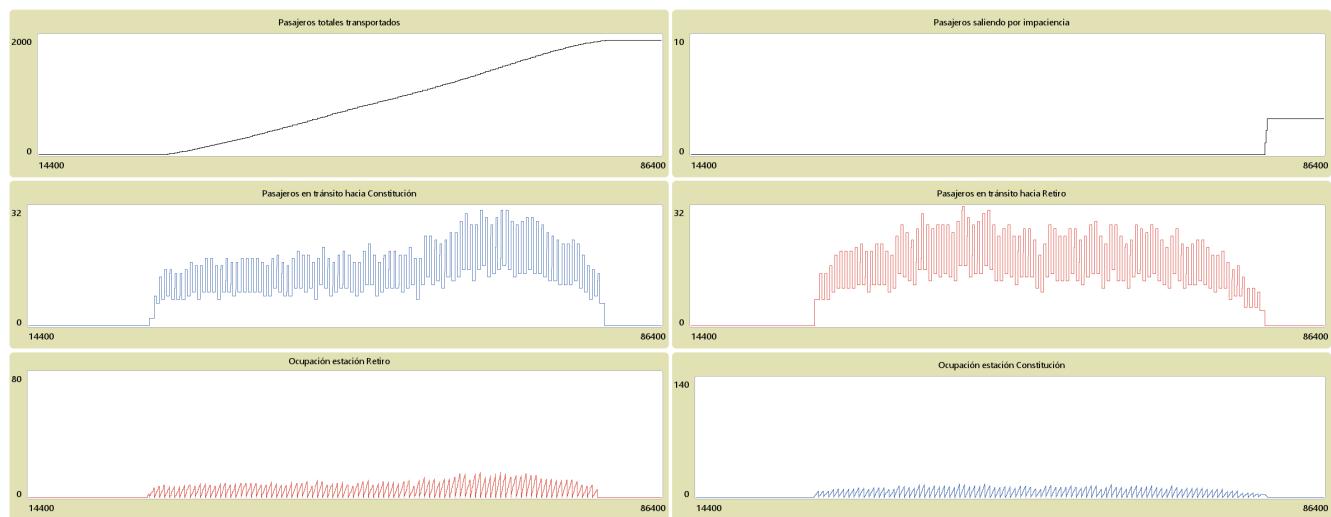


Figura 20. Resultados de la simulación un domingo con trenes cada 11 minutos, formaciones de 1 vagón y una paciencia media de 14 minutos.