

Sistemas Multiagente (SMA)

# **Simulación basada en Agentes con AnyLogic**

**Alumnos:** Marcos Esteve Casademunt  
David Gimeno Gómez

**Curso:** MIARFID 2019/2020

## Índice

1. Introducción .....	2
2. Simulación Computacional .....	2
2.1. Introducción a la Simulación basada en Agentes .....	3
3. AnyLogic: Herramienta de desarrollo de Simulación Computacional .....	5
3.1. Introducción a Anylogic .....	5
3.2. Desarrollo de un ejemplo práctico .....	6
3.2.1. Simulación de un Modelo de Mercado .....	6
3.2.2. Población .....	7
3.2.3. Comportamiento del Agente .....	8
3.2.4. Extracción de Analíticas .....	12
3.2.5. Interpretación de los resultados .....	14
3.3. Desarrollo de un Modelo de Mercado más complejo .....	14
3.3.1. Modificaciones sobre el ejemplo práctico .....	14
3.3.2. Interpretación de los resultados .....	15
4. Usos de AnyLogic .....	17
5. Conclusiones .....	19
6. Bibliografía .....	19

## 1. Introducción

A menudo, debido a diversos motivos, no es conveniente o no es posible realizar experimentos en el mundo real. Existe una gran cantidad de escenarios en los que sería necesario realizar una experimentación previa como por ejemplo: cuando se desea mejorar las salidas de emergencia en un estadio para reducir el riesgo al mínimo, estudiar si debemos invertir en un activo concreto, analizar la propagación de enfermedades en una sociedad, planificar las líneas de montaje en una fábrica, entre otros. En otras palabras, surge la necesidad de emplear herramientas que nos permitan emular estas situaciones sin ningún tipo de riesgo y, a partir de las cuales, poder observar los datos generados para extraer una serie de conclusiones. Más concretamente, estaríamos hablando de la Simulación Computacional.

A lo largo de esta memoria, se va a presentar el ámbito de la Simulación Computacional en un marco general. Tras ello, nos centraremos en la Simulación Basada en Agentes, ya que ha sido el tipo de simulación estudiada durante la asignatura Sistemas Multiagente cursada en el máster. Posteriormente, presentaremos la herramienta AnyLogic, la cual es ampliamente empleada en el entorno industrial. Nuestro enfoque consiste en ir describiendo la herramienta y sus funcionalidades a través del desarrollo de un ejemplo de modelado sencillo. Además, se mostrará cómo poder analizar los datos generados por nuestra simulación a través de gráficas y análisis. Cabe destacar que este desarrollo se ha dividido en distintas fases con la finalidad de facilitar la comprensión del modelaje. Una vez, conozcamos las bases de AnyLogic, procederemos a realizar modificaciones sobre el ejemplo con la finalidad de introducir otros aspectos a simular o incluir una mayor complejidad al modelo. Para finalizar, comentaremos una batería de conclusiones respecto al proyecto realizado.

## 2. Simulación Computacional

La Simulación Computacional trata de englobar todas aquellas técnicas que permiten la creación de modelos capaces de simular un escenario en concreto del mundo real a través de computadores. Además, proporciona un método de análisis que se puede verificar, comunicar y comprender de manera eficaz. Por lo tanto, tal y como se ha comentado anteriormente, debe incluir herramientas necesarias para el tratamiento de los datos surgidos durante la simulación para poder analizar y extraer conclusiones que ayuden en la toma de decisiones del mundo real. Es por ello, que el modelado de simulación es ampliamente utilizado en un entorno industrial, así como académico con la finalidad de encontrar soluciones a problemas complejos, tal y como sugiere la Figura 1. Como se ha comentado anteriormente, su utilidad reside en la inviabilidad de llevar a cabo experimentos que, ya sea por el coste temporal o económico como por motivos éticos, no puedan realizarse en el mundo real.

La capacidad de analizar el modelo a medida que se ejecuta, diferencia el modelado de simulación de otros métodos, como los que utilizan Excel o la programación lineal. Al poder inspeccionar los procesos e interactuar con un modelo de simulación en acción, tanto la comprensión como la confianza se construyen rápidamente. Por ello, aunque este tipo de *software* se fundamenta principalmente en modelos matemáticos encargados de gobernar y

provocar la evolución de las distintas situaciones, también requieren del soporte gráfico para que el usuario pueda observar, de una manera más natural, el comportamiento que emerge en el escenario que esté simulando con la herramienta.



**Figura 1.** Síntesis del proceso de Simulación Computacional

Por otro lado, la Simulación Computacional presenta diferentes vertientes de simulación entre las que destacamos las siguientes:

- Simulación por eventos discretos: se trata de aquella simulación que modela un sistema mediante cambios en momentos discretos del tiempo que son accionados por eventos, de forma que evolucione el estado del modelo.
- Simulación basada en agentes: este es el caso estudiado principalmente en nuestro trabajo, se trata de modelar el comportamiento individual de cada entidad en el problema y observar cuales son los comportamientos que emergen de la interacción entre ellos.
- Simulación por sistemas dinámicos: simulación basada en modelos matemáticos para modelar el comportamiento y la dinámica de cada entidad que conforma el escenario que queremos modelar.

En nuestro caso, nos centraremos en la Simulación Basada en Agentes, tal y como introducimos al inicio de la memoria. Por ello, antes de avanzar con la herramienta utilizada y nuestro entorno de simulación, es conveniente presentar este paradigma en el siguiente apartado.

## **2.1. Introducción a la simulación basada en Agentes**

Como hemos ido comentando a lo largo de la memoria, la Simulación Computacional nos permite modelar distintos escenarios con el objetivo de analizar toda la posible casuística que pueda acontecer sobre nuestro entorno simulado. De este modo, conseguimos evitar los grandes costes económicos que implicaría una simulación real, poner en riesgo a personas, entre otras circunstancias. Por otro lado, presenta la ventaja de permitir una

anticipación o planificación de sucesos futuros. En definitiva, la Simulación Computacional proporciona una herramienta fundamental en una gran multitud de ámbitos.

