



UNIVERSITAT JAUME I
ESCOLA SUPERIOR DE TECNOLOGIA I CIÈNCIES EXPERIMENTALS
MÀSTER UNIVERSITARI EN ENGINYERIA INDUSTRIAL

***Desarrollo y aplicación de un modelo de simulación para
mejorar el almacenamiento intermedio de una planta de
producción cerámica***

TRABAJO FIN DE MÁSTER

AUTOR

Carlos Cantos González

DIRECTORES

Fernando Romero Subirón

Pedro Rosado Castellano

Castellón, Julio de 2021

“Cuando las cosas se pongan difíciles, pon un pie delante del otro y sigue adelante. No te rindas.” Roy T. Bennett

ÍNDICE GENERAL

**DESARROLLO Y APLICACIÓN DE UN MODELO DE SIMULACIÓN PARA
MEJORAR EL ALMACENAMIENTO INTERMEDIO DE UNA PLANTA DE
PRODUCCIÓN CERÁMICA**

UNIVERSITAT JAUME I

Máster en Ingeniería Industrial

Autor: Carlos Cantos González

Directores: Fernando Romero Subirón

Pedro Rosado Castellano

CASTELLÓN, JULIO 2021

ÍNDICE GENERAL

I. Memoria 9

II. Pliego de condiciones 107

III. Presupuesto 115

IV. Anexo 123

MEMORIA

**DESARROLLO Y APLICACIÓN DE UN MODELO DE SIMULACIÓN PARA
MEJORAR EL ALMACENAMIENTO INTERMEDIO DE UNA PLANTA DE
PRODUCCIÓN CERÁMICA**

UNIVERSITAT JAUME I

Máster en Ingeniería Industrial

Autor: Carlos Cantos González

Directores: Fernando Romero Subirón

Pedro Rosado Castellano

CASTELLÓN, JULIO 2021

ÍNDICE. MEMORIA

1. Objeto y campo de aplicación.....	15
2. Alcance	15
3. Antecedentes	15
3.1. El proceso cerámico	15
3.1.1. Introducción	15
3.1.2. Cerámica aplicada a pavimentos y revestimientos	16
3.1.3. Procesos y técnicas aplicados en la fabricación cerámica	17
3.1.3.1. Descripción general del proceso de fabricación	17
3.1.3.2. Moldeado/conformado de piezas	17
3.1.3.3. Secado de productos cerámicos	18
3.1.3.4. Esmaltación y decoración de productos cerámicos	19
3.1.3.5. Almacenamiento intermedio	21
3.1.3.6. Proceso de cocción de productos cerámicos	22
3.1.3.7. Clasificación y embalaje	24
3.2. Técnicas de modelado de procesos	26
3.2.1. Diagrama de flujo	27
3.2.2. Análisis de los flujos de materiales	28
3.3. Situación actual de Azulejos S.A.	29
3.4. Características de los recursos de la línea de producción	29
3.4.1. Diagrama de flujo de la línea de producción cerámica de Azulejos S.A.	32
3.5. Operaciones de producción en la línea de producción cerámica de Azulejos S.A.	34
3.5.1. Medidas de flujo básicas de las operaciones de producción	34
3.5.1.1. Medidas de flujo deterministas del almacén intermedio I	35
3.5.1.2. Medidas de flujo deterministas del horno	35
3.5.1.3. Medidas de flujo deterministas de la línea de Prensado-secado-esmaltado	36
3.5.1.4. Medidas de flujo deterministas de la línea de clasificación	37
3.5.1.5. Medidas de flujo deterministas de la línea de embalaje	37
3.5.1.6. Medidas de flujo teóricas del almacén intermedio II	38
3.5.1.7. Resumen de los resultados	38
4. Normas y referencias	39
4.1. Aplicaciones informáticas	39
4.1.1. Anylogic	39
4.1.1.1. Experimento de optimización en Anylogic	40
4.1.2. Hojas de cálculo MS EXCEL	41
4.1.3. Procesador de textos: MS WORD	41
4.1.4. Hardware	41
4.2. Plan de gestión de calidad	41
4.3. Referencias	41
5. Requisitos del modelo de simulación	42
5.1. Verificación del modelo de simulación	43
5.2. Experimento de optimización	43
5.2.1. Condiciones iniciales del experimento de optimización	43
6. Alternativas	44
6.1. Alternativa adoptada	44
6.2. Solución alternativa: desarrollo de un algoritmo de optimización para el funcionamiento de los LGV.	44
6.3. Solución alternativa: desarrollo en detalle del movimiento de las piezas cerámicas a lo largo de la ..	44
línea de Prensado-secado-esmaltado.	
7. Solución adoptada	45

7.1. Implementación de Anylogic	45
7.1.1. Entrada de la materia prima	46
7.1.2. Línea de Prensado-secado-esmaltado	49
7.1.2.1. Sección de prensado	50
7.1.2.2. Sección de secado	53
7.1.2.3. Sección de esmaltado	55
7.1.3. Almacenamiento intermedio de los azulejos (I)	57
7.1.3.1. Modelado de los Boxes en Anylogic	58
7.1.3.2. Área de Picking del final de etapa Prensado-secado-esmaltado (I)	59
7.1.3.3. Modelado de los LGV en Anylogic	61
7.1.3.4. Inicialización del Almacén intermedio	63
7.1.3.5. Área de Picking del final de etapa Prensado-secado-esmaltado (II)	65
7.1.4. Almacenamiento intermedio de los azulejos (II)	68
7.1.4.1. Funcionamiento del almacén intermedio de Bizcocho	70
7.1.5. Almacenamiento intermedio de los azulejos (III)	71
7.1.5.1. Área de Picking del inicio de la línea de cocción	73
7.1.5.2. Entrada de agentes en el Área de Picking del inicio de la línea de cocción	74
7.1.5.3. Salida de agentes en el Área de Picking del inicio de la línea de cocción	77
7.1.6. Línea de cocción	79
7.1.6.1. Pulmones en la línea de cocción	80
7.1.7. Línea de clasificación y embalaje	80
7.2. Implementación del experimento de optimización en el modelado de procesos	82
7.2.1. Diseño del interfaz de optimización	82
7.2.2. Definición de la función objetivo	84
7.3. Experimento de simulación del modelo	85
7.3.1. Inicio de la simulación del modelo	85
7.3.2. Comportamiento del modelo de simulación en régimen transitorio	87
7.3.3. Comportamiento del modelo de simulación en régimen permanente (periodo entre semana) .	89
7.3.4. Comportamiento del modelo de simulación en régimen permanente (periodo de fin de	96
semana).	
7.3.5. Finalización de la simulación	97
7.3.6. Experimento de optimización	98
8. Verificación de los resultados	100
8.1. Desarrollo del experimento	100
8.1.1. Desarrollo de la simulación durante el periodo de fin de semana	101
8.2. Medidas de flujo básicas de producción de los bloques que componen el flujo de procesos durante el periodo de entre semana.	102
8.3. Análisis de los resultados del experimento de optimización	103
8.3.1. Caso 1: Velocidad de los LGV 5 m/s	103
8.3.2. Caso 2: Velocidad de los LGV 7,5 m/s	103
8.3.3. Velocidad de los LGV 5 m/s considerando paradas por recarga de baterías	104

1. OBJETO Y CAMPO DE APLICACIÓN

Como consecuencia de las últimas crisis económicas y sanitaria, la industria cerámica ha tenido que acometer toda una serie de ajustes en la producción para hacer frente a la variación de la demanda. Ante este escenario, muchas empresas se ven obligadas a hacer frente a una posible recesión, optimizando el uso de sus recursos productivos. Una posible línea de actuación puede consistir en adecuar la infraestructura productiva mediante la adaptación o la modificación de las plantas ya existentes. De esta manera, se puede reducir la cantidad de líneas de producción operativas, y concentrar en ellas toda la producción.

Para poder llevar a cabo estos ajustes con garantías es fundamental contar con herramientas de análisis avanzadas que permitan dimensionar y gestionar los recursos de la planta de producción de una forma eficiente, en concreto los recursos de almacenamiento intermedio.

Para ello se plantea poner a punto una herramienta de simulación que permita optimizar los recursos de almacenamiento intermedio, garantizando la continuidad del ritmo de producción de una línea de producción cerámica y, en el caso de haber variaciones debidas a cambios o ajustes en la línea, poder reajustar estos recursos mínimos garantizando de nuevo la continuidad de la producción.

2. ALCANCE

En este proyecto se procederá a la creación de un modelo de simulación mediante el software Anylogic, donde se reproduzca el comportamiento real de una línea de producción cerámica. Quedan fuera del alcance de este proyecto los procesos de fabricación relacionados con la obtención de materias primas, el almacenamiento de las materias primas y la preparación de las materias primas. También queda fuera del alcance de este proyecto los procesos de fabricación de rectificado y pulido de azulejos. Por lo tanto, se considerará producto acabado a las baldosas resultantes del proceso de cocción.

El modelo se validará mediante un caso de aplicación en una empresa ficticia, Azulejos S.A., la cual tiene como fin representar una empresa tipo del sector cerámico. Es por ello que el modelo computacional a desarrollar presentará una estructura básica y simplificada del proceso de fabricación de azulejos.

Concluido el modelado del proceso de producción cerámico, se validará este modelo de simulación en base a los requisitos establecidos; y a continuación se procederá al diseño de un experimento (dentro de la aplicación Anylogic) que permita optimizar la utilización de los elementos encargados de realizar las tareas de almacenamiento intermedio de azulejos dentro del modelado de procesos (LGV o Laser Guided Vehicle). El objetivo de este experimento, será reducir el número de estos elementos para maximizar su utilización sin que los tiempos de producción se vean afectados.

3. ANTECEDENTES

3.1. EL PROCESO CERÁMICO

3.1.1. INTRODUCCIÓN

En general, el término «cerámica» (productos cerámicos) se utiliza para materiales inorgánicos (que pueden tener parte de contenido orgánico) formados por compuestos no metálicos y estabilizados mediante un proceso de cocción. Además de los materiales a base de arcilla, en cerámica existen hoy en día multitud de productos con una pequeña porción de arcilla o sin ningún contenido de la misma. Asimismo, la cerámica puede ser esmaltada o no esmaltada, porosa o vitrificada [1].

La cocción de cuerpos cerámicos produce una transformación de los minerales constituyentes, que depende del tiempo y de la temperatura, y que, en general, da lugar a una mezcla de nuevos minerales y fases vítreas. Entre las propiedades más características de los productos cerámicos figuran la elevada resistencia, la resistencia al desgaste, una larga vida útil, la inercia química y la inocuidad, la resistencia al calor y al fuego, la resistencia eléctrica (generalmente) y a veces también una porosidad específica [1].

Los principales pasos en la fabricación de productos de cerámica suelen ser bastante independientes de los materiales utilizados y del producto final. El proceso consta de los siguientes pasos: extracción de materias primas y su transporte a la instalación de fabricación de productos cerámicos (este documento no trata ninguno de estos dos pasos), almacenamiento de materias primas, preparación de materias primas, moldeado, secado, tratamiento de la superficie, cocción y tratamiento posterior [2].

3.1.2. CERÁMICA APLICADA A PAVIMENTOS Y REVESTIMIENTOS

Las baldosas de cerámica (véase EN 14411) son losas finas fabricadas a partir de arcillas y/u otros materiales inorgánicos que suelen utilizarse para cubrir suelos y paredes. Las baldosas de cerámica suelen moldearse por extrusión o prensado en seco a temperatura ambiente, seguidamente se secan y posteriormente se someten a un proceso de cocción a temperaturas suficientes para obtener las propiedades deseadas [1].

Las baldosas suelen ser cuadradas y rectangulares, pero también hay baldosas con formas poligonales (hexágonos, octágonos, etc.). En cuanto al tamaño, los lados de las baldosas van de unos pocos centímetros (mosaicos) a losas de hasta 60-100 cm. El grosor de una baldosa puede ser de sólo 5 mm (azulejos) a más de 25 mm (algunas baldosas extrusionadas) [1].

En la actualidad, hay una amplia gama de baldosas de cerámica disponible en el mercado: formadas por prensado en seco o extrusión; con cuerpos porosos, compactos o vitrificados; con cuerpos blancos o de color (rojizos); así como esmaltadas o sin esmaltar [1].

Los tipos de baldosas de cerámica fabricadas en los Estados miembros de la Unión Europea son similares; sin embargo, difieren en algunas características debido a motivos culturales, formales, funcionales, comerciales o técnicos. Por lo tanto, resulta difícil establecer la equivalencia entre los diferentes tipos de productos y sus designaciones en los distintos Estados miembros. Para superar esta dificultad, se ha adoptado una clasificación general de baldosas de cerámica en la normalización europea e internacional. De conformidad con EN 14411, las baldosas de cerámica se dividen en nueve grupos según su método de fabricación (moldeo) y su capacidad de absorción de agua. La absorción de agua se asocia con la porosidad del material: una absorción de agua elevada equivale a una alta porosidad, mientras que una baja absorción de agua se asocia con una estructura compacta y vitrificada [1].

La industria europea vende baldosas por todo el mundo y es, con diferencia, el mayor exportador. Una cuarta parte de la producción europea se exporta a países de fuera de la UE-15: la industria europea cuenta con 3/4 partes del comercio internacional de baldosas. En 2001 la industria europea vendió cerca de 1.400 millones de m² de baldosas por un valor total de 10.000 millones de euros. Esta industria dió trabajo a unas 71.000 personas durante ese mismo año [3, CERAME-UNIE, 2003]. Una parte importante de la industria se concentra en dos regiones: la región del Sassuolo (Emilia-Romaña) en Italia y Castellón (Comunidad de Valencia) en España [1].

Para la fabricación de baldosas se utilizan arcillas muy refinadas, que en muchos casos proceden de la propia Europa. Además, para el esmaltado de las mismas se emplean diversas sustancias. La energía también puede considerarse una materia prima, pues transforma las arcillas en cerámica mediante la cocción. Las fuentes de energía que se utilizan son principalmente el gas natural y la electricidad [1].

3.1.3. PROCESOS Y TÉCNICAS APLICADOS EN LA FABRICACIÓN CERÁMICA

3.1.3.1. Descripción general del proceso de fabricación

Por norma general, el proceso se inicia con las materias primas se mezclan, se moldean y se presan hasta darles forma. El agua se utiliza usualmente para conseguir una mezcla y un moldeado precisos; esta agua se evapora en secaderos. A continuación, se procede a la introducción de la pieza resultante en una línea de esmaltado para poder aplicar sobre ésta una capa de esmalte. Una vez se ha aplicado la capa de esmalte, la pieza se coloca en el horno o túnel mediante rodillos que funcionan de forma continua. En la mayoría de los casos, los hornos se calientan con gas natural, pero también se puede utilizar el gas licuado de petróleo, fueloil, carbón, coque de petróleo, biogás/biomasa o directamente electricidad [1]. Un esquema simplificado del proceso de fabricación de baldosas cerámicas es ilustrado en la Figura 1.

Durante el proceso de cocción en el horno se obtiene la estructura cerámica irreversible del producto. Esto requiere un gradiente de temperatura muy preciso para conseguir el tratamiento adecuado de los productos. Después, es necesario un enfriamiento controlado, de manera que los productos liberen el calor gradualmente y conserven su estructura cerámica. Por último, se finaliza con el embalaje y almacenamiento de los productos para su expedición [1].



Figura 1. Proceso de fabricación de baldosas cerámicas [5].

3.1.3.2. Moldeado/conformado de piezas

Los productos cerámicos tradicionalmente se moldeaban a partir de materias primas en estado plástico y durante milenios se realizaron de forma manual. Actualmente existen especificaciones técnicas cada vez más exigentes en cuanto a propiedades tales como la tolerancia dimensional, la densidad, la resistencia, la durabilidad o la refractariedad. El método escogido para conformar los materiales puede influir considerablemente en las propiedades finales del producto cerámico y es por ello que con los años han surgido numerosas técnicas de moldeado en los diversos sectores de la industria cerámica [1].

La mayoría de la oferta actual de baldosas cerámicas se fabrica por prensado semiseco o por extrusión. El presente estudio se centrará en el primero de los procesos de moldeado y conformado mencionado y será detallado seguidamente. En la Figura 2 se muestran dos ejemplos de prensa hidráulica para prensado semiseco.



Figura 2. Prensas hidráulicas de prensado semiseco.

- **Prensado semiseco**

Las materias primas, en forma de bolitas huecas con una humedad de entre el 5% y el 7% o en forma de granos macizos con una distribución controlada de tamaño de grano, se someten a un prensado mecánico que da forma definitiva a la baldosa cerámica mediante una presión homogénea de entre 250 Kg/cm² y 500 Kg/cm². La optimización de este proceso mecánico ha llevado a la fabricación de baldosas cada vez más delgadas y con mayor uniformidad en cuanto a la densidad de materia por unidad de volumen además de alcanzar una alta resistencia mecánica en crudo. Esa homogeneidad de la materia prima prensada permite un secado igualmente uniforme y evita todos los efectos derivados del comportamiento desigual de la materia durante los procesos de secado y cocción [3].

La alta resistencia en crudo, ha permitido también optimizar los tratamientos superficiales realizados sobre la pieza sin necesidad de someterla a la primera cocción. En cualquier caso, el prensado en semiseco no sólo ha permitido avanzar en el proceso de fabricación por monococción, sino también aumentar considerablemente el formato de las piezas sin disminución de la calidad ni de las características del producto final [3].

3.1.3.3. Secado de productos cerámicos

Tradicionalmente la mayoría de artículos de arcilla se secaban de forma natural, dejándolos al aire a temperatura ambiente. En muchas zonas de Europa este procedimiento sólo era eficaz durante el verano. En la industria tradicional de los ladrillos y las tejas se empleaban baldas de secado simples que permitían colocar filas de ladrillos unos encima de otros bajo una sencilla cubierta ligera de madera que los protegía de la lluvia [1].

La producción actual de baldosas cerámicas ha incorporado equipamientos de secado que aúnan una gran productividad y uniformidad en la operación de pérdida de agua del cuerpo cerámico. La productividad es la adecuada tanto para lo que se suministra al secadero como para lo que se sirve después de él en las siguientes operaciones de fabricación. La uniformidad en el secado, absolutamente imprescindible para evitar roturas y deformaciones en la pieza, es alcanzada mediante una correcta gestión del flujo de aire caliente y la separación física de las piezas para permitir que la pérdida de agua sea homogénea en toda la sección y superficie de la baldosa cerámica [3].

La regularidad y la uniformidad en la operación de secado es fundamental, junto con el modelado y la cocción, para mantener constantes las dimensiones del producto final, a la par que evitar diferentes defectos derivados de la falta de uniformidad del secado, como el exceso de agua en la pieza cruda inmediatamente antes de la cocción [3].

- **Secaderos de productos cerámicos**

Durante el proceso de secado de las baldosas cerámicas, los cuerpos prensados se secan principalmente en secaderos de túnel, secaderos de rodillos o secaderos verticales, como el que muestra en la Figura 3. El calor residual del horno o de los quemadores de cocción de gas natural y fueloil se utiliza para calentar el secadero. El secado se lleva a cabo a temperaturas que varían según el tipo de tecnología; en los secaderos verticales está entre 200-220 °C y en los secaderos de túnel va de 300 a 350 °C. El tiempo de secado se sitúa entre 15-25 minutos para los secaderos horizontales, y entre 35-60 minutos para los secaderos verticales; sin embargo, la duración depende del tipo de composición, de la dimensión y del espesor de las piezas. El contenido de humedad residual necesario es inferior al 1% para evitar fisuras y errores de esmaltado durante el proceso de cocción [1].

Por otro lado, la fisionomía de un secadero vertical (objeto de este proyecto) se compone básicamente de un armazón vertical, dentro del cual se desplazan cestas arrastradas por cadenas compuestas por diferentes planos de rodillos o bastidores, donde se sitúan las baldosas cerámicas a secar. Estas baldosas se cargan automáticamente sobre los planos de las cestas mediante la rotación de los rodillos que las componen, lo que permite la concatenación de este módulo con otros dispuestos en serie



Figura 3. Secadero rápido vertical de productos cerámicos.

3.1.3.4. Esmaltación y decoración de productos cerámicos

Los engobes y los vidriados (comúnmente denominados esmaltes) se aplicaban en el pasado sobre el cuerpo cerámico ya cocido, normalmente poroso (lo cual permitía absorber el agua de esas aplicaciones), bien antes de su decoración o después de ésta. Dichas aplicaciones eran totalmente manuales y solamente en el caso de decoraciones seriadas se contaba con elementos auxiliares que permitían reproducir los diseños de forma constante; éste es el caso de los estarcidos para siluetear los grafismos o las trepas para rellenar superficies con un determinado contorno. Es precisamente de este hecho de donde proviene la denominación de “pintador” comúnmente empleada para llamar a la sección de una fábrica de baldosas cerámicas que efectúa las operaciones de esmaltado y aplicación de las decoraciones en parte de su producción [3].

En el proceso de esmaltado, se preparan componentes de esmaltado bien molidos (principalmente fritas en el caso de las baldosas) como suspensiones acuosas que posteriormente son aplicadas en la pieza mediante

técnicas de esmaltado. Estas técnicas de esmaltado se dividen en dos grupos:

- **Esmaltado a cortina:** Consiste en hacer pasar la baldosa cerámica por debajo de una cortina de esmalte para aplicar una capa de dicho material. El espesor de la cantidad de esmalte aplicado va a depender de la velocidad de paso del azulejo y de la fisiología de la cortina de esmalte generada.
- **Esmaltado mediante pulverización:** Consiste en disparar pequeñas cantidades de esmalte en las piezas a partir de un sistema de pulverización a presión. El espesor de la cantidad de esmalte aplicado va a depender de la velocidad de paso del azulejo y de la presión con la que actúe el sistema de pulverización.

En esta sección del proceso de fabricación de baldosas cerámicas la innovación más importante es la posibilidad concatenar en una única línea de esmaltado o cinta transportadora todas las aplicaciones que se efectúen sobre la baldosa cerámica en estado crudo (pulverización, esmaltado a cortina, esmaltado seco o decorativo). En la Figura 4 se ofrece la representación de una línea de esmaltado en perspectiva. Esta estrategia de línea de esmaltado permite la alimentación continua del soporte crudo recién salido de los secaderos continuos; y, por otra parte, también permitirá dar alimentación ininterrumpida en el proceso de fabricación por monococción.



Figura 4. Perspectiva de una línea de esmaltado.

- **Máquinas de impresión digital**

La incorporación de la técnica de impresión digital ha enriquecido notablemente la diversidad de las decoraciones obtenidas bajo proceso industrial y antes de la cocción. Gracias a este tipo de tecnología, es posible decorar las piezas cerámicas directamente y sin contacto alguno.

La técnica de impresión digital de azulejos consiste en inyectar la pintura directamente sobre la baldosa, consiguiendo la impresión total de relieves y otorgándoles a los diseños naturalidad. Para conseguir este resultado, primero es necesario realizar un renderizado de la trepa que se quiera dibujar sobre el azulejo. A continuación, se transforma este renderizado en señales eléctricas y se cargan en las controladoras que manejan las tarjetas electrónicas que disparan las válvulas piezoeléctricas de los cabezales de pintura. Por último, la controladora de los tarjeteros de los cabezales dispara cada gota de pintura sobre la pieza, reproduciendo con exactitud el diseño de la trepa. Para ello, se ayuda de un sensor de detección de pieza al inicio del módulo y de un encoder para poder calcular la posición exacta de cada disparo de pintura. Mediante esta técnica de

decoración, se obtienen resultados de altísima calidad y fiabilidad en pequeños y grandes formatos.

Este tipo de maquinaria está concebida mediante un diseño modular, tal y como se muestra en la Figura 5. Dicho tipo de diseño permite concatenar estos equipos con la línea de esmaltado, permitiendo la posibilidad hacer cambios en la decoración de las piezas insitu desde el propio software de la máquina y sin detener el proceso productivo.



Figura 5. Máquina de impresión digital KeraJET S7.jpg

3.1.3.5. Almacenamiento intermedio

Una parte muy importante a la hora de analizar el proceso de fabricación cerámico es la detección de las necesidades de almacenamiento intermedio que tienen este tipo de sistemas, ya que el proceso de fabricación por monococción se caracteriza por adoptar una forma de cuello de botella cuyo efecto se ve intensificado por los procesos de fabricación concatenados de prensado-secado-esmaltado que le preceden, con una mayor tasa de producción. Esto supone la necesidad de almacenar el excedente resultante de los procesos de fabricación concatenados de prensado-secado-esmaltado en un almacén intermedio a la espera de que sea admitido en el proceso de monococción.

Por otra parte, si el equipamiento encargado de llevar a cabo el proceso de monococción (el horno), se queda en algún momento sin alimentación, la temperatura en su interior comenzaría a aumentar de manera descontrolada y los rodillos de transporte de piezas que hay instalados en su interior pueden llegar a fundirse y dejar el horno inutilizable. La detención del funcionamiento del horno tiene un impacto económico significativo derivado a la ingente cantidad de tiempo para su puesta en marcha, durante el cual no sería posible producir nada, y que conllevaría las pérdidas equivalentes a tal producción perdida.



Figura 6. Boxes de almacenamiento de bizcocho a la espera de ser cocidos en el horno.

- **Boxes de almacenamiento intermedio**

Los boxes son vagonetas en forma de prisma cuadrangular formadas por un conjunto de planos donde se almacenan las baldosas cerámicas a la espera de ser admitidas en el siguiente proceso de fabricación. En la Figura 6 se muestra un parque de boxes de almacenamiento de bizcocho a la espera de ser cocidos en el horno.

- **Vehículo de Guiado por Laser (LGV)**

Para poder manipular y transportar los boxes entre los distintos emplazamientos dentro de la planta industrial dónde se desarrollan los distintos procesos de fabricación y el almacén de estas vagonetas, se emplean vehículos de transporte automatizados o LGVs. En la Figura 7 se aprecia como un vehículo guiado manipula un box de almacenamiento intermedio.

Estos vehículos se encargan de dar soporte a las distintas secciones de la planta industrial en función de las necesidades de producción de cada una de ellas, de almacenar o de procesar producto semielaborado. Estos vehículos se desplazan de manera totalmente autónoma y su funcionamiento es automático.



Figura 7. LGV desplazando un box por el parque de boxes.

3.1.3.6. Proceso de cocción de productos cerámicos

La cocción es un proceso clave en la fabricación de productos cerámicos, ya que controla muchas propiedades importantes de las piezas acabadas. Estas propiedades incluyen: la resistencia mecánica, la resistencia a la abrasión, la estabilidad dimensional, la resistencia al agua y a los productos químicos y la resistencia al fuego [1].

- **Cambios físico-químicos durante la cocción**

Las materias primas utilizadas en cuerpos cerámicos suelen ser mezclas complejas de minerales arcillosos con otras sustancias minerales tales como el cuarzo, feldespato, carbonatos, yeso, óxidos de hierro y, en ocasiones, materia orgánica. Los productos refractarios también están hechos de una amplia variedad de minerales no arcillosos, junto con aditivos y aglutinantes especializados (que pueden incluir algunas arcillas). Cuando los productos cerámicos a base de arcilla se cuecen en un horno, la humedad residual se extrae a

temperaturas entre 100 y 200 °C. Si tienen sustancias orgánicas y piritas de hierro, la oxidación tiene lugar a temperaturas entre 300 y 500 °C. El agua combinada con la estructura de los minerales arcillosos (agua de cristalización) suele eliminarse a temperaturas entre 500 y 650 °C, mientras que los carbonatos como la calcita y la dolomita se disocian con la liberación de dióxido de carbono a temperaturas entre 750 y 950 °C [1].

Los cambios más importantes relacionados con el desarrollo de las propiedades cerámicas implican la ruptura de la estructura reticular de los minerales arcillosos originales, seguidos por la formación de nuevos compuestos cristalinos y fases vítreas. La temperatura a la cual tiene lugar la vitrificación (formación de cristales) varía según la mineralogía de la arcilla. La vitrificación normalmente se inicia cuando se alcanzan los 900 °C y se completa hacia los 1.050 °C o a los 1.100 °C en el caso de las arcillas más refractarias [1].

Durante la fase de vitrificación de la cocción cerámica, se incorporan muchos minerales no arcillosos como el cuarzo, los óxidos o el hierro, compuestos de cal y álcalis (óxidos de sodio y potasio). Se da alguna solución sólida y de sinterizado y tienen lugar reacciones eutécticas en la interfaz de los granos minerales y las fases de fusión. Los productos no arcillosos, como algunos productos refractarios, también dependen de las fases de sinterizado, vitrificación o recristalización, pero en la mayoría de casos se necesitan temperaturas mucho más elevadas para obtener las propiedades deseadas [2][4].

- Hornos de solera con rodillos

Los hornos de rodillos de plataforma, como el que se muestra en la Figura 8, se utilizan habitualmente en la producción de pavimentos y revestimientos. Por lo que respectan los tiempos de cocción, éstos se han reducido a menos de 40 minutos. Las baldosas se transportan sobre rodillos y la cocción se lleva a cabo con quemadores de aire-gas natural ubicados a los costados del horno. Los principales mecanismos de transmisión térmica son la convección y la radiación. Dado que no son hornos de mufla, los coeficientes de transmisión térmica se elevan, reduciendo el ciclo de cocción y el consumo de energía. Los hornos de solera con rodillos se utilizan en ocasiones para la producción de tejas de arcilla, tuberías de gres vitrificado, cerámica sanitaria y de mesa. [2][4].



Figura 8. Horno de rodillos de plataforma.

3.1.3.7. Clasificación y embalaje

Los métodos aplicados de control de calidad en los procesos de la cocción han reducido considerablemente la rotura de piezas y los sistemas están cada vez más automatizados. Por este motivo, en algunas empresas, las tareas de clasificación y el embalaje siguen siendo operaciones manuales llevadas a cabo por los operarios. Sin embargo, en organizaciones donde se gestionan grandes volúmenes de producción, es necesario automatizar el proceso de inspección final y clasificación para seguir manteniendo la eficacia en esta tarea sin afectar al rendimiento productivo. Actualmente es posible medir instrumentalmente incluso el color, lo que resulta importante en las baldosas de cerámica.

- **Inspección de la planitud de la baldosa cerámica: Planar**

El planar consiste en una serie de sensores ópticos que realizan un muestreo de medidas de los puntos de la superficie de la baldosa. En la Figura 9 se aprecia claramente como 5 sensores ópticos de estas características están realizando la tarea de inspección de la pieza. Estas medidas se basan en la distancia entre el punto medido de la superficie y la del sensor óptico, donde la posición del sensor siempre es fija y sirve para referenciar las medidas tomadas. Cada medida se ubica en unas coordenadas creadas a partir de las dimensiones del azulejo. Una vez ha finalizado el muestreo, se obtiene un mapeado preciso de la topografía del azulejo.



Figura 9. Máquina de inspección de planitud DICO Planar de System.

Una vez se ha obtenido el mapa topográfico de la baldosa cerámica inspeccionada, se comparan los valores medios y se clasifica en función del resultado. Este tipo de equipamiento se puede integrar en la línea de producción en serie; las tecnologías existentes hoy en día permiten que este tipo de máquinas se puedan adaptar a todo tipo de formatos.

- **Inspección dimensional: Calibre**

La inspección mediante el calibre consiste en medir el tamaño de fabricación, en mm, de las baldosas, así como su tolerancia. Esta información se debe indicar siempre en el embalaje. Como muestra la Figura 10, estos equipos suelen estar compuestos por un conjunto de láseres fijos que peinan los bordes del azulejo y realizan un muestreo de sus medidas dimensionales para poder generar posteriormente un mapeado preciso de su geometría y poder clasificarlo en función de la calidad de su geometría y su tolerancia.



Figura 10. Máquina de inspección de calibre Liner 2000.

El tamaño de la baldosa cerámica puede ser superior o inferior de pocos milímetros respecto al tamaño indicado en cm en los catálogos. Según las normas UNE EN ISO vigentes, el tamaño de fabricación en mm puede tener una tolerancia del $\pm 2\%$ (como máximo de 5 mm de desviación) de la dimensión nominal en cm. Es necesario añadir que, por motivos de falta de precisión en las técnicas de fabricación de baldosas cerámicas, las rectificadas presentan dimensiones ligeramente diferentes y, por este motivo, durante el proceso de clasificación, son agrupadas en partidas (calibres) del mismo tamaño, siempre respetando los márgenes de tolerancia establecidos por la normativa de calidad.

- **Inspección de visual: Visión artificial**

La inspección visual se realiza para comprobar que realmente se ha impreso correctamente el patrón decorativo. Antiguamente esta tarea era realizada por los propios trabajadores. Éstos estaban colocados en la estación de inspección y delante de ellos iban pasando los azulejos; su tarea era detectar de manera visual qué azulejos debían ser retirados. Pasado un tiempo ejerciendo esta tarea, la fatiga visual y la distracción a causa de una operación tan monótona mermaban la capacidad de inspección del trabajador. Por ello, los turnos de trabajo en los cuales se realizaba esta actividad no podían ser muy largos. Era importante planificar muy bien las rotaciones de trabajadores para poder disponer de una capacidad de inspección efectiva.

Actualmente, la gran cantidad de la gama de decoraciones que pueden ser aplicables a las piezas cerámicas, la complejidad de las trepas que se dibujan sobre las piezas y la alta definición con las que son aplicadas gracias a las máquinas de impresión digital han motivado a que la industria haya desarrollado soluciones para poder llevar a cabo una inspección visual precisa y de manera automática. Es en este punto donde entra el concepto de visión artificial. La visión artificial consiste en una cámara de alta definición, como la que se muestra en la Figura 11, que captura la decoración del azulejo para poder comparar la estructura de su decoración con el render original aplicado por la máquina de inspección digital.



Figura 11. Máquina de visión artificial.

Esta máquina de inspección puede integrarse en la línea de producción gracias a su diseño modular. Por consiguiente, se puede intercalar este módulo de inspección con el resto, pudiendo realizar una inspección visual de alta precisión de las piezas insitu y sin detener el proceso productivo.

- **Embalaje**

Una vez se ha finalizado el procedimiento de inspección con éxito, se procede al embalaje de las baldosas cerámicas. Para ello se utilizan módulos de embalaje concatenados al final de la línea, donde directamente se empaquetan los azulejos apilados en cajas de cartón.

La tecnología existente hoy en día ha avanzado mucho en las técnicas de embalaje. Estas mejoras permiten organizar la producción en tiempo real sin necesidad de perder tiempo en el diseño de los troqueles. A día de hoy, existen equipos, como el que aparece en la Figura 12, que los crean y los personalizan directamente en su interior sin importar el tipo de baldosa que se embale, ya sea natural o rectificada. Además, es posible cambiar de formato en tiempo real y sin necesidad de ajustes manuales.



Figura 12. Máquina de embalaje System 4 Fases.

Esta nueva generación de máquinas de embalaje puede trabajar con un único formato de cartón (habitualmente 1200 x 1200 mm), y además existe la posibilidad de que la propia embaladora pueda personalizar las cajas de cartón con logotipos y marcas de varios colores. Esta característica de poder utilizar un único formato de cartón para alimentar la máquina permite reducir los costes de almacenaje de este ítem, así como poder salvar la obsolescencia de lotes de cartón que se hayan preparado y que finalmente no se vayan a utilizar.

3.2. TÉCNICAS DE MODELADO DE PROCESOS

La planificación de procesos consiste en, a partir de un diseño general, la identificación de todas las actividades individuales que son necesarias para cumplir con los objetivos del proceso y decidir la secuencia en la que estas actividades deben realizarse y quién o qué es el encargado de realizarlas. El modelado de procesos a menudo se realiza mediante un enfoque visual simple; es por ello que se hace uso de diagramas de flujo. El resultado de esa actividad se puede plasmar en unos modelos descriptivos simples, como son los diagramas de flujo.

3.2.1. DIAGRAMA DE FLUJO

Un diagrama de flujo es la representación de la secuencia del desarrollo de un proceso. En esta representación, se desglosa un proceso de cualquier complejidad en un cúmulo de operaciones simples relacionadas en donde se establece una relación cronológica entre sí. Esta técnica de modelado de la información sirve para poder comprender qué operaciones componen el proceso y para identificar las entradas de materias primas y materiales.

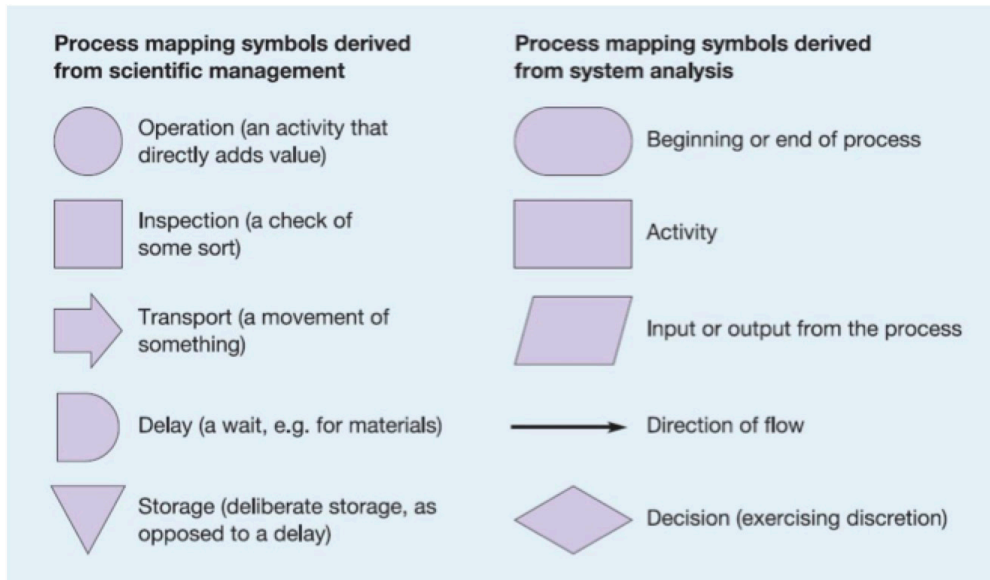


Figura 13. Símbolos comúnmente utilizados para el mapeado de diagramas de flujo. [1]

En la Figura 13 se muestran los símbolos más comúnmente utilizados a la hora de realizar diagramas de flujo. Estos símbolos se utilizan para clasificar los diferentes tipos de actividad que se pueden dar en un diagrama de flujo y, aunque no existe un conjunto universal de símbolos utilizados en todo el mundo para cualquier tipo de proceso, éstos son los que se utilizan de forma más habitual. Dentro de un diagrama de flujo, estos símbolos se pueden disponer en serie o en paralelo en función de si las actividades que se representan en el diagrama se desarrollan de manera secuencial o, por el contrario, de manera simultánea.

En el modelado de procesos complejos, el mapeado de un diagrama de flujo se hace muy complejo y poco intuitivo, por lo que en este tipo de casos se suele constituir un diagrama de flujo de alto nivel compuesto por diagramas de flujo de bajo nivel agregados. En la Figura 14 se muestra un ejemplo de un diagrama de flujo de alto nivel basado en el proceso de elaboración de emparedados.

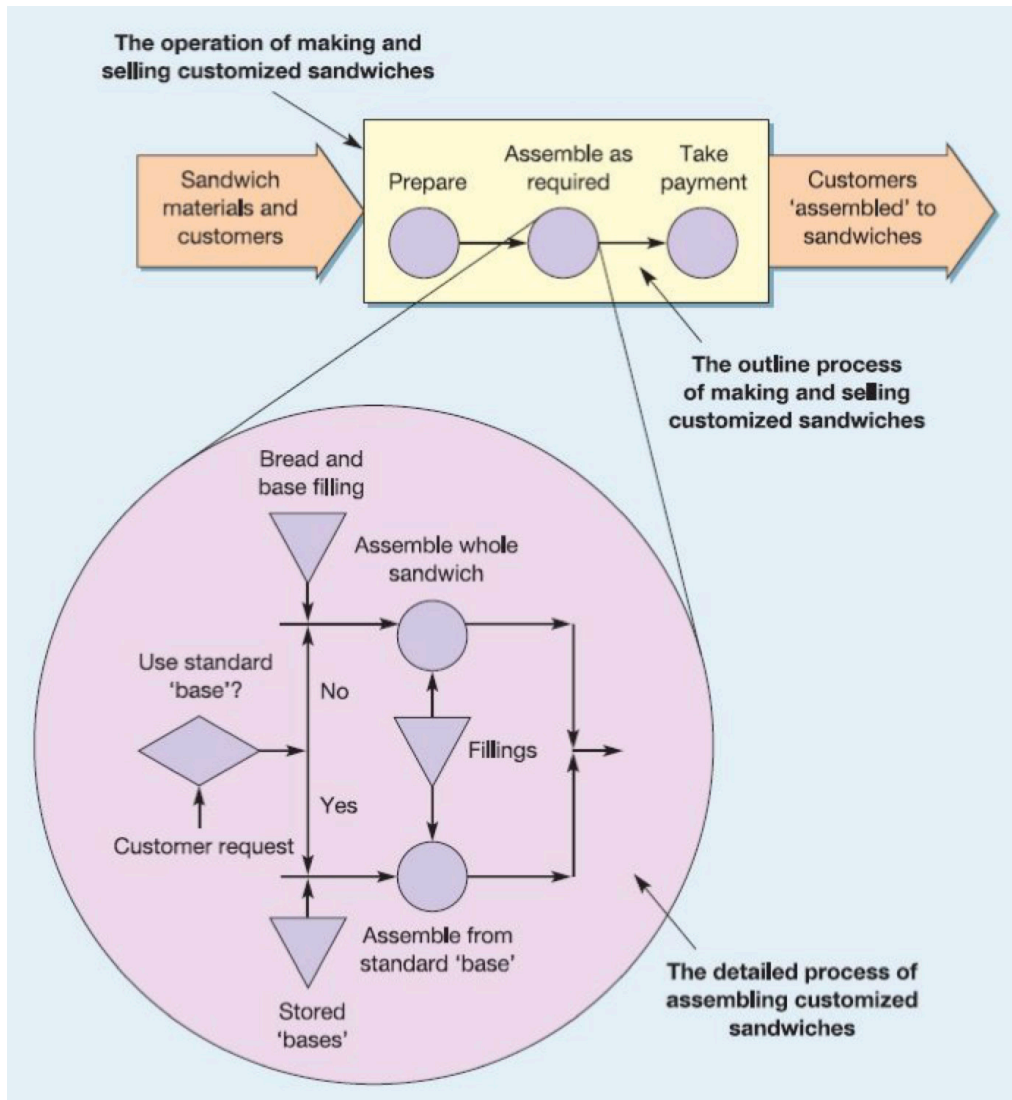


Figura 14. Proceso de elaboración de emparedados a 3 niveles por cortesía de OM (6ª edición).

3.2.2. ANÁLISIS DE LOS FLUJOS DE MATERIALES

El flujo de materiales a lo largo de un proceso de fabricación se caracteriza a partir de 3 medidas de flujo básicas:

- **TH (Throughput time):** Se conoce como tasa de producción a la cantidad media de unidades producidas por unidad de tiempo por un proceso o actividad. El TH a su vez, también puede interpretarse como tasa de llegada o la cantidad de unidades que llegan por unidad de tiempo a un proceso o actividad.
- **CT (tiempo ciclo)** es el tiempo medio de permanencia de una unidad dentro de un proceso o actividad productiva.
- **WIP (trabajos en curso o inventario en curso)** al número de unidades existentes que en término medio hay dentro de una operación o actividad a la espera de ser procesadas.

El inventario en curso de un proceso es directamente proporcional al tiempo de ciclo y, a su vez, también directamente proporcional a la tasa de producción, esta relación matemática se materializa en la expresión matemática de la ley de Little:

$$WIP=TH \cdot CT \quad (1)$$

Esta expresión matemática es la relación entre los promedios del inventario en curso, el tiempo de ciclo, y la tasa de producción del sistema.

3.3. SITUACIÓN ACTUAL DE AZULEJOS S.A.

Azulejos S.A. es un fabricante especialista en producir baldosas cerámicas de formato grande, 120X120 cm. Debido a ser una empresa muy especializada, es una empresa pequeña, su fábrica se basa en una única línea de producción, y su estrategia de producción consiste en trabajar contra stock realizando lotes de producción de 50.000 metros.

El cuello de botella del proceso de producción cerámico, se encuentra en el proceso de cocción. El horno, debido a la naturaleza de su diseño, debe funcionar de manera ininterrumpida. Si por algún motivo dejara de estar alimentado con bizcocho mientras este se encuentra en funcionamiento, este sufriría severos daños internos. Es por ello que existe un almacenamiento intermedio en forma de boxes de bizcocho (ver 3.1.3.5 Almacenamiento intermedio) con esto se consigue desacoplar la línea de producción Prensado-secado-esmaltado con la línea de cocción. Aun así, es necesario definir una estrategia de planificación de la producción que permita gestionar los niveles de este almacenamiento intermedio, ya que, si se mantiene un ritmo de producción continuo durante un tiempo prolongado, los niveles del almacenamiento intermedio del bizcocho tenderán a infinito y el almacén intermedio puede llegar a colapsar.

Azulejos S.A., adopta una solución para solventar este problema. Durante la semana se mantiene un ritmo de producción normal en la línea de Prensado-secado-esmaltado, lo cual hace que los niveles de almacenamiento intermedio de bizcocho aumenten; pero una vez llega el fin de semana, se paraliza la línea de Prensado-secado-esmaltado y el horno se encarga de cocer todo el bizcocho que se ha ido acumulando durante la semana. Cuando llega el lunes, el horno ya ha cocido prácticamente todo el bizcocho que se ha ido acumulando; y el ciclo de producción vuelve a empezar. Esta es una solución ampliamente utilizada en el sector cerámico para poder resolver esta problemática.

Si se analiza esta situación desde un punto de vista analítico se puede determinar que, durante una ventana de tiempo VT se produce un inventario acumulado IA, producto de la diferencia entre una tasa de producción de entrada y una tasa de producción de salida por la duración de la ventana en la que produce la línea y el horno:

$$IA = (TH_{IN} - TH_{OUT}) \cdot VT$$

3.4. CARACTERÍSTICAS DE LOS RECURSOS DE LA LÍNEA DE PRODUCCIÓN

En este punto, se pretende describir las características funcionales de los diferentes elementos que componen la línea de la planta de Azulejos S.A. haciendo hincapié en aquellas que influyen en el comportamiento a nivel productivo de la línea de producción cerámica.

- **Moldeado de piezas**

Para el moldeado de piezas se utiliza una prensa hidráulica de prensado en semiseco (ver punto 3.1.3.2 Moldeado/conformado de piezas) (ver Tabla 1).

Prensa hidráulica en semiseco	
Velocidad máxima de la prensa (Golpes/minuto)	3,5
Capacidad de la plancha (Piezas/golpe)	1

Tabla 1. Características de la prensa hidráulica en semiseco.

- **Secado de piezas:**

El secado de las piezas se lleva a cabo a partir de un secadero vertical (ver punto 3.1.3.3 Secado de productos cerámicos) (ver Tabla 2). El secadero vertical dispone a su salida un almacén intermedio a su salida (Pulmón) con capacidad de absorber todo su WIP en caso de necesidad.

Secadero vertical	
Ciclo de secado de las piezas cerámicas (min)	40
Número máximo de cestas	140

Tabla 2. Características del secadero vertical.

- **Esmaltado**

El proceso de esmaltado de las piezas cerámicas se realiza a partir de una línea de esmaltado con tecnología de pulverización y que además cuenta con una máquina de impresión digital para realizar la decoración de las piezas (ver punto 3.1.3.4 Esmaltación y decoración de productos cerámicos) (ver características principales en Tabla 3). La línea de esmaltado dispone a su salida un almacén intermedio a su salida (Pulmón) con capacidad de absorber todo su WIP en caso de necesidad.

Línea de esmaltado	
Longitud (m)	120
Velocidad de la línea (m/min)	50
Nivel de merma (%)	5

Tabla 3. Características de la línea de esmaltado.

- **LGV**

Los LGV encargados de realizar el movimiento de boxes para el almacenamiento intermedio de bizcocho (ver 3.1.3.5 Almacenamiento intermedio) se caracterizan únicamente por su velocidad (ver Tabla 4).

LGV	
Velocidad máxima del LGV (m / s)	5

Tabla 4. Características de las líneas de esmaltado.

- **Boxes**

Los boxes o vagonetas donde se almacena el bizcocho que sale de la línea de Prensado-secado-esmaltado (ver 3.1.3.5 Almacenamiento intermedio) se caracterizan en la Tabla 5.

Boxes	
Número máximo de boxes	120
Número de planos	45
Azulejo por plano (formato de 120x120 cm)	1

Tabla 5. Características de los boxes.

- Horno cerámico

La cocción de las piezas cerámicas se lleva a cabo a partir de un horno de solera con rodillos ver (3.1.3.6 Proceso de cocción de productos cerámicos) (ver Tabla 6). El horno de solera con rodillos dispone a su entrada y a su salida un almacén intermedio (Pulmón) con capacidad de absorber un tercio de su WIP en caso de necesidad.

Horno cerámico	
Longitud del horno (m)	240
Ciclo de cocción (min)	75
Nivel de merma (%)	5

Tabla 6. Características del horno cerámico.

- Almacén intermedio de bizcocho

Está localizado entre el fin de la línea de Prensado-secado-esmaltado y el inicio del horno cerámico. Este almacén se divide en dos partes, una para albergar boxes vacíos y otra para albergar los boxes llenos de bizcocho. Cada parte del almacén consta de dos pasillos de 550m con capacidad para albergar 30 boxes a ambos lados del pasillo.

Por otro lado, al final de la línea de esmaltado, hay un área de picking donde se cargan los azulejos en los planos de los boxes para después ser recogidos y transportados al almacén intermedio de bizcocho (ver Figura 15).

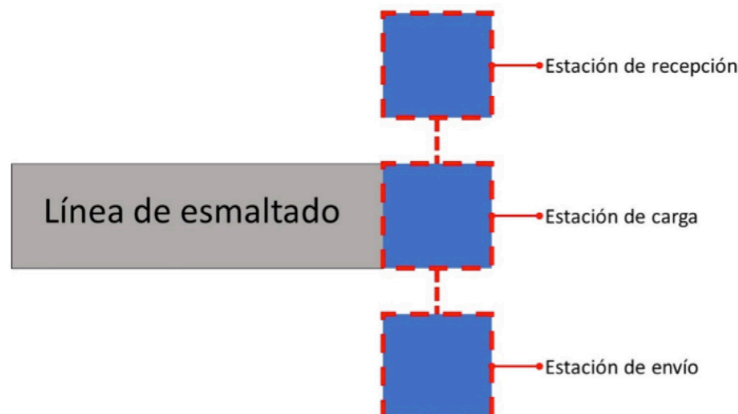


Figura 15. Área de picking del final de la línea de esmaltado.

Su funcionamiento es sencillo, cuando el sistema está trabajando en régimen permanente, siempre hay un box en la estación de recarga llenándose con las piezas que salen de la línea de esmaltado y un box vacío en espera localizado en la estación de recepción. En el momento en el que el box que se encuentra en la estación de carga se ha llenado, mediante un mecanismo automático, este pasa a la estación de envío a la espera de ser recogido y transportado por un LGV al almacén intermedio de bizcocho; y simultáneamente, el box que se encontraba en la estación de recepción, pasa automáticamente a la estación de carga de azulejos. A continuación, se envía una orden al parque de LGV para que transporten y depositen un box vacío en la estación de recepción.

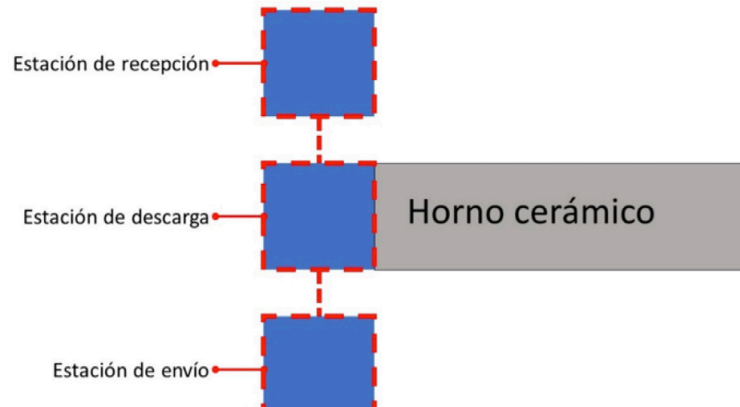


Figura 16. Área de picking del inicio del horno cerámico.

Este mismo sistema se aplica a la línea de cocción; al inicio del horno cerámico, hay un también un área de picking donde se realiza la operación de descarga del bizcocho almacenado en los boxes (ver Figura 16).

En este caso, cuando el sistema está trabajando en régimen permanente, siempre hay un box en la estación de descarga alimentando al horno con bizcocho, y un box lleno localizado en la estación de recepción. En el momento en el que el box que se encuentra en la estación de descarga se ha vaciado, automáticamente pasa a la estación de envío a la espera de ser recogido y transportado por un LGV al almacén intermedio de boxes vacíos; y simultáneamente, el box que se encontraba en la estación de recepción, pasa automáticamente a la estación de descarga de bizcocho. A continuación, se envía una orden al parque de LGV para que transporten y depositen un box lleno en la estación de recepción.

- **Clasificación**

La sección de inspección consta de una línea compuesta de 3 etapas: Un planar, un calibre y una máquina de visión artificial (ver punto 3.1.3.7 Clasificación y embalaje) Estas etapas están dispuestas en serie y determinan el nivel de calidad de los azulejos producidos.

- **Embalaje**

La máquina empaquetadora (ver punto 3.1.3.7 Clasificación y embalaje) es un módulo concatenado al final de la sección de clasificación que genera cajas a partir de 25 piezas cerámicas apiladas.

3.4.1. DIAGRAMA DE FLUJO DE LA LÍNEA DE FABRICACIÓN DE AZULEJOS S.A.

En la Figura 17 se ilustra el diagrama de flujo (ver punto 3.2.1 Diagrama de flujo) de la línea de producción cerámica constituido a partir del listado de operaciones de producción necesarias para poder desarrollar el proceso de fabricación de baldosas cerámicas. La representación de los procesos que se representan en el diagrama está basada en la simbología que se muestra en la Figura 13.

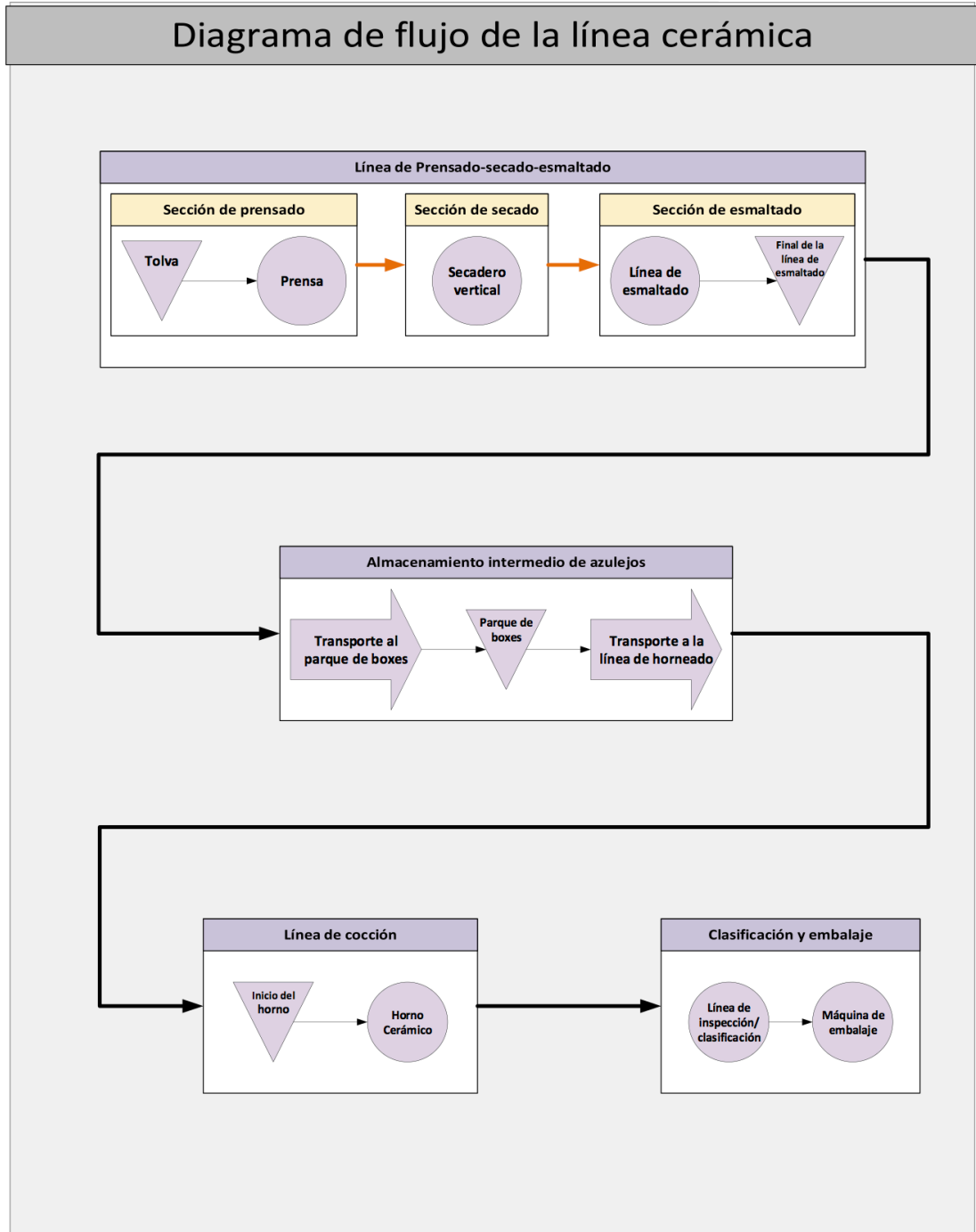


Figura 17. Diagrama de flujo de la línea de producción cerámica.

3.5. OPERACIONES DE PRODUCCIÓN EN LA LÍNEA DE PRODUCCIÓN CERÁMICA DE AZULEJOS S.A.

En la Tabla 7 se recogen las operaciones de producción necesarias para poder desarrollar el proceso de fabricación de baldosas cerámicas en base a la caracterización de la distribución en planta de Azulejos S.A. desarrollada en los puntos 3.4. y 3.5. En ellos se describe, de manera cualitativa, dichas operaciones en términos de medidas de flujo básicas (ver 3.2.2 Análisis de los flujos de producción).

Ello permite las medidas del flujo de materiales entre las distintas etapas atendiendo a las características funcionales de cada una de ellas.

Id	Operación	CT (tiempo de ciclo)	WIP (inventario en curso)	TH (tasa de producción)
1	Prensado	Tiempo necesario para procesar toda la tierra de la tolva.	Material de la tolva	Golpes por unidad de tiempo de la prensa.
2	Secado	Ciclo de secado de las piezas	Piezas dentro de las cestas del secadero.	Tasa de llegada de piezas desde la prensa.
3	Esmaltado	Tiempo del recorrido de una pieza a través de la línea de secado.	Piezas recorriendo la línea de esmaltado.	Tasa de llegada de piezas desde el secadero.
4	Llenado de boxes	Tiempo de llenado de un Box.	Cantidad de azulejos por box.	Tasa de llegada de piezas desde la línea de esmaltado.
5	Transporte del box al parque	Tiempo medio de recorrido de un box desde el área de picking de la línea de esmaltado hasta el parque de boxes.	Cantidad de boxes por transportar.	Tasa de producción del área de picking de la línea de esmaltado
6	Transporte del box horno	Tiempo medio de recorrido de un box desde el parque hasta el área de picking del horno.	Cantidad de boxes por transportar.	Tasa de producción del área de picking del horno.
7	Vaciado de boxes	Ciclo de descarga de un box.	Cantidad de piezas por descargar	Tasa de producción del horno.
8	Cocción	Ciclo de cocción.	Piezas dentro del horno	Tasa de producción del horno.
9	Inspección y clasificado	Tiempo de inspección.	Piezas dentro de la línea.	Tasa de llegada de piezas desde el horno.
10	Embalado	Ciclo de embalaje.	Piezas por caja.	Tasa de llegada de piezas desde la línea de inspección.

Tabla 7. Listado de las operaciones de producción que componen el proceso de producción de la línea cerámica.

3.5.1. MEDIDAS DE FLUJO BÁSICAS DE LAS OPERACIONES DE PRODUCCIÓN.

El proceso de producción cerámico se caracteriza por tener una baja variabilidad cuando se encuentra en régimen permanente. Sin contar efectos como cambios de formato, ajustes en la línea de esmaltado, o paradas de mantenimiento, por ejemplo; el funcionamiento de la línea de producción es muy constante y no hay apenas variabilidad en las distintas etapas que componen el proceso. Por ese motivo, se va a adoptar un

enfoque determinista y no se va a considerar variabilidad a la hora de determinar los niveles deterministas de las medidas de flujo básicas de las operaciones de producción; únicamente se van a determinar los valores medios.

3.5.1.1. Medidas de flujo deterministas del almacén intermedio I.

Según la estrategia de planificación de la producción descrita en el Punto 3.3, dedicado a la Situación actual de Azulejos S.A, al finalizar la ventana de tiempo de entre semana el nivel del inventario acumulado del almacenamiento intermedio de bizcocho debe alcanzar su nivel máximo. El incremento del nivel de este IA se define como proporcional a la diferencia del TH de llegada del almacén (Tasa producción de la línea de Prensado-secado-esmaltado) y el TH de salida del almacén (Tasa de producción del horno) (ver Ecuación 1) dentro de esa ventana de tiempo.

$$IA_{Bizcocho} = (TH_{P-S-E} - TH_{HORNO}) \cdot VT_{Entre\ semana}$$

Ecuación 1. Definición del IA del almacenamiento intermedio

Por otra parte, en el almacén intermedio de la planta cerámica de Azulejos S.A. se dispone de 2 pasillos con capacidad para albergar 30 boxes a cada lado del pasillo (ver punto 3.4 Caracterización los Recursos de la línea de producción); por lo tanto, se tiene una población de 120 boxes. Cuando el proceso de producción se encuentra en régimen permanente, siempre habrá 2 boxes en la zona de picking del final de la línea de esmaltado y 2 boxes en la zona de picking del horno cerámico. Teniendo en cuenta además que los boxes disponen de 45 planos, en los cuales solo se puede albergar un azulejo por plano, en el almacén intermedio se tiene una capacidad para albergar 5220 piezas.

Por lo que, según las premisas anteriores, en un periodo de 5 días (7200 minutos) solo se puede acumular un total de 5220 piezas (ver Tabla 8); ello supone que la diferencia entre el TH de llegada y el TH de salida del almacén intermedio debe ser 0,725 piezas por minuto en esa ventana de entre semana.

$$IA_{Bizcocho} = (TH_{P-S-E} - TH_{Horno}) \cdot VT_{Entre\ semana}$$

$$5220 = (TH_{P-S-E} - TH_{Horno}) \cdot 7200$$

$$(TH_{P-S-E} - TH_{Horno}) = 0,725 \text{ piezas/minuto}$$

Tabla 8. Diferencia entre la tasa de llegada y la tasa de salida del almacén intermedio de bizcocho.

3.5.1.2. Medidas de flujo deterministas del horno.

Siguiendo con esta estrategia de planificación, durante el periodo de fin de semana el horno debe ser capaz de consumir todo el IA. Por lo que el horno deberá cocer 5220 piezas en un periodo de 48h (2880 minutos) (ver Tabla 9).

$$IA_{Bizcocho} = TH_{Horno} \cdot VT_{Entre\ semana}$$

$$5220 = TH_{Horno} \cdot 2880$$

$$TH_{Horno} = 1,8125 \text{ piezas/minuto}$$

Tabla 9. Tasa de producción teórica del horno.

El horno cerámico tiene un ciclo de cocción de 75 minutos por lo que mediante este TH el horno acumulará en su interior un WIP de 135,937 piezas; teniendo en cuenta además que el horno mide 200 m, el horno podrá asumir este WIP con una separación entre piezas de 27.12 cm (ver Tabla 10).

$$\begin{aligned}
 WIP_{Horno} &= TH_{Horno} \cdot CT_{Horno} \\
 WIP_{Horno} &= 1,8125 \cdot 75 \\
 WIP_{Horno} &= 135,937 \text{ piezas} \\
 D_{Separación} &= \frac{L_{Horno} - (WIP_{Horno} \cdot L_{Pieza})}{WIP_{Horno}} \\
 D_{Separación} &= \frac{200 - (135,937 \cdot 1,2)}{135,937} \\
 D_{Separación} &= 0,2712 \text{ m}
 \end{aligned}$$

Tabla 10. WIP teórico del horno.

3.5.1.3. Medidas de flujo deterministas de la línea de Prensado-secado-esmaltado.

En el punto 3.6.1.1 de este capítulo se ha determinado teóricamente cual es la diferencia entre el TH de llegada y el TH de salida del almacén intermedio (ver Tabla 8) y posteriormente, en el punto 3.6.1.2, se ha calculado cual es el TH teórico del horno; lo que permite determinar cuál debe ser el TH teórico de la línea de Prensado-secado-esmaltado (ver Tabla 10).

$$\begin{aligned}
 TH_{P-S-E} - TH_{Horno} &= 0,725 \text{ piezas/minuto} \\
 TH_{P-S-E} &= 2,5375 \text{ piezas/minuto}
 \end{aligned}$$

Tabla 10. TH teórico de la línea de Prensado - secado - esmaltado..

Una vez se ha conseguido calcular esta medida de flujo, es necesario ajustar la tasa de producción del cuello de botella de los elementos que componen la línea de Prensado-secado-esmaltado para adecuar su ritmo de producción a este TH.

La sección que hay inmediatamente a la entrada del parque de boxes es la línea de esmaltado; esta se puede concebir como una cinta de transporte de una longitud de 120 m que se mueve a una velocidad de 50 m/min, donde debido a su nivel de merma, el 5% de las piezas se descartan. Ello significa que deberá mantener un TH de 2,671 piezas/minuto (El TH de la línea de Prensado-secado-esmaltado mayorado según el nivel de merma) con lo que acumulará en su interior un WIP de 6,41 piezas (ver Tabla 11).

$$\begin{aligned}
 WIP_{L.Esmaltado} &= TH_{L.Esmaltado} \cdot CT_{L.Esmaltado} \\
 WIP_{L.Esmaltado} &= \frac{TH_{P-S-E}}{r_{merma}} \cdot \frac{L_{L.Esmaltado}}{v_{L.Esmaltado}} \\
 WIP_{L.Esmaltado} &= \frac{2,5375}{0,95} \cdot \frac{120}{50} \\
 WIP_{L.Esmaltado} &= \frac{2,5375}{0,95} \cdot \frac{120}{50} \\
 WIP_{L.Esmaltado} &= 6,41 \text{ piezas}
 \end{aligned}$$

Tabla 11. WIP teórico de la línea de esmaltado

A simple vista, se puede observar que en esta sección el WIP de la línea de esmaltado queda holgadamente distribuido (de 6 a 7 piezas de 120x120 cm repartidas en 120 m de cinta); por lo que no se genera cuello de botella y su TH es el mismo que el de la sección anterior, el secadero vertical.

Por su parte, el secadero vertical se puede concebir como una cinta de transporte formada por 140 cestas, donde en cada cesta cabe una pieza cerámica. Según lo expuesto en el párrafo anterior, el TH del secadero vertical debe ser de 2,671 piezas/minuto lo que provocará que su WIP alcance un valor de 106,84 piezas (ver Tabla 12). Teniendo en cuenta que el WIP del secadero vertical está por debajo de su capacidad máxima se observa que este elemento no genera cuello de botella, y que su TH va a ser igual que el TH de la sección anterior a este, la prensa hidráulica.

$$WIP_{S,vertical} = TH_{S,vertical} \cdot CT_{Secado}$$

$$WIP_{S,vertical} = 2,671 \cdot 40$$

$$WIP_{S,vertical} = 106,84 \text{ piezas}$$

Tabla 12. WIP teórico del secadero vertical.

Por lo tanto, la prensa hidráulica es el proceso que define la tasa de producción de la línea Prensado-secado-esmaltado. Esta prensa cuenta con planchas con capacidad de una pieza por golpe de prensa; por lo que, para adecuar su ritmo de producción a la planificación semanal de producción, deberá reducir su tasa de producción de 3,5 a 2,671 golpes/minuto.

3.5.1.4. Medidas de flujo deterministas de la línea de clasificación.

Esta etapa de producción se encuentra a la salida del horno y tiene un TH muy superior que el del horno. Por ese motivo sus medidas de flujo no son significativas con respecto al proyecto porque no tienen ningún efecto sobre el IA del almacén intermedio. Por ese motivo, se parte de la premisa de que esta etapa de producción tiene un CT=0 y solo se llega a definir a nivel descriptivo. Por otro lado, es necesario considerar esta etapa ya que completa, a nivel conceptual, la definición de la línea de producción de Azulejos S.A.

3.5.1.5. Medidas de flujo deterministas de la línea de embalaje.

Al igual que en el punto anterior, esta etapa de producción se encuentra más allá de la salida del horno y tiene un TH muy superior que el del horno. Por ese motivo sus medidas de flujo de producción no son significativas con respecto al proyecto porque no tienen ningún efecto sobre el IA del almacén intermedio. El único parámetro que se indica es la cantidad de unidades que se empaquetan por caja (ver punto 3.4 Características de los recursos de la línea de producción); y sabiendo que la TH del horno (Tasa de llegada de piezas desde el horno es 1,8125 piezas/minuto), el CT por caja será de 13,79 min (ver Tabla 13).

$$WIP_{Caja} = TH_{Horno} \cdot CT_{Embalaje}$$

$$25 = 1,8125 \cdot CT_{Embalaje}$$

$$CT_{Embalaje} = 13,79 \text{ minutos}$$

Tabla 13. CT teórico por caja de azulejos.

3.5.1.6. Medidas de flujo teóricas del almacén intermedio II.

La etapa de almacenamiento intermedio cuenta con cuatro operaciones de producción (ver Tabla 7):

- **Operación 4:** Llenado de boxes con bizcocho.
- **Operación 5:** Transporte de boxes desde el área de picking de la línea de esmaltado hasta el parque de boxes, donde se engloba la tarea de reponer boxes vacíos en el área de picking.
- **Operación 6:** Transporte de boxes desde el parque hasta el área de picking del horno cerámico, donde se engloba la tarea de almacenar los boxes ya vacíos en el parque de boxes.
- **Operación 7:** Descarga del bizcocho de los boxes en el horno.

En la operación de llenado de boxes, el CT viene determinado por el TH de la línea de Prensado-secado-esmaltado. Lo que significa que cada 17,73 min se generará un box; o lo que es lo mismo, que el TH de la línea de Prensado-secado-esmaltado será de 3,38 boxes/hora (ver Tabla 14).

$$WIP_{Box} = TH_{P-S-L} \cdot CT_{Box}$$

$$45 = 2,5375 \cdot CT_{Box}$$

$$CT_{Box} = 17,73 \text{ minutos}$$

$$TH_{P-L-S}^{Box} = \frac{1 \cdot box}{CT_{Box}}$$

$$TH_{P-L-S}^{Box} = 3,38 \text{ box/hora}$$

Tabla 14. CT teórico por box llenado.

En la operación de descarga del bizcocho ocurre algo similar, el CT viene determinado por el TH del horno cerámico. Lo que significa que cada 24,82 minutos se vaciará un box; o lo que es lo mismo, que el TH del horno cerámico será de 2,41 boxes/hora (ver Tabla 15).

$$WIP_{Box} = TH_{Horno} \cdot CT_{Box}$$

$$45 = 1,8125 \cdot CT_{Box}$$

$$CT_{Box} = 24,81 \text{ minutos}$$

$$TH_{P-L-S}^{Box} = \frac{1 \cdot box}{CT_{Box}}$$

$$TH_{P-L-S}^{Box} = 2,41 \text{ boxes/hora}$$

Tabla 15. CT teórico por box vaciado.

3.5.1.7. Resumen de resultados.

En la Tabla 16 se ofrece un resumen de los resultados del cálculo de las medidas de flujo que caracterizan el comportamiento a nivel productivo del proceso en régimen permanente según la estrategia de planificación semanal planteada. Puesto que el desarrollo del modelo de simulación se va a realizar desde un enfoque determinista, estas medidas de flujo básicas pueden servir para verificar el correcto funcionamiento de las simulaciones comparando los datos que arroja el modelo con estos previamente calculados.

Id	Operación	CT	WIP	TH
1	Prensado	9 días	34722 piezas	2,671 golpes/min
2	Secado	40 min	106,84 piezas	2,671 piezas/min
3	Esmaltado	2,4 min	6,41 piezas	2,671 piezas/min
4	Llenado de boxes	17,73 min	45 piezas	2,537 piezas/min
5	Transporte del box al parque	-	-	3,38 boxes/h
6	Transporte del box horno	-	-	2,41 boxes/h
7	Vaciado de boxes	24,82 min	45 piezas	1,8125 piezas/minuto
8	Cocción	75 min	135,937 piezas	1,8125 piezas/minuto
9	Inspección y clasificado	-	-	1,8125 piezas/minuto
10	Embalado	13,79 min	25 piezas	1,8125 piezas/minuto

Tabla 16. Resultados del cálculo de las medidas de flujo básicas de las operaciones de producción.

4. NORMAS Y REFERENCIAS

4.1. APLICACIONES INFORMÁTICAS

4.1.1. ANYLOGIC

La principal herramienta para la concepción del modelo de simulación de la línea de fabricación cerámica y para la ejecución de la simulación correspondiente es Anylogic. AnyLogic es una desarrollada por The AnyLogic Company que incluye todos los métodos de simulación más comunes en práctica de hoy. Los modelos en AnyLogic pueden basarse en cualquiera de los principales paradigmas de modelado de sistemas, ilustrados en la Figura 18:

- **Eventos discretos (DE).** Para nivel medio y nivel bajo de abstracción.
- **Dinámica de sistemas (SD).** Para modelado estratégico, específico en niveles altos de abstracción.
- **Basada en agentes (AB).** Para prácticamente cualquier tipo de modelo.

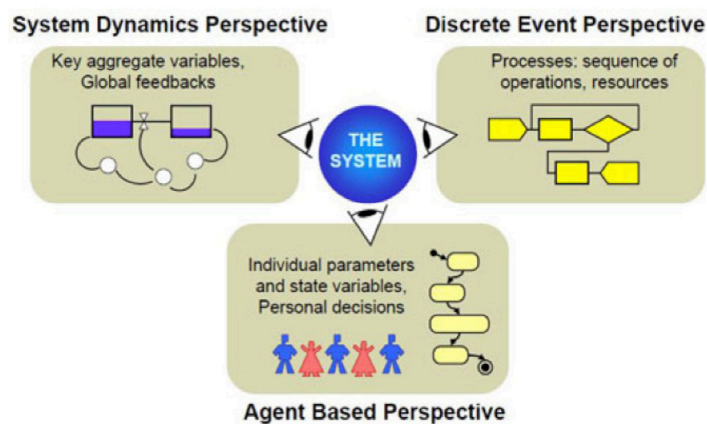


Figura 18. Tres enfoques de modelado.

La dinámica de sistemas y eventos discretos son enfoques tradicionales de simulación, pero el método

basado en el *agente* es relativamente nuevo. En el paradigma de la dinámica de sistemas, se pueden modelar procesos continuos; mientras que en el de elementos discretos se ejecutan elementos de manera continuada. La Dinámica de sistemas, utiliza el más alto nivel de abstracción mientras que los sistemas de eventos discretos utilizan un nivel de abstracción mediano. La simulación basada en *agentes* funciona a través de todos los niveles de la abstracción. Los *agentes* puede ser objetos de muy diversa naturaleza y escala: en el nivel “físico” pueden, por ejemplo, ser peatones o vehículos o robots, en el nivel mediano pueden simular clientes, al más alto nivel pueden ser empresas en funcionamiento.

AnyLogic permite que el modelador combine estos métodos de simulación dentro de un mismo modelo. Por ejemplo, se podría crear un modelo de conjunto de la industria del transporte marítimo basado en transportes que actúen como *agentes* reaccionando independientemente. Al mismo tiempo el funcionamiento interno de sus redes de transporte y la infraestructura podrían ser representados por un modelo de simulación de eventos discretos. Del mismo modo, un modelo puede representar tanto consumidores como *agentes*, cuyo comportamiento global alimenta a un modelo de dinámica de sistemas que describa flujos de variables como ingresos o gastos que no necesitan ser vinculados a los *agentes* individuales.

AnyLogic incluye un lenguaje de modelado gráfico y también permite que los usuarios puedan ampliar los modelos de simulación con código de Java. Java es un lenguaje de programación de propósito general, concurrente, orientado a objetos. Su sintaxis deriva en gran medida de C y C++, pero tiene menos utilidades de bajo nivel que cualquiera de ellos. Las aplicaciones de Java son compiladas a bytecode (clase Java), que puede ejecutarse en cualquier máquina virtual Java (JVM) sin importar la arquitectura de la computadora subyacente.

4.1.1.1. Experimento de optimización en Anylogic.

Si se necesita ejecutar una simulación y observar el comportamiento del sistema bajo ciertas condiciones, así como mejorar el rendimiento de este, (por ejemplo, tomando decisiones sobre los parámetros y/o la estructura del sistema), se puede utilizar la capacidad de optimización de AnyLogic. La optimización consiste en encontrar la combinación óptima de condiciones que resulta en la mejor solución posible. Por ejemplo, la optimización puede ayudarle a encontrar, por ejemplo, el rendimiento óptimo de un servidor o el mejor método para procesar facturas.

La optimización de AnyLogic se basa en el motor de optimización de OptQuest, una de las herramientas de optimización más flexibles y fáciles de usar del mercado. El motor de OptQuest encuentra automáticamente los mejores parámetros de un modelo, con respecto a ciertas restricciones. AnyLogic proporciona una cómoda interfaz gráfica de usuario para configurar y controlar la optimización.

OptQuest es una marca registrada de OptTek Systems, Inc. Para obtener información avanzada sobre el motor de OptQuest, visite el sitio web de OptTek [11]

El proceso de optimización consiste en simulaciones repetitivas de un modelo con diferentes parámetros. Utilizando sofisticados algoritmos, el motor de OptQuest varía los parámetros controlables desde la simulación hasta encontrar los parámetros óptimos para resolver un problema.

Para información específica acerca de la naturaleza de los experimentos de optimización en Anylogic, el lector es remitido al soporte técnico de Anylogic disponible en el propio software o en línea a través de su página web [12].

4.1.2. HOJAS DE CÁLCULO: MS EXCEL

Diversas hojas de cálculo y tablas Excel son utilizadas a lo largo del proceso de elaboración del proyecto con el fin confeccionar tablas de presentación de resultados.

4.1.3. PROCESADOR DE TEXTOS: MS WORD

Un procesador de textos es escogido para la redacción de la presente memoria, así como la confección de todos aquellos informes técnicos derivados de la misma. Si bien la mayoría de procesadores de texto contienen, en la base, funcionalidades similares, se he seleccionado el software MS Word como herramienta principal para tales tareas.

4.1.4. HARDWARE

El soporte hardware utilizado para el empleo de las herramientas soft descritas precedentemente consiste en un ordenador de la marca HP con las especificaciones listadas a continuación:

- Disco duro de 256 Gb, RAM 4 Gb
- Intel Core i5 i5-6200U 2.4 GHz
- Tarjeta gráfica Intel 3 Gb
- Sistema operativo Windows 10

4.2. PLAN DE GESTIÓN DE CALIDAD

Este proyecto es redactado según los criterios generales recogidos en la norma UNE 157001 relativos a la elaboración formal de documentos constituyentes de un proyecto técnico. Dicha normativa dicta los requisitos formales de redacción de los proyectos de productos y obras, entre otros elementos, permitiendo así alcanzar un grado de calidad aceptable del documento técnico presentado.

Por lo que respecta a la calidad del contenido del informe técnico final, diferentes reuniones con carácter consultativo y de revisión son efectuados a medida que se progresa en el proyecto y se alcanzan metas intermedias. Con ello se busca asegurar que el trabajo no acarrea errores de etapas precedentes y que la calidad del mismo no decae. Sendas memorias serán redactadas tras la finalización de las reuniones con el fin de tener un registro escrito de los puntos más notables comentados.

4.3. REFERENCIAS

[1] Mejores Técnicas Disponibles de referencia europea - Sector de la fabricación cerámica. (2011). [pdf] Madrid: Ministerio para la Transición Ecológica. Disponible en: http://www.prtr-es.es/Data/images/MTD_Ceramica_ES.pdf [Acceso el 7 de enero de 2019].

[2] TWG Ceramics (2005). Merged and sorted comments master spread sheet on draft 1.

[3] Institut de Promoció Ceràmica (2019). IPC – Clasificaciones comercial y técnica. [online] Disponible en: http://www.ipc.org.es/guia_colocacion/info_tec_colocacion/los_materiales/baldosas/clas_com_tec.html [Acceso el 7 de enero de 2019].

[4] UBA (2001). Exemplary Investigation into the State of Practical Realization of Integrated Environmental Protection within the Ceramics Industry under Observance of the IPPC-Directive and the Development of BAT

Reference Documents.

[5] García, A. (2019). Procesos de fabricación de baldosas cerámicas - Conocimientos básicos. [online] Slideshare. Disponible en: <https://es.slideshare.net/AgustinGarcia7/procesos-fabricacion-baldosas-ceramicas> [Acceso el 7 de enero de 2019].

[6] Wikipedia (2019). AnyLogic. [online] Disponible en: <https://es.wikipedia.org/wiki/AnyLogic> [Acceso el 7 de enero de 2019].

[7] Wikipedia (2019). Java (lenguaje de programación). [online] Disponible en: [https://es.wikipedia.org/wiki/Java_\(lenguaje_de_programaci3n\)](https://es.wikipedia.org/wiki/Java_(lenguaje_de_programaci3n)) [Acceso el 7 de enero de 2019].

[8] Asociación Española de Normalización y Certificación (AENOR) (febrero de 2002). UNE 1570001:2002 – Criterios generales para la elaboración formal de documentos que constituyen un proyecto.

[9] Asociación Española de Normalización y Certificación (AENOR) (junio de 2013). UNE-EN 14411:2013 – Baldosas cerámicas. Definiciones, clasificación, características, evaluación de la conformidad y marcado.

[10] Asociación Española de Normalización y Certificación (AENOR) (mayo de 2013). UNE 50132:2013 – Sistemas de alarma. Sistemas de vigilancia CCTV para uso en aplicaciones de seguridad.

[11] www.opttek.com

[12] <https://www.anylogic.com>

5. REQUISITOS DEL MODELO DE SIMULACIÓN

El modelo de simulación debe de disponer de un método de entrada de parámetros, de fácil manipulación, para que se pueda adaptar el modelo de simulación a los recursos de la distribución en planta de Azulejos S.A. Por otro lado, debe permitir modificar la dinámica de las etapas de producción definidas en el modelo. Además, debe permitir simular el comportamiento de plantas azulejeras con una distribución en planta similar.

Como requisito de diseño, se pide también modelar un lay-out del almacén intermedio en el propio software de simulación en base a las características básicas que se especifican en el punto 3.4 Caracterización de la distribución en planta de Azulejos S.A. De manera que, posteriormente sea posible ajustar, de forma iterativa, el comportamiento de los recursos de este almacén, con el objetivo de que tengan la capacidad suficiente para dar servicio a la línea de producción cerámica. Este modelado a su vez debe contar a su vez con una interfaz gráfica donde se represente de manera visual el movimiento de los elementos que componen el almacén intermedio, así como su interacción dinámica

Por último, el modelo de simulación debe ser robusto, por lo que, aunque la parametrización del modelo de simulación fuerce al modelo a bloquear el flujo de elementos (por ejemplo, forzar una etapa de producción con una tasa de llegada mayor que una tasa de salida), la simulación debe seguir ejecutándose sin ningún problema.

5.1. VERIFICACIÓN DEL MODELO DE SIMULACIÓN.

Se realizará una simulación del modelado de procesos según la parametrización especificada a partir de las características de los recursos de la línea de producción para comprobar que el modelo de simulación es correcto:

- Se comprobará que el nivel los WIP de todas las etapas del proceso es equivalente al nivel determinista calculado.
- Se comprobará que cuando es viernes a última hora, el parque de boxes donde se almacena el bizcocho está prácticamente lleno;
- Se comprobará que cuando llega el lunes a primera hora, el parque de boxes donde se almacena el bizcocho está completamente vacío.

5.2. EXPERIMENTO DE OPTIMIZACIÓN.

Como requisito de diseño se pide, mediante la herramienta de optimización de Anylogic, plantear un experimento de optimización para maximizar la utilización de los elementos que modelan el comportamiento de los LGV dentro del modelo de simulación. Esta utilización se definirá como la ratio entre el tiempo en que el LGV está parado y el tiempo en el que el LGV se está utilizando mientras hay una demanda de uso.

5.2.1. CONDICIONES INICIALES DEL EXPERIMENTO DE OPTIMIZACIÓN.

Para realizar el experimento de optimización, se plantearán varios casos prácticos. Se modificarán los parámetros de los elementos que modelan el comportamiento de los LGV dentro del modelo de simulación. De esta manera, se podrá observar su efecto en la optimización.

- **Caso 1: Velocidad de los LGV 5 m/s.**

En este caso, se analizará cual es la cantidad de elementos que maximizan la utilización del parque de LGV cuando estos trabajan a la velocidad que tienen por defecto.

- **Caso 2: Velocidad de los LGV 7,5 m/s**

En este caso, se analizará cual es la cantidad de elementos que maximizan la utilización del parque de LGV cuando estos trabajan a una velocidad superior a la que tienen por defecto.

- **Caso 3: Velocidad de los LGV 5 m/s considerando paradas por recarga de baterías.**

En este caso, se analizará cual es la cantidad de elementos que maximizan la utilización del parque de LGV. Por otro lado, también se tendrán en cuenta los efectos que tienen las paradas de mantenimiento en el número de elementos que maximiza la utilización de este parque.

Los LGV son vehículos eléctricos que funcionan con baterías de plomo-ácido de alto contenido en antimonio, ello significa que periódicamente tienen que parar a recargar dichas baterías. Partiendo del supuesto de que un LGV tiene unas 12 horas de autonomía funcionando a una velocidad de 5 m/s y necesita unas 2 horas de recarga, para este caso la utilización máxima será del 83,33%.

6. ALTERNATIVAS

La herramienta para la concepción del modelo de simulación de la línea de fabricación cerámica, tal y como se ha comentado en el Capítulo 4, es AnyLogic. Este software incluye todos los métodos de simulación más comunes en práctica de hoy, los cuales pueden basarse en cualquiera de los principales paradigmas de modelado de sistemas. A continuación, se presentan brevemente las diferentes alternativas basadas en esta herramienta de simulación.

6.1. ALTERNATIVA ADOPTADA

Para desarrollo del modelo de simulación se han utilizado, fundamentalmente, los objetos de la librería del modelado de procesos del propio software de Anylogic. Mediante los bloques de esta librería se han modelado las distintas etapas del proceso de producción cerámico a partir un diagrama de flujo. En este caso, algunas variables de los bloques del modelado de procesos, presentes en el diagrama de flujo, están controladas por secuencias de código que se ejecutan periódicamente.

Por otro lado, en algunos casos se ha utilizado el método basado en *agentes* para construir algunos bloques específicos. Se han desarrollado varios tipos propios de *agente* para poder modelar el movimiento de los Boxes y LGV a lo largo del almacén intermedio de bizcocho. Para ello, se ha constituido, dentro del modelo de simulación, un espacio 2D a partir de los objetos de la librería Space Markup donde se representa el Lay-Out del almacén intermedio de bizcocho.

Estos *agentes* se han desarrollado a nivel de objeto, son entes con características y atributos que se desplazan por el modelo de simulación interactuando con la dinámica del sistema, pero que no toman decisiones. Por otro lado, la librería de modelado de procesos es un marco abierto que permite combinar el modelado de procesos con modelos basados en *agentes*. En este caso, se han utilizado objetos para poder insertar estos *agentes* en el flujo de procesos con la finalidad de poder integrar ambos paradigmas.

Por último, para poder desarrollar la herramienta de optimización para maximizar la utilización de los elementos que modelan el comportamiento de los LGV dentro del modelo de simulación, se ha optado por utilizar el experimento de optimización que viene incluido dentro del marco de experimentos que ofrece Anylogic.

6.2. SOLUCIÓN ALTERNATIVA: DESARROLLO DