



SIMULACIÓN DE UNA ESTACIÓN FERROVIARIA

GRUPO N°05 – SISTEMAS DE TIEMPO REAL

REPASO DEL PROYECTO

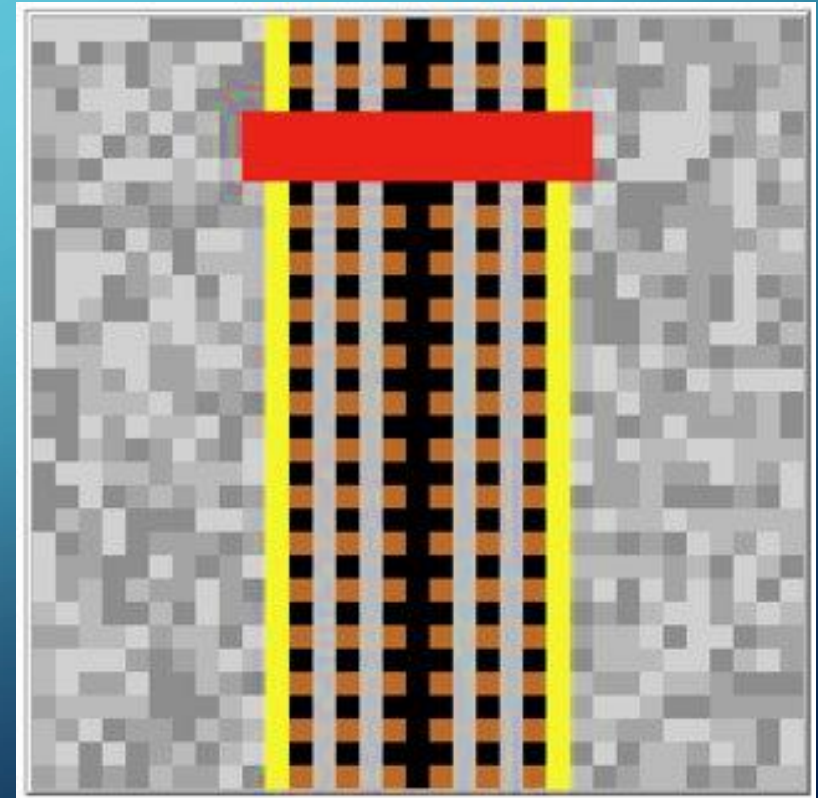


- Estación Berazategui
- Simulación con datos de 2019
- Agente Pasajero-Tren



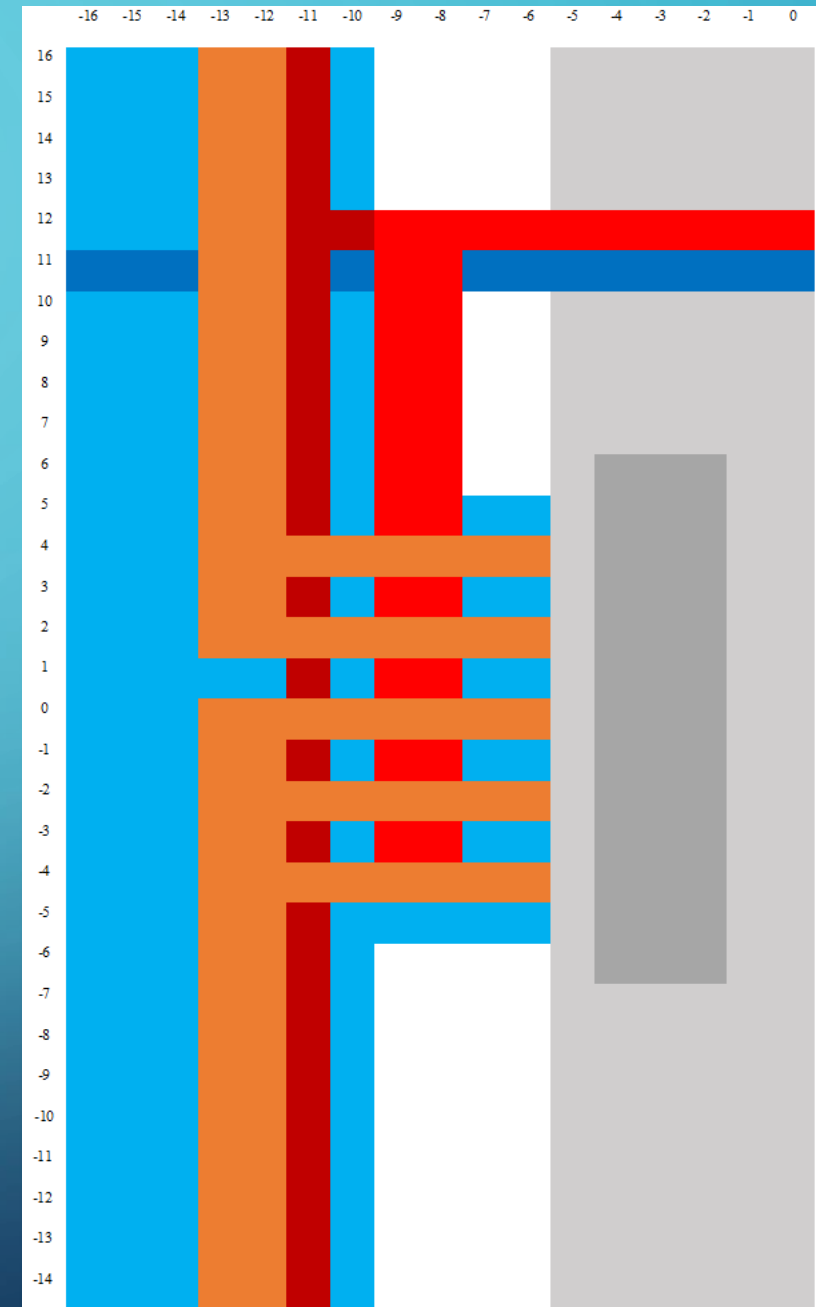
ZONAS DEFINIDAS

- Variables de Patches (celdas):
 - Elevación: 0, 1, 2
 - Andén: 0, 1
 - Lugar de espera: 0, 1
 - ¿Está Libre?: true, false



ZONAS DEFINIDAS

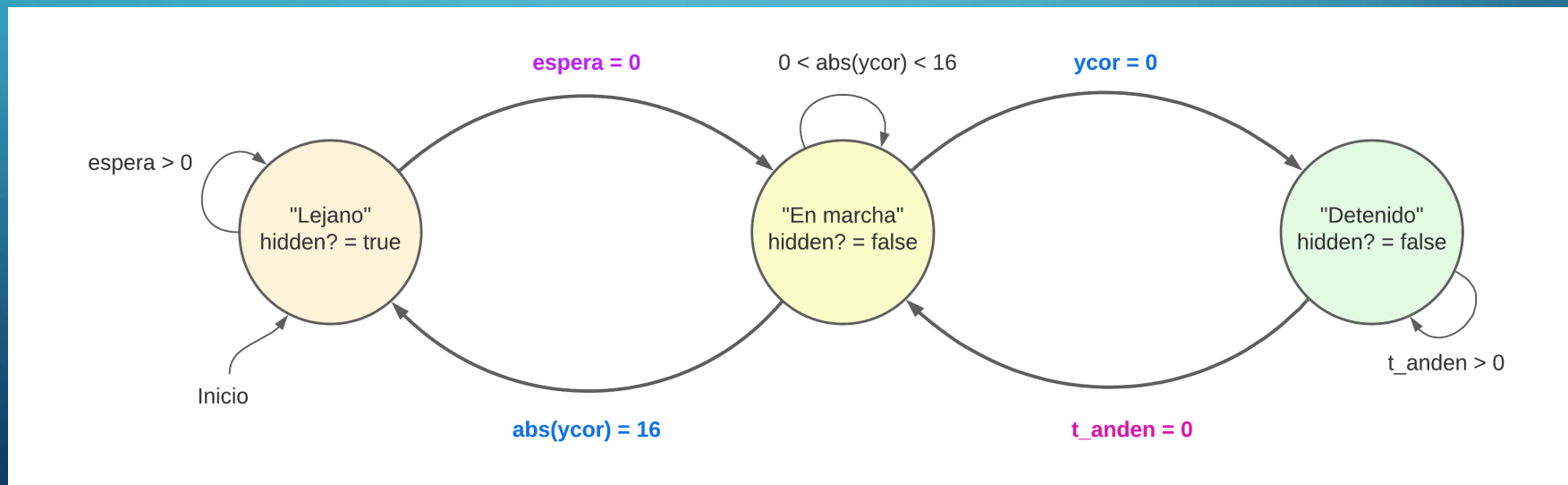
- Los que bajan del tren y se dirigen a un molinete
- Los que bajan del tren y deben cruzar el puente
- Los que ingresan por molinete y deben cruzar el puente
- Los que deben subir al tren
- Los que vienen del andén contrario para subir al tren



AGENTE TREN



- 3 estados: en marcha, detenido, lejano.
- 2 atributos agregados:
 - espera: tiempo hasta próximo arribo
 - t_anden: tiempo hasta volver a avanzar



AGENTE TREN



- Tabla de estados resumida:

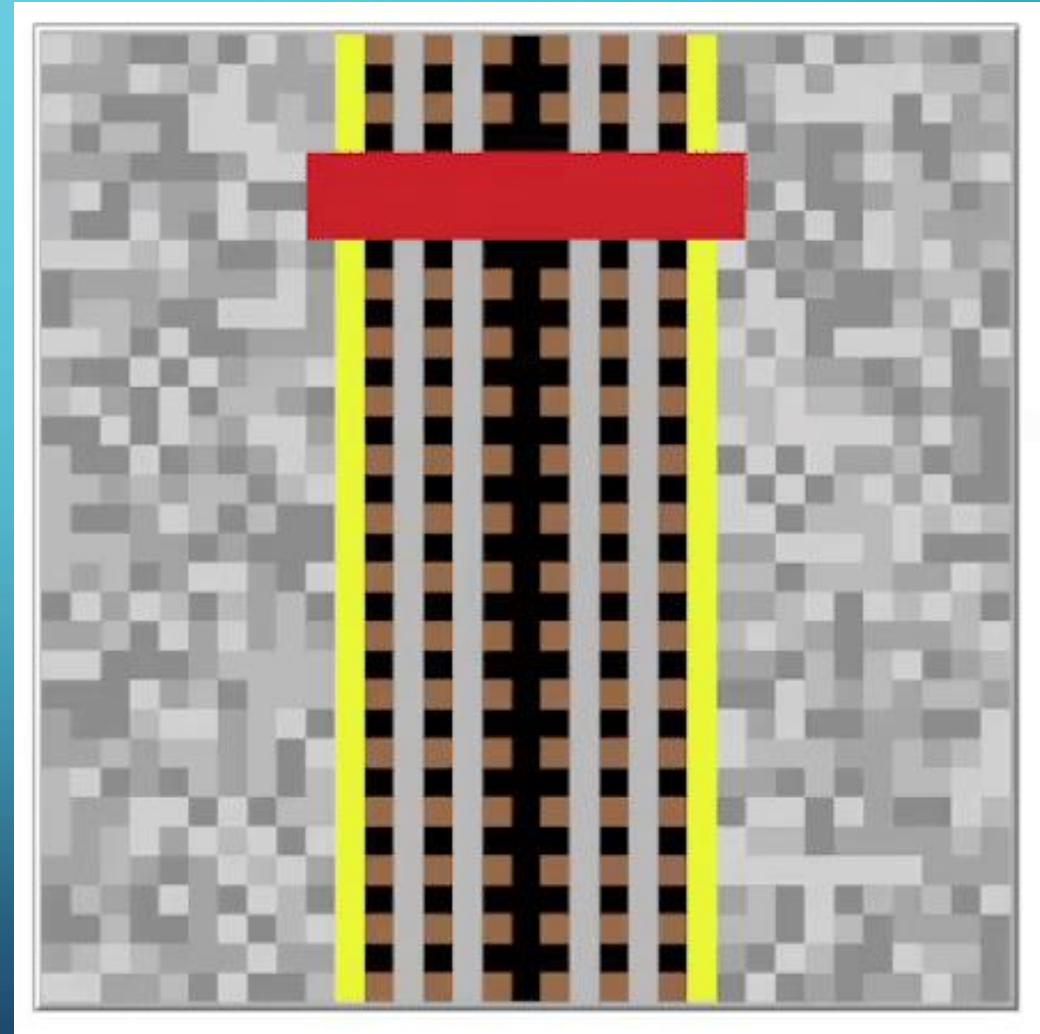
ESTADO ACTUAL	ENTRADAS POS	ENTRADAS ATRIBUTOS			ENTRADAS GLOBALES		ESTADO SIGUIENTE	ACCION
	ycor	t_anden	espera	who	pasajeros rest_izquierda	pasajeros rest_derecha		
YENDO	$abs() < 0.01$	X	X	X	X	X	EN ANDEN	Espera tiempo en anden
	$abs(ycor) \geq \max-pycor$	X	X	X	X	X	LEJANO	hidden? = true hasta próx. arribo
EN ANDÉN	X	X	X	0	0	X	EN ANDEN	Decrementar t_anden
	X	X	X	1	X	0	EN ANDEN	Decrementar t_anden
	X	0	X	X	X	X	YENDO	Empieza a avanzar
LEJANO	X	X	0	X	X	X	YENDO	Aparece en la estacion del lado adecuado

AGENTE PASAJERO



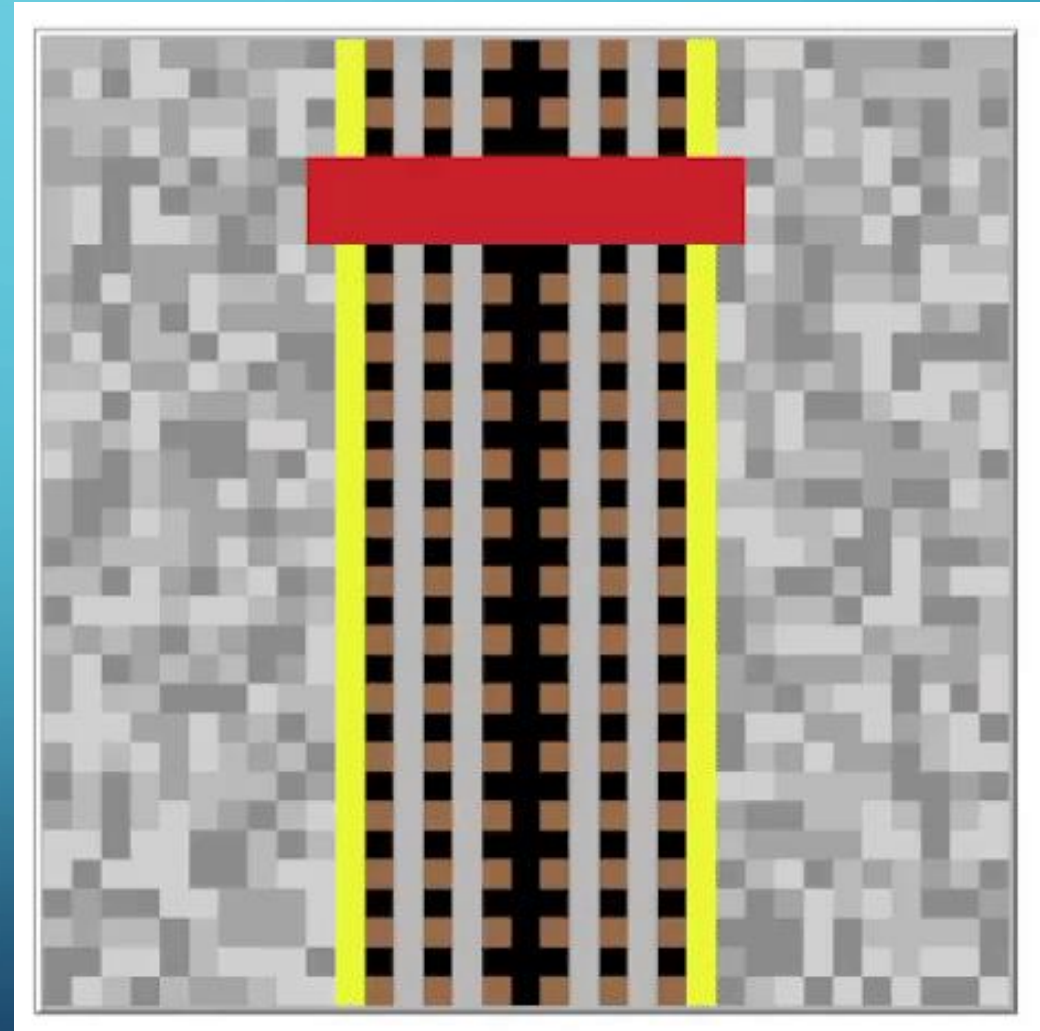
AGENTE PASAJERO

- Estado “Saliendo” (de la estación)
- Es creado en un punto próximo al tren ($x = -6$ o $x = 6$)
- Avanza hacia un molinete
- Es eliminado al llegar al mismo



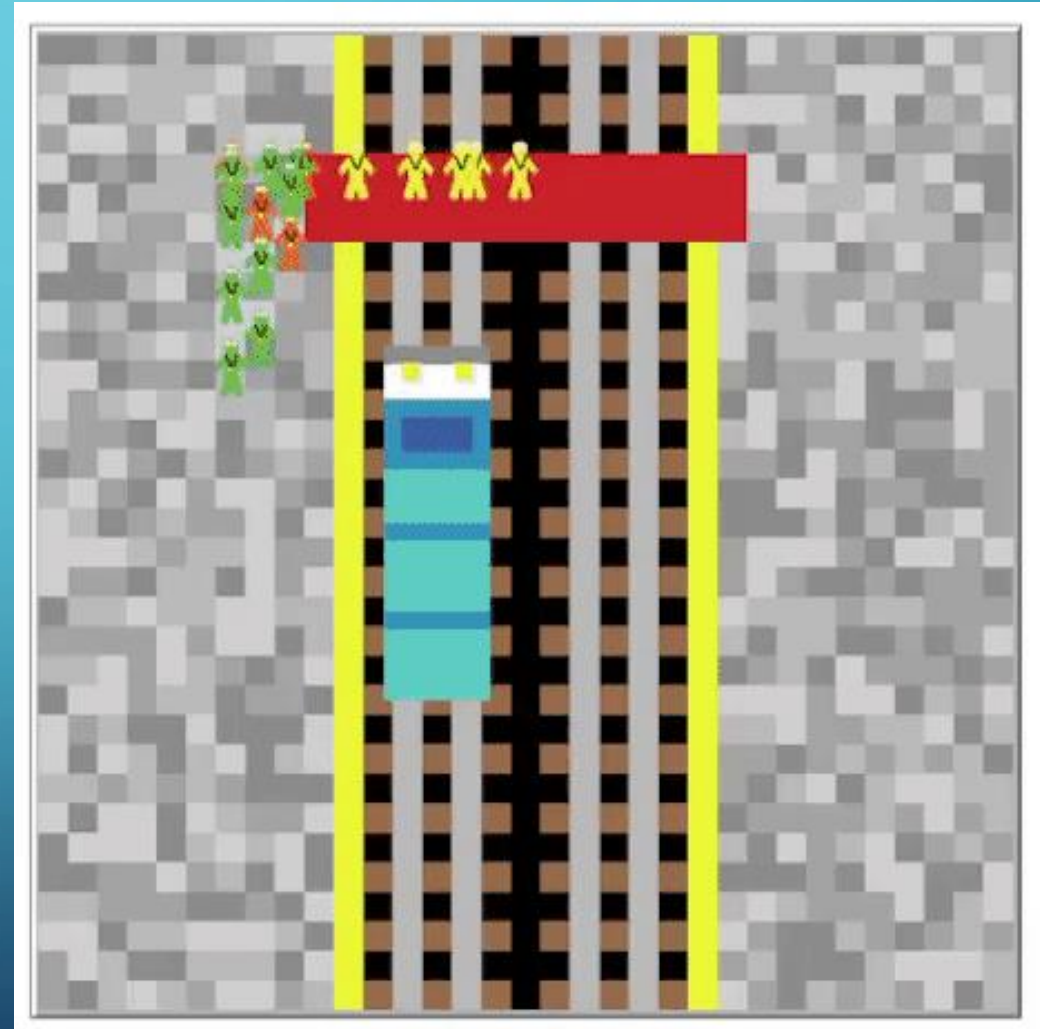
AGENTE PASAJERO

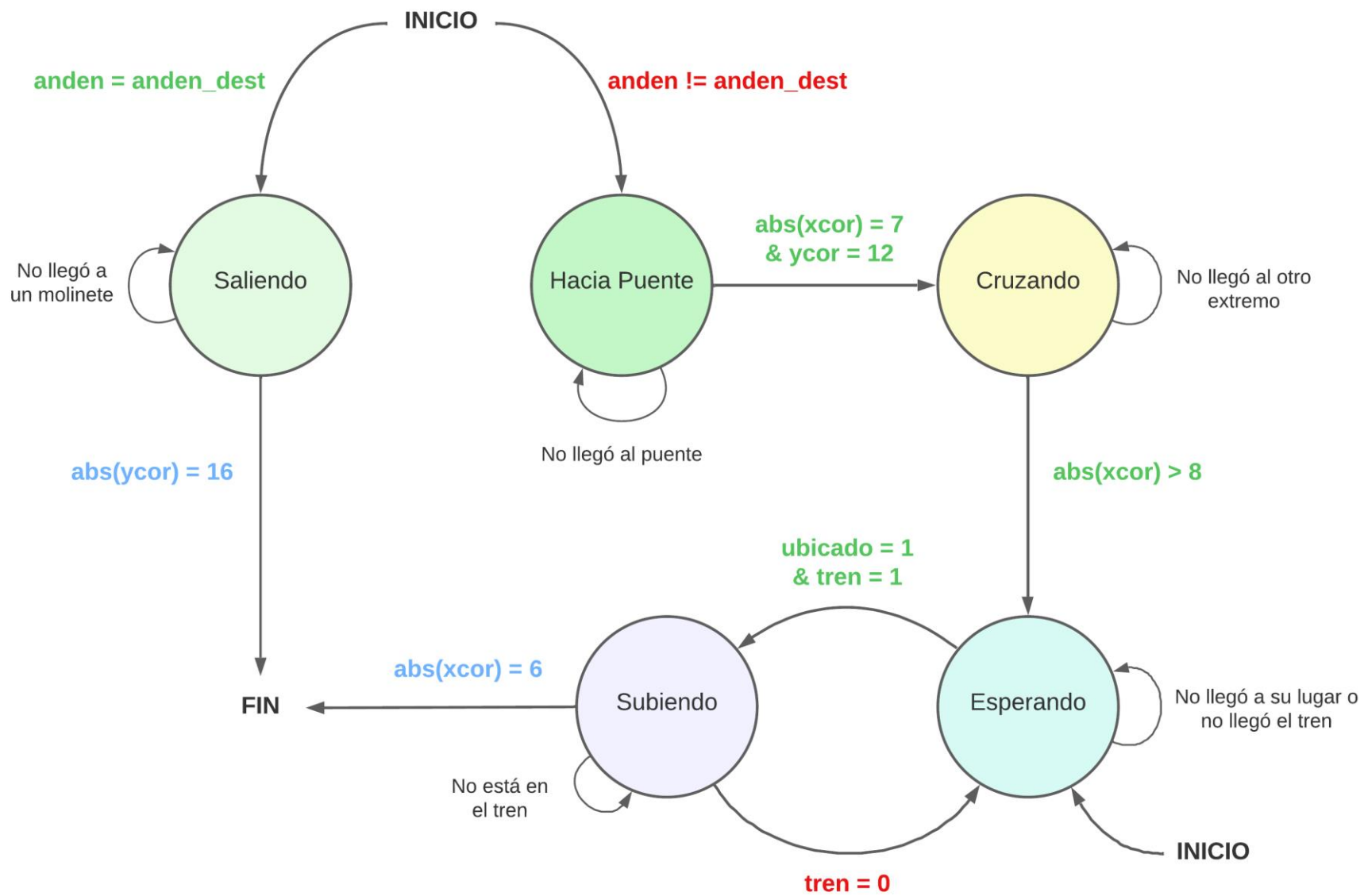
- Estado "Hacia Puente"
 - Creado desde tren o molinete
 - Reserva su lugar para andén destino
 - Avanza hacia la base del puente
-
- Estado "Cruzando" (en amarillo)
 - Avanza hacia el andén destino.



AGENTE PASAJERO

- Estado "Esperando"
- Luego de cruzar puente, o bien, creado desde un molinete.
- Se ubica en el lugar de espera.
- Estado "Subiendo"
- Se acerca al tren e intenta subir





AGENTE PASAJERO

Tren = el tren con “who = anden” se encuentra en estado “detenido” (en andén)

Faltan bajar = el contador global de pasajeros restantes es mayor a 0, el que corresponda.

ESTADO ACTUAL	ENTRADAS POSICIÓN		ENTRADAS GLOBALES		ESTADO SIGUIENTE	ACCION
	x_esperada	y_esperada	tren?	faltan_bajar?		
SALIENDO	0	(x)	(x)	(x)	Saliendo	Acercarse horizontalmente
	1	0	(x)	(x)	Saliendo	Avanzar hacia molinete
	1	1	(x)	(x)	-	die
HACIA PUENTE	0	(x)	(x)	(x)	Hacia Puente	Acercarse horizontalmente
	1	0	(x)	(x)	Hacia Puente	Avanzar hacia escalera
	1	1	(x)	(x)	Cruzando	Girar hacia andén destino
CRUZANDO	0	(x)	(x)	(x)	Cruzando	Seguir avanzando
	1	(x)	(x)	(x)	Esperando	Girar hacia el sur
ESPERANDO	0	(x)	(x)	(x)	Esperando	Acercarse horizontalmente
	1	0	(x)	(x)	Esperando	Avanzar hacia lugar de espera
	1	1	0	(x)	Esperando	-
	1	1	1	(x)	Subiendo	Girar hacia el tren
SUBIENDO	0	(x)	0	(x)	Esperando	Girar hacia lugar de espera
	0	(x)	1	0	Subiendo	Acercarse al tren
	0	(x)	1	1	Subiendo	-
	1	(x)	(x)	(x)	-	die

```
;; para cada tick del sistema  
to go
```

```
let t (ticks mod 8)
```

```
(ifelse
```

```
  ;; actualización de trenes según su estado
```

```
  t = 0 [ ask trenes with [estado = 0] [update_tren_en_marcha] ]
```

```
  t = 1 [ ask trenes with [estado = 1] [update_tren_detenido] ]
```

```
  t = 2 [ ask trenes with [estado = 2] [update_tren_lejano] ]
```

```
  ;; actualización de pasajeros según su estado
```

```
  t = 3 [ ask pasajeros with [estado = 10] [update_pasajero_saliendo] ]
```

```
  t = 4 [ ask pasajeros with [estado = 11] [update_pasajero_hacia_puente] ]
```

```
  t = 5 [ ask pasajeros with [estado = 12] [update_pasajero_cruzando] ]
```

```
  t = 6 [ ask pasajeros with [estado = 13] [update_pasajero_esperando] ]
```

```
  t = 7 [ ask pasajeros with [estado = 14] [update_pasajero_subiendo] ]
```

```
)
```

```
;; update 4/11 - creación de pasajeros que bajan del tren
```

```
;; se crean cada 50 ticks = 50/1000 min = 0.05 min = 3 segundos
```

```
if (ticks mod 50 = 0 and pasajeros_restantes_izq > 0) [ create-pasajeros 1 [crear_pasajero_izq] ]
```

```
if (ticks mod 50 = 25 and pasajeros_restantes_der > 0) [ create-pasajeros 1 [crear_pasajero_der] ]
```

```
;; generación espontánea de pasajeros (desde molinetes)
```

```
;; se crean cada 1000 ticks = 1 minuto
```

```
if (ingresantes = true and ticks mod 1000 = 999) [create-pasajeros 1 [crear_pasajero_ingreso] ]
```


DATOS REALES: TRENES



- Horario de operación: entre las 5 AM y las 10 PM
- Son 160 servicios por día (80 x andén)
- Frecuencia:
 - Cada 12 min entre las 5:00 y 12:00
 - Cada 15 min entre las 12:00 y 17:00
 - Cada 12 min entre las 17:00 y 21:00
 - Cada 15 min entre las 21:00 y 22:00

```
;; actualizar frecuencia (solo 1 vez por hora)
if (ticks mod 60000 = 1) [
  set hora_actual floor(ticks / 60000) + hora_inicio

  (ifelse
    hora_actual = 5 [set frecuencia_tren 12000]
    hora_actual = 12 [set frecuencia_tren 15000]
    hora_actual = 17 [set frecuencia_tren 12000]
    hora_actual = 21 [set frecuencia_tren 15000]
    hora_actual >= 22 [ask turtles [die] stop]
  )
]
```

10000 ticks = 1 minuto

DATOS REALES: PASAJEROS

- Máxima cantidad:
 - 311.000 en el mes de Octubre
 - 10.000 por día
- 10.000 personas / 160 servicios, resultan 64 personas por servicio.
- Son 32 que suben y 32 que bajan.

```
to cant_bajan_segun_mes
  let total 0

  (ifelse
    mes = "Enero"      [set total 212613]
    mes = "Febrero"    [set total 231473]
    mes = "Marzo"      [set total 263145]
    mes = "Abril"      [set total 275146]
    mes = "Mayo"       [set total 285549]
    mes = "Junio"      [set total 218172]
    mes = "Julio"      [set total 197322]
    mes = "Agosto"     [set total 288444]
    mes = "Septiembre" [set total 308205]
    mes = "Octubre"    [set total 311335]
    mes = "Noviembre"  [set total 308932]
    mes = "Diciembre"  [set total 286682]
  )

  ;; divido por 30 días, 160 servicios
  ;; la mitad bajan, la mitad suben
  set cant_bajan round(total / (30 * 160 * 2))

end
```

CONFIGURACIONES APLICABLES

- Hora inicio: seleccionar hora del día que comienza la simulación.
- Mes: seleccionar mes del año para la simulación.
- Tolerancia andén: tiempo máximo que espera el tren cuando está detenido en el andén.
- Frecuencia de tren: tiempo que transcurre entre que pasa un tren y el siguiente.
- Cantidad bajan: cantidad de pasajeros que bajan del tren.



hora_inicio 5

mes
Enero

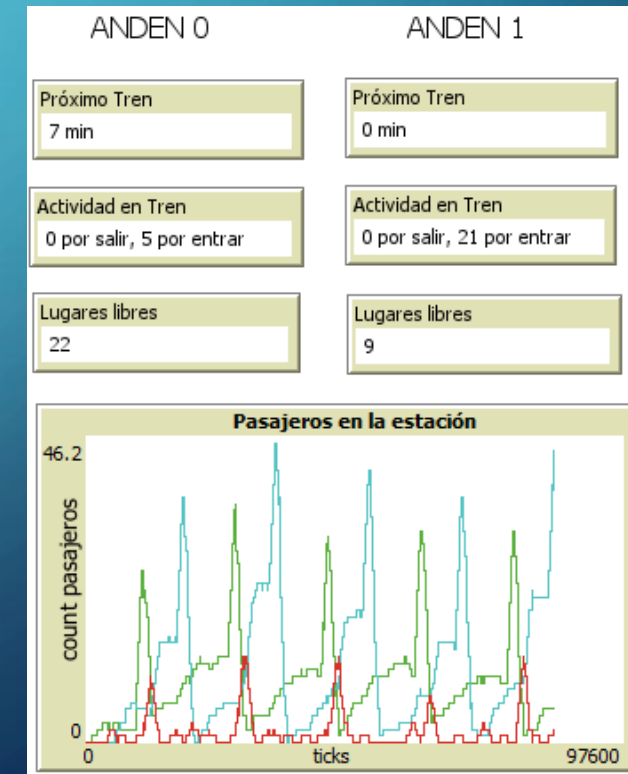
tolerancia_anden 1000

frecuencia_tren 12000

cant_bajan 22

SALIDAS GENERADAS

- Próximo Tren: tiempo de llegada del siguiente tren al andén.
- Actividad en Tren: indica cantidad de pasajeros que saldrán y que se deben generar, y cuántos pasajeros deben subir al tren.
- Lugares libres: celdas que ningún pasajero tiene “reservada”.
- Pasajeros:
 - Rojo: aquellos que están en el puente.
 - Verde: aquellos que se encuentran en el andén izquierdo.
 - Azul: aquellos que están en el andén derecho.



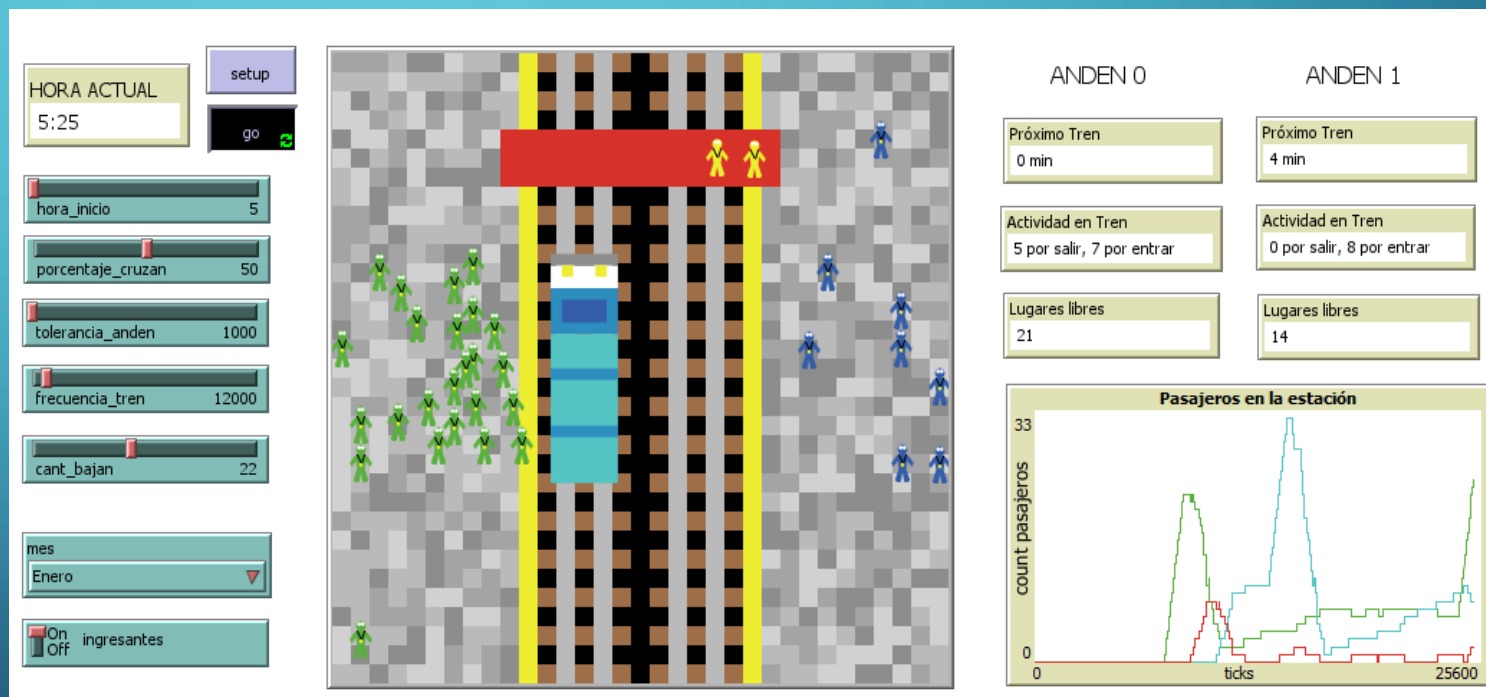
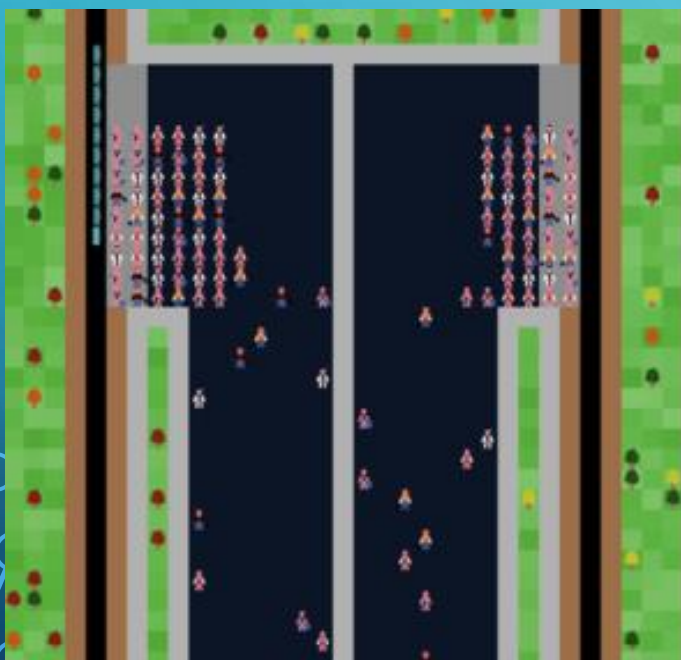
VERSIÓN DE LA HERRAMIENTA



- Finalmente se optó por NetLogo 2D.
- El puente en 2D es suficiente para poder desarrollar la simulación.
- Posible mejora estética en NetLogo 3D.



VS. PROYECTO ANTERIOR



REFERENCIAS



- Horarios actualizados: <https://www.trenroca.com.ar/img/pdf/Horarios-Trenes-Constitucion-La-Plata.pdf>
- Boletos vendidos por estación: <https://www.argentina.gob.ar/transporte/cnrt/estadisticas-ferroviarias>
- Documentación NetLogo: <https://ccl.northwestern.edu/netlogo/docs/>



GRACIAS POR SU ATENCIÓN