No obstante, a menudo puede resultar inviable modelar todos los comportamientos que pueden tener lugar en el dominio que estemos simulando. Incluso, a menudo podrán aparecer situaciones o comportamientos que no habíamos tenido en cuenta o que simplemente no habríamos imaginado que pudiera suceder. Además, normalmente nuestro entorno estaría conformado por una serie de entes o seres, cada uno con funciones o comportamientos distintos; y en los que tomaría especial relevancia, además, un factor aleatorio. En consecuencia, esta característica provocaría comportamientos influenciados por las acciones de otros agentes, los cuales serían bastante complejos de modelar. Todo esto, nos proyecta hacia el empleo de la denominada Simulación Basada en Agentes. A diferencia de otros tipos de simulación en los que aproximamos las interacciones bajo extensos modelos matemáticos, en la Simulación Basada en Agentes el pilar fundamental de la emulación reposa sobre el comportamiento independiente de cada uno de los agentes que definamos sobre el entorno. De esta manera, tal y como se ha estudiado en la asignatura, lograríamos modelar y analizar *comportamientos emergentes*. Este concepto, también conocido como *comportamiento colectivo*, se refiere a aquellos comportamientos que surgen a partir de la interacción entre los agentes del entorno, cada uno de ellos comportándose bajo su propia política pero que resultan en un conductas que podrían ser impredecibles. En otras palabras, las decisiones tomadas por otros agentes pueden condicionar el comportamiento de otro agente. Por esta razón, este tipo de modelado es ampliamente utilizado cuando se trata de simular distintos fenómenos sociales y económicos, como sería el caso de ejemplo que desarrollaremos más adelante o el ejemplo mostrado en la Figura 2, donde se ha simulado el tráfico en una determinada zona.



**Figura 2.** Simulación Basada en Agentes del tráfico en una ciudad mediante la herramienta AnyLogic

Además de proporcionar un entorno libre de riesgo, ahorrarnos costes y tiempo, la Simulación Basada en Agentes nos permite visualizar nuestro entorno modelo de una manera gráfica e intuitiva, donde podemos observar aquellos comportamientos emergentes que puedan ser de gran interés. Esta virtualización se realiza a través de un espacio reticular parecido a los Autómatas Celulares y donde cada celda representaría una posición geográfica, básicamente como podemos observar en la Figura 2. Por otra parte, estas herramientas nos permitirán interactuar, también de forma gráfica e incluso durante la

simulación, con distintos atributos de nuestro modelo como podría ser cambiar la frecuencia con la que cambian los semáforos de la intersección. Y, por último indicar, tal y como se ha ido avanzando, que también seremos capaces de extraer análíticas con las que poder extraer una serie de conclusiones o aspectos relevantes de cara a una planificación.

### 3. Anylogic: Herramienta de desarrollo de Simulación Computacional

#### 3.1. Introducción a Anylogic

AnyLogic es una plataforma para la construcción de modelos de simulación. A grandes rasgos, se trata de un entorno gráfico de simulación basado en un entorno Java. Cabe destacar que, si se desea, se puede modificar y ampliar los modelos de simulación a través de bibliotecas y código externo o, bien, utilizar el entorno predefinido. Esta característica proporciona, además, la capacidad de crear aplicaciones Java completas para su uso incluso en terminales sin el *software* instalado. Por otro lado, AnyLogic presenta la posibilidad de modelar nuestro sistema mediante distintos paradigmas. En nuestro caso, como ya se ha ido avanzando, nos centraremos en la Simulación basada en Agentes.

Más concretamente, AnyLogic aporta una interfaz que integra todas las herramientas que permiten el modelado de nuestro escenario, tal y como muestra la Figura 3. De este modo, en resumidas cuentas, la interfaz consigue reducir la cantidad de código a introducir por parte del usuario.

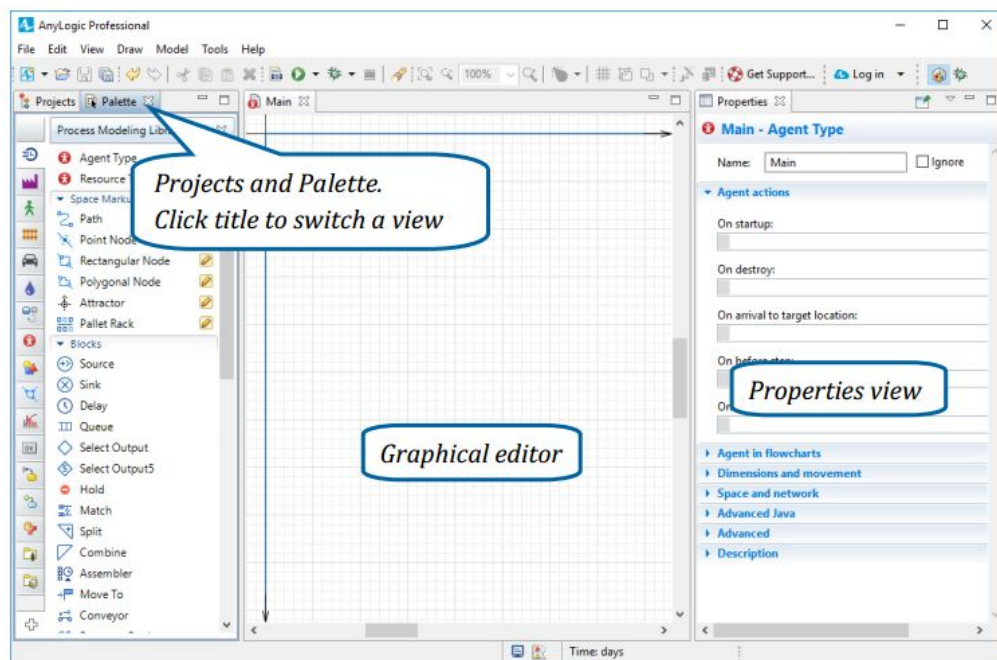


Figura 3. Interfaz de la herramienta AnyLogic

Antes de seguir avanzando con el desarrollo del ejemplo práctico será necesario familiarizarnos primero con la interfaz de usuario de AnyLogic. Tal y como puede observarse en la Figura Apolo distinguimos entre las distintas áreas de trabajo:

- La pestaña *Projects* nos permite gestionar y navegar entre los distintos modelos o proyectos que estemos desarrollando.

- El área del *Graphical Editor* proporciona un entorno de interacción donde podemos construir y editar el diagrama de estados de nuestro agente, así como arrastrar (desde la pestaña *Palette*) los distintos elementos que vayamos definiendo en nuestro entorno. Además, aquí se define un marco con bordes azules que delimita la ventana que se mostrará durante la simulación. Es en este marco donde pondremos la visualización gráfica de nuestro escenario, las analíticas y demás funcionalidades que permitan interactuar con el usuario como botones, *sliders*, entre otros.
- La pestaña *Palette* agrupa en una serie de categorías los distintos objetos y herramientas que podemos añadir a nuestro entorno con el simple hecho de arrastrarlos sobre el *Graphical Editor*.
- La zona de *Properties* muestra todas las propiedades y parámetros que podamos añadir o modificar sobre un objeto en concreto. Esta pestaña es dinámica, es decir, su contenido dependerá del objeto que estemos seleccionando en ese momento.

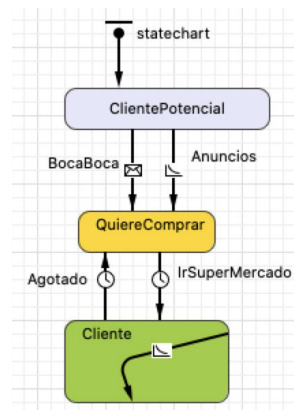
Una vez conocemos nuestro *workspace* ya podríamos ponernos manos a la obra. No obstante, antes de nada, a modo de motivación, conoceremos varios ejemplos realizados con AnyLogic para saber qué seríamos capaces de construir con esta herramienta.

### 3.2. Desarrollo de un ejemplo práctico

Tal y como se comentó en la introducción de la memoria, nuestro objetivo es presentar la herramienta AnyLogic para el modelado de un Sistema de Simulación basado en Agentes a través del desarrollo de un ejemplo práctico. De esta manera, conseguimos explicar y mostrar, de un manera interactiva y ligera, las distintas funcionalidades básicas de la herramienta describiendo el paso a paso del proceso.

#### 3.2.1. Simulación de un Modelo de Mercado

Nuestro ejemplo pretende modelar un entorno o escenario de mercado para analizar cómo la publicidad y el boca a boca entre las personas influyen sobre las ventas y la opinión de nuestro producto. En otras palabras, permite simular que ocurriría al introducir un nuevo producto en el mercado y qué acciones, respecto a publicidad, podemos realizar, así como analizar o predecir posibles situaciones futuras.



**Figura 4.** Diagrama propuesto en el ejemplo práctico

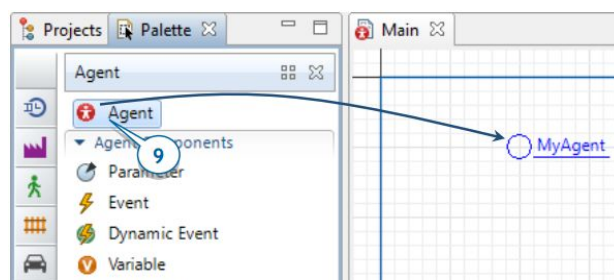
Más concretamente, en la Figura 4 podemos observar el diagrama de ejecución que refleja el comportamiento de nuestro agente. En un inicio, el agente empieza en un estado en el



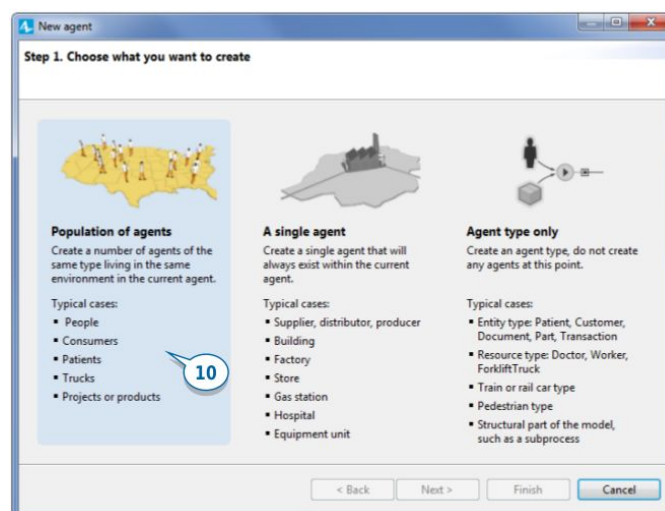
cual es un cliente potencial, mediante la influencia de la publicidad o del boca a boca pasa a un estado donde quiere comprar el producto y, tras un tiempo que supone el desplazamiento al supermercado, el cliente potencial pasa a ser cliente. Además, este modelo básico asume que cuando un usuario se convierte en cliente le gusta el producto y, por tanto, trata de convencer a otros agentes de que compren el producto mediante el boca a boca. Por último, cuando al agente se le acaba el producto vuelve a pasar a un estado donde quiere comprar dicho producto. En los sucesivos apartados, comentaremos por encima los detalles seguidos para construir dicho modelo básico paso a paso.

### 3.2.2. Población

En primer lugar, es necesario definir una población de agentes donde cada agente asuma un rol. Anylogic nos permite definir una población de una forma intuitiva y visual. Simplemente debemos especificar que queremos crear un nuevo agente y, además, que este debe estar dentro de una población con un tamaño de población específico. Para ello, primero debemos arrastrar un objeto *Agent* sobre el diagrama principal (*Main*) de nuestro modelo de mercado. Una vez situado nuestro agente, se desplegará un diálogo para especificar los detalles del agente y la población que conformará, tal y como se pueden observar en las Figuras 5 y 6 mostradas a continuación.



**Figura 5.** Creación de un objeto Agente en AnyLogic

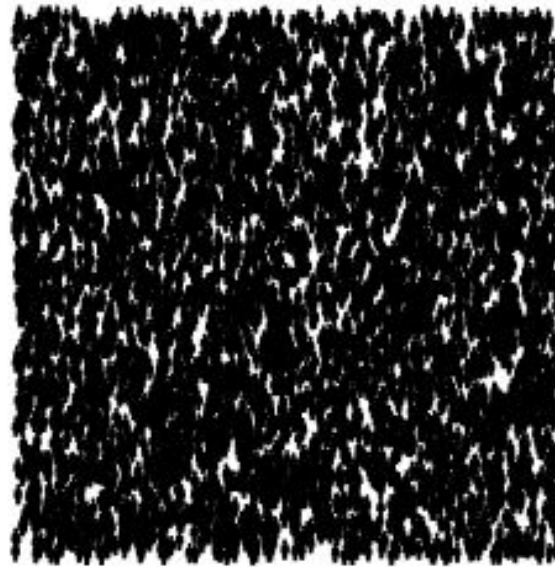


**Figura 6.** Concretar el tipo de objeto Agente que deseamos definir

Una vez creada la población, si realizamos una simulación podemos observar un comportamiento estático por parte de los agentes, tal y como se observa en la Figura 7. No



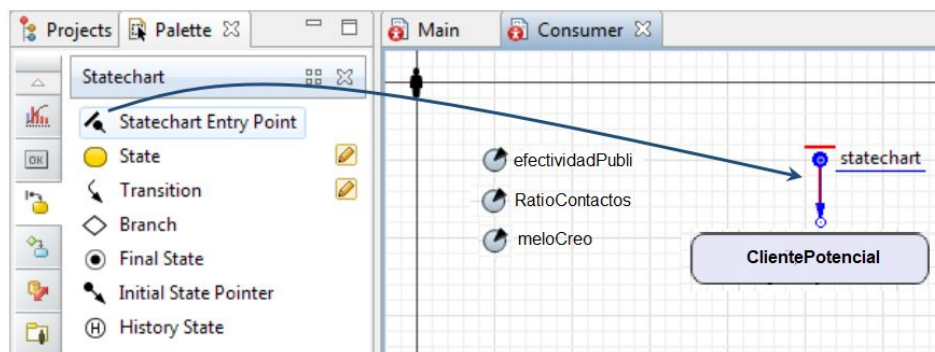
obstante, estos agentes no interactúan entre ellos ni con el entorno. Por ello, en los sucesivos apartados explicaremos los detalles para definir el comportamiento del agente.



**Figura 7.** Simulación inicial del ejemplo práctico sin ningún tipo de interacción

### 3.2.3. Comportamiento del Agente

Como se ha indicado anteriormente, modelaremos las características y el comportamiento de nuestro agente mediante la definición de un diagrama de estados. En primer lugar, comenzamos arrastrando sobre *Graphical Editor* un punto de inicio, a partir del cual seremos capaces de ir añadiendo estados y transiciones, tal y como sugiere la Figura 8.



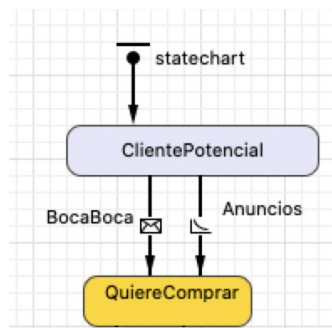
**Figura 8.** Creación del estado inicial del Diagrama de Estados de nuestro Agente

De la misma forma, introducimos una serie de parámetros con los que modelar distintos aspectos de nuestro agente, los cuales tomarán importancia en sucesivos pasos. Más concretamente, cada uno de ellos simboliza:

- *efectividadPubli*: modela la influencia de la publicidad sobre la opinión de nuestro agente. A través del panel *Properties*, podemos especificarle un valor inicial de 0.01.
- *RatioContactos*: nos permite definir el número mensajes que enviaría nuestro agente a los *potenciales clientes* para convencerles de que compren el producto.

- *meLoCreo*: determina, mediante una probabilidad, la confianza de los *potenciales clientes* en el boca a boca de otros agentes. En otras palabras, modela cuán fácil de convencer son nuestros agentes.

Teniendo en cuenta todos estos aspectos, introduciremos los primeros estados en los que puede encontrarse nuestro agente. En primer lugar, arrastramos sobre el editor los objetos necesarios para modelar dos de los estados y las transiciones entre ellos, tal y como sugiere la Figura 9 mostrada a continuación.



**Figura 9.** Transiciones del estado Cliente Potencial al estado QuiereComprar

Antes de nada, con el objetivo de distinguir en qué estado se encuentra un agente durante la posterior visualización gráfica del entorno de simulación, debemos comentar cómo asociar un color a cada estado. Para ello, debemos acceder al panel de *properties* del estado y realizar las indicaciones mostradas en la Figura 10.

Name:  ☒ Show name

☐ Ignore

Fill color:

Entry action:

**Figura 10.** Establecer propiedades del estado Cliente Potencial respecto al color asociado

Por otro lado, como ya comentamos, nuestro agente comienza como un posible cliente del producto y las únicas formas para que decida comprarlo es mediante la influencia de los anuncios o mediante la interacción con otros agentes. Por lo tanto, hemos definido una transición para cada una de estas opciones.

Comenzamos, detallando el denominado *BocaBoca*. Accediendo al panel de propiedades de la transición, bastará con indicar que la transición se activa cuando recibamos un mensaje y su contenido sea "Compra". Entonces, tal y como sugiere la Figura 11, de forma aleatoria en función del atributo *meLoCreo* determinaremos si nuestro cliente potencial ha sido convencido o no.

Name:  ☒ Show name ☐ Ignore

Triggered by:

Message type:

Fire transition: ☐ Unconditionally  
☒ On particular message  
☐ If expression is true

Message:

Action:

Guard:

**Figura 11.** Definición de la transición que modela la interacción oral entre agentes

Respecto a la publicidad, debemos crear una transición de tipo *rate*, es decir, lograremos transitar al estado *QuiereComprar* una vez pase un tiempo medio gobernado por una distribución exponencial, la cual será parametrizada mediante un ratio, en nuestro caso, nuestro atributo *efectividadPubli*. Además, debemos especificar la unidad de tiempo en la que se basará el modelado de la publicidad, tal y como sugiere la Figura 12. Este tipo de transición sería similar a una transición *TimeOut* que utilizaremos más adelante. Cabe destacar que AnyLogic proporciona un gran abanico de transiciones que nos permiten modelar multitud situaciones y que podemos consultar en cualquier momento en su documentación.

Name:  ☐ Show name

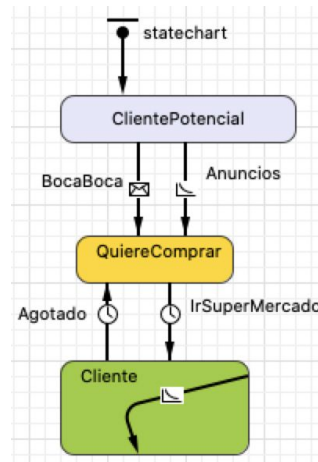
☐ Ignore

Triggered by:

Rate:

**Figura 12.** Definición de la transición que modela la influencia de la publicidad sobre los agentes

No obstante, faltan detalles por aclarar: quién produce los mensajes que pueden influir sobre nuestros agentes, cuando una agente compra el producto y qué ocurre cuando consumimos la vida útil del producto. Todo esto queda reflejado sobre la Figura 13 que se muestra a continuación, donde ya se han añadido los elementos necesarios para resolver las cuestiones anteriores. Definiendo de esta manera, el diagrama de estados que presentamos al inicio de esta sección.



**Figura 13.** Definición de las transiciones entre el estado QuiereComprar y Cliente

Para modelar las dos nuevas transiciones hemos empleado una transición de tipo *Timeout*. En otras palabras, una vez el agente pertenece al estado fuente de la transición, cambiará de estado cuando transcurra un tiempo especificado. Bastará con que mostremos como ejemplo la definición de una de las transiciones y, para ello, se introduce la Figura 14.

Name:  ☒ Show name ☐ Ignore

Triggered by:

Timeout:

**Figura 14.** Definición de la transición que modela el tiempo de vida del producto

En cuanto al nuevo estado *Cliente*, lo único a destacar sería la capacidad de enviar mensajes al resto de agentes. Para ello, en primer lugar definiremos una transición interna en dicho estado mediante el objeto *transition* que hemos ido empleando a lo largo de la memoria. Una vez más, accederemos al panel de propiedades de la transición en cuestión para modelar el comportamiento deseado: enviar mensajes a otros agentes. Existen varios métodos para este propósito desde mandar un mensaje a un agente en concreto (*send(smg, agt)*), a un agente aleatoriamente (*sendToRandom(msg)*) o enviar un mensaje a todos los agentes (*sendToAll(msg)*). En nuestro caso, emplearemos el último método y, como vimos anteriormente, sólo tendremos en cuenta estos mensajes si somos un cliente potencial. Además, el número de mensajes estará modelado por un ratio guiado por el atributo *RatioContacto*, tal y como expone la Figura 15.

Name:  ☒ Show name ☐ Ignore

Triggered by:

Rate:

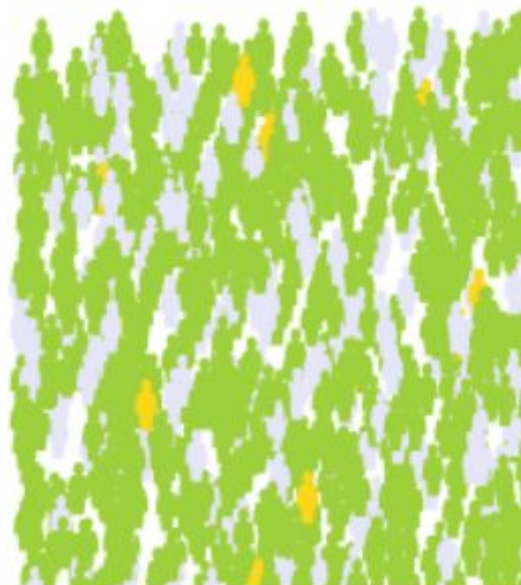
Action:

**Figura 15.** Definición de la transición que permite enviar mensajes a otros agentes aleatoriamente

Con todo esto, ya tenemos nuestro modelo de mercado básico, a partir del cual podremos construir gráficas con las que extraer una serie de conclusiones y análisis, tal y como explicamos en el siguiente apartado.

#### 3.2.4. Extracción de analíticas

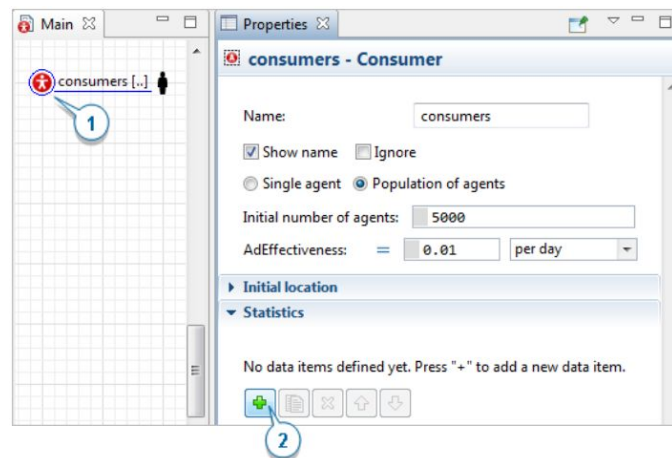
Tras la definición de las interacciones con el entorno y los usuarios, así como el comportamiento del agente, nuestra simulación se presenta de una forma más visual, exponiendo los distintos estados por los que transcurren los agentes de un modo más claro, tal y como se puede observar en la Figura 16.



**Figura 16.** Simulación generada una vez definido el comportamiento del agente y sus interacciones

No obstante, no podemos extraer ninguna conclusión por nosotros mismos a partir de una visualización. Es por ello que uno de los aspectos más importantes a la hora de realizar simulaciones es el uso de gráficas y analíticas que permitan al usuario entender que está ocurriendo en su simulación. Anylogic dispone de una gran cantidad de herramientas para facilitar esta tarea permitiendo al desarrollador prototipar soluciones rápidas de una forma intuitiva y visual.

En un primer lugar, debemos ser capaces de definir estas gráficas mediante la herramienta AnyLogic. Para ello, desde el diagram *Main* accederemos a las propiedades de nuestro agente consumidor. En esta ventana, desplegamos la pestaña *Statistics*. En nuestro caso, construiremos una gráfica que muestre la cantidad de agentes que se encuentran en cada estado a lo largo del tiempo. Por ello, añadiremos tantos *items* como estados tengamos, aunque en la memoria sólo expongamos un *item* a modo de ejemplo. Básicamente, estos *items* nos permiten especificar si deseamos realizar un conteo, sumar o calcular la media entre otras opciones. El proceso descrito queda reflejado en las Figuras 17 y 18.

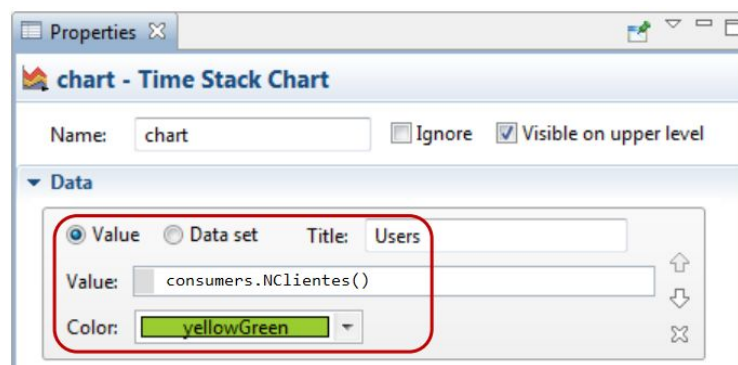


**Figura 17.** Serie de pasos a seguir para poder introducir estadísticas respecto a nuestro agente



**Figura 18.** Definición de una estadística que cuenta el número de clientes potenciales

Ahora, solo nos queda arrastrar desde la pestaña *Palette* un objeto de la clase *Analysis*, más concretamente, una gráfica del tipo *Time Stack Chart*, sobre nuestro diagrama principal. Al igual que con las transiciones, es necesario indicar que existen un gran abanico de gráficas y analíticas, cuya definición puede consultarse en la documentación de la herramienta AnyLogic. Respecto a nuestra gráfica, simplemente tendremos que indicar las fuentes de datos que acabamos de definir, es decir, nuestros *items*. Como siempre, accederemos a la ventana de propiedades y procederemos tal y como muestra la Figura 19 que mostramos a continuación. Con el propósito de mantener una coherencia, asignaremos los colores respectivos a cada una de las estadísticas.



**Figura 19.** Definición de una fuente de datos concreta para una analítica

Por otro lado, de una forma análoga a como se haría en NetLogo (herramienta de simulación estudiada en clase), se ha añadido un *slider* para permitir al usuario modificar el tiempo de vida del producto durante la simulación en tiempo real.



### 3.2.5. Interpretación de los resultados

Destacamos, entre las múltiples pruebas realizadas, la analítica mostrada en la Figura X. Este comportamiento emergente ha surgido definiendo un tiempo de vida corto del producto, tal y como sugieren los valles reflejados en la imagen mencionada, y los valores definidos a lo largo de la memoria para el resto de atributos. Estos valles indican un descenso en el número de clientes, ya que éstos al consumir el producto, vuelven a un estado en el que desean comprar otra vez y, de ahí, se explica el ascenso progresivo de clientes hasta un nivel similar al anterior pico. En definitiva vemos, cómo al cabo de un tiempo dejan de existir potenciales clientes, es decir, todos los agentes acaban sucumbiendo ante la publicidad y la presión social.

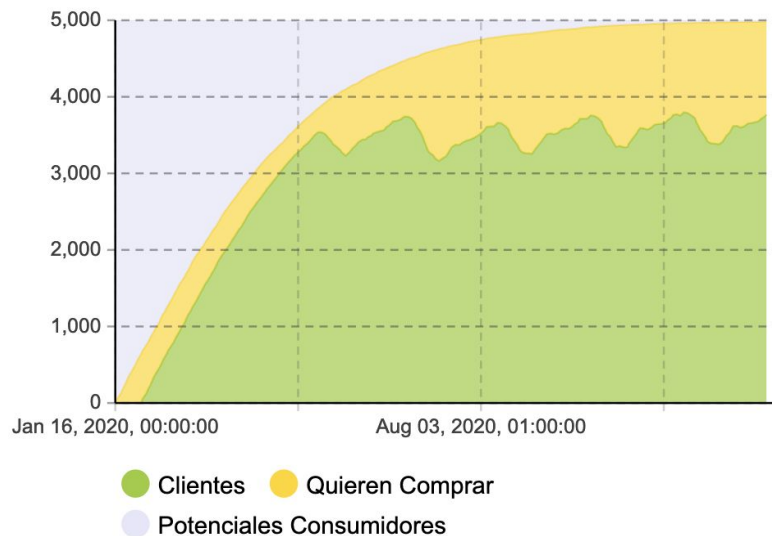


Figura 20. Análisis del comportamiento emergente de nuestra simulación

### 3.3. Desarrollo de un Modelo de Mercado más complejo

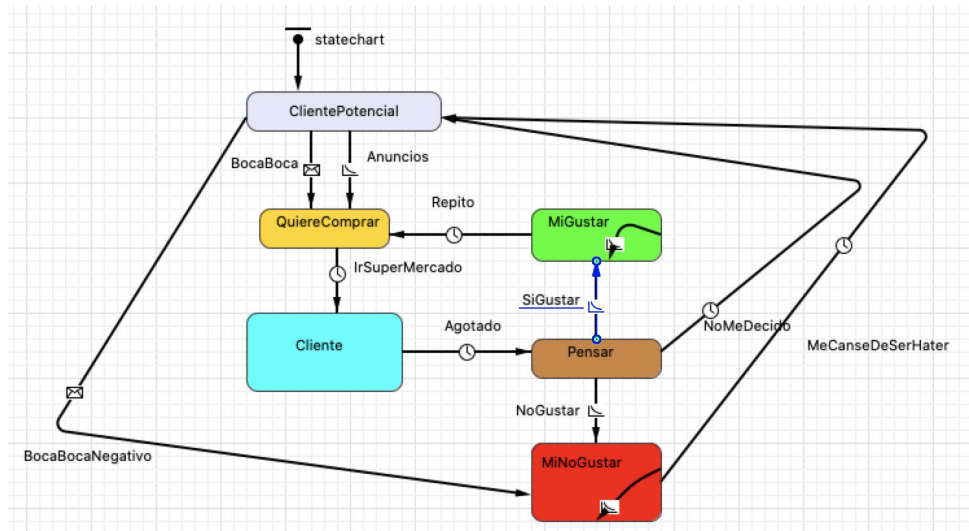
#### 3.3.1. Modificaciones sobre el ejemplo práctico

Una vez comentado los aspectos básicos de la construcción de la simulación, pasamos a comentar algunas de las modificaciones realizadas sobre el entorno básico para conseguir llegar al entorno descrito en la Figura 21.

En esta nueva simulación tenemos un comportamiento similar al original. En primer lugar, todos los agentes empiezan siendo **clientes potenciales**, mediante la publicidad o el boca a boca positivo pueden pasar a clientes que **quieren comprar** el producto y, tras un tiempo de ir al supermercado, convertirse en **clientes**. Una vez se le ha acabado el producto al agente, pasa al estado **pensar** donde decide de forma estocástica si le gusta el producto o no (de esta modo ya no se asume que al cliente siempre le va a gustar el producto). En caso de gustarle el producto, pasará en querer comprar de nuevo dicho producto y, además, enviará a otros agentes un mensaje para convencerles de que lo compren. En caso contrario, pasará o bien a ser de nuevo cliente potencial, si no se decide; o bien a un estado donde no le gusta el producto y envía mensajes de disgusto a otros agentes para que no lo compren, expresando, de esta forma, su disconformidad. Por otro lado, existe



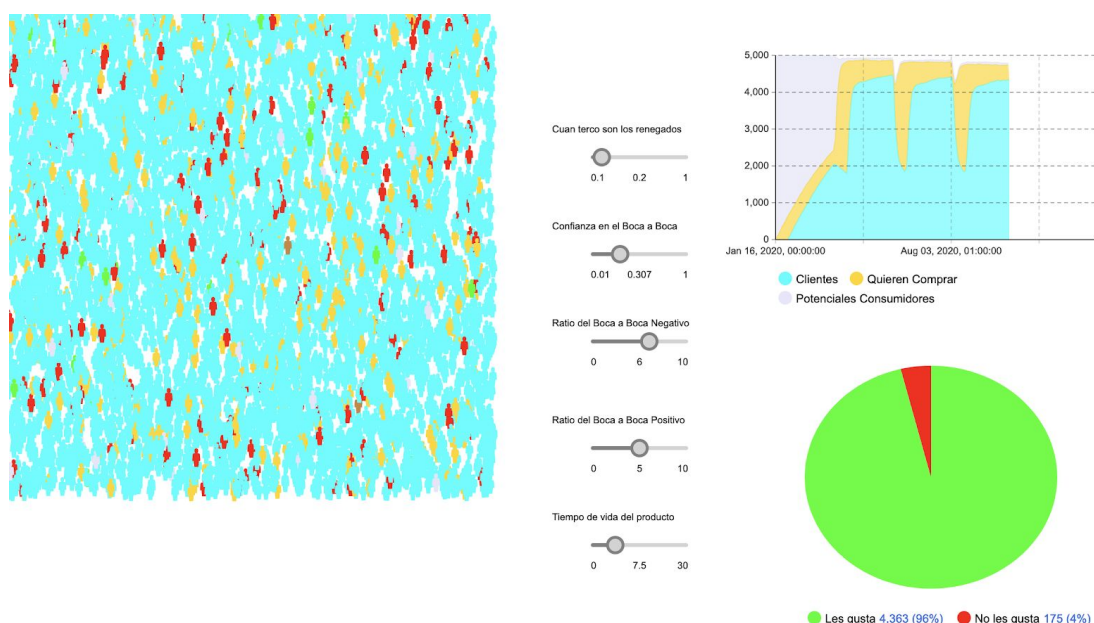
también una transición de cliente potencial al estado **no me gusta** en caso de recibir muchos mensajes de disgusto con el producto. En otras palabras, ya no se asume que las interacciones entre los agentes siempre sean beneficiosas para la empresa que fabrica el producto. Por último, con el fin de evitar que el estado **no me gusta** sea un estado sumidero, se ha incluido una transición a **cliente potencial** tras un tiempo concreto, modelando de este modo un cambio de opinión en el agente.



**Figura 21.** Modificaciones realizadas sobre el Modelo de Mercado propuesto en el ejemplo práctico

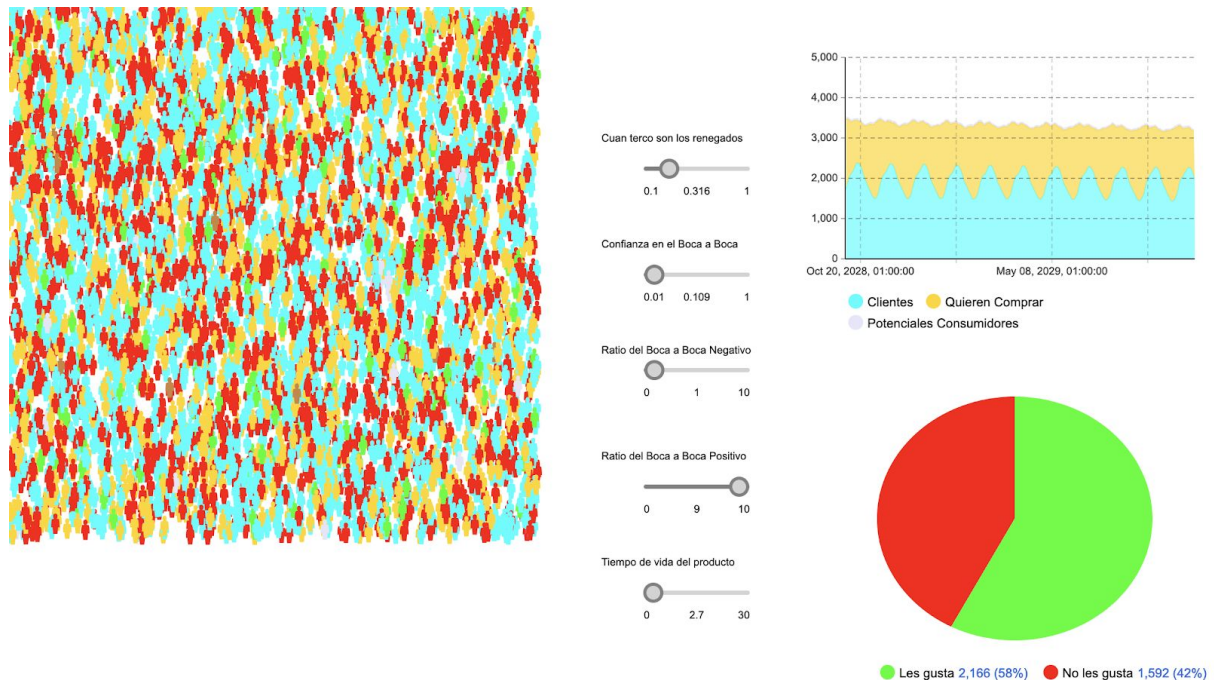
Para la construcción de este modelo se han utilizado las distintas funcionalidades descritas en el ejemplo práctico, es decir, se han tenido que añadir más atributos para modelar los detalles de cada transición, así como nuevos estados que proporcionasen al modelo de mercado un mayor realismo. Todo este modelo puede consultarse en mayor detalle en el archivo adjunto en la entrega de esta memoria.

### 3.3.2. Interpretación de los resultados

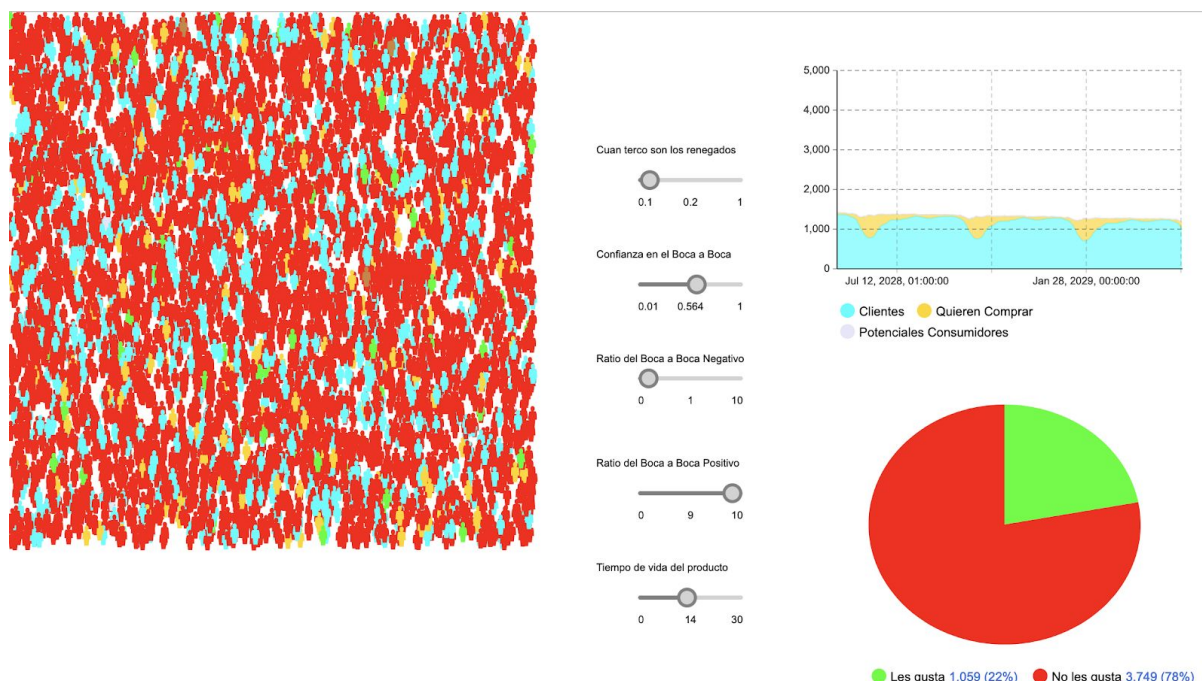


**Figura 22.** Ventana de simulación con interacción en tiempo real con el usuario

Una vez ejecutamos la simulación podemos observar en la Figura 22 como disponemos de un cuadro de mando formado por distintos *sliders* para modificar el comportamiento de ciertos parámetros así como gráficos que permiten al usuario ver cómo va evolucionando la población. Más concretamente los gráficos nos permiten observar cual es la evolución de los clientes potenciales, cuantos quieren comprar y cuantos acaban siendo clientes. Así mismo, nos permite observar cuántos agentes pasan por un estado en el que les gusta el producto.



**Figura 23.** Simulación del Modelo de Mercado cuando, principalmente, la vida del producto es reducida



**Figura 24.** Simulación del Modelo de Mercado cuando, principalmente, la influencia del boca a boca es alta

En la Figura 23, podemos observar cómo actúa el modelo del mercado al poner un tiempo de vida corto al producto. Por otro lado, un aspecto interesante a destacar es que ocurre cuando le damos una confianza al boca a boca alta, donde, tal y como se puede observar en la Figura 24, obtenemos que la mayoría de los agentes pasan a un estado donde no les gusta el producto y además el número de ventas de dicho producto disminuyen lo que demuestra un comportamiento con sentido si pensamos en el mundo real.

#### 4. Usos de AnyLogic

Existen una gran cantidad de librerías que implementan métodos para modelar el comportamiento de distintos tipos de entes como puede ser el flujo de peatones en un terminal de vuelos o el modelado de tráfico vial para el movimiento de automóviles, camiones y autobuses en carreteras, o incluso el modelado del flujo de trenes y buques de carga. En este apartado vamos a comentar algunos posibles casos de uso mediante el uso de la herramienta AnyLogic.

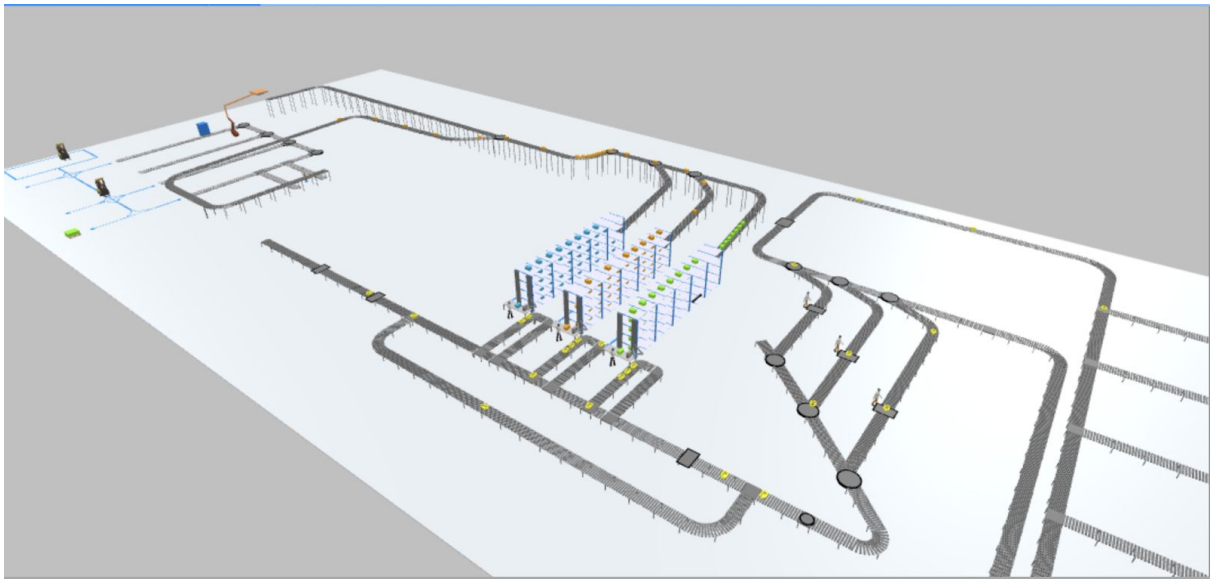
El primer caso se basa en modelar el tráfico de una ciudad o, más concretamente, en un cruce. Tal y como se puede observar en la Figura 25, este tipo de modelados nos permite obtener los posibles cuellos de botella en una ciudad (segmentados en rojo) y de esta forma tomar mejores decisiones de diseño de las calles y carreteras.



**Figura 25.** Simulación del tráfico de vehículos y su regulación dentro de una ciudad

Otro posible ejemplo podría ser la planificación de las líneas de distribución en un centro de logística, tal y como podemos observar en la Figura 26, se observan distintas líneas así como operarios y robots de transporte. Este tipo de simulaciones permiten por ejemplo a empresas como Amazon desarrollar centros de distribución con unas características óptimas para evitar cuellos de botella o errores en la entrega de los paquetes.





**Figura 26.** Simulación de un centro logístico y sus líneas de distribución

## 5. Conclusiones

En este trabajo hemos visto como la simulación puede ayudar a resolver una gran cantidad de problemas donde realizar una experimentación del tipo ensayo-error en el mundo físico puede acarrear grandes sobrecostos o incluso ser imposible como podría ser en el lanzamiento de cohetes espaciales.

En nuestro caso, aunque existen una gran cantidad de herramientas en el mercado para realizar simulaciones hemos decidido centrarnos en una herramienta muy empleada en entornos industriales como es el caso de AnyLogic. Esta herramienta permite modelar una gran cantidad de problemas de manera sencilla y eficiente donde con una curva de aprendizaje ligera se pueden conseguir los primeros modelos. Además, a diferencia de NetLogo podemos realizar modelados completos con muy pocas líneas de código; aprovechando para ello la interfaz que nos proporciona Anylogic. Por último, hemos visto cómo podemos realizar una simulación basada en agentes utilizando la herramienta y las conclusiones que se pueden extraer de la interacción entre las distintas entidades.

## 6. Bibliografía

- [1] Ilya Grigoriev, AnyLogic in Three Days: Modeling and Simulation Textbook.  
<https://www.anylogic.com/resources/books/free-simulation-book-and-modeling-tutorials/>
- [2] Herramienta AnyLogic: <https://www.anylogic.com/>
- [3] Simulación Basada en Agentes. Recursos del Máster IARFID, Universidad Politécnica de Valencia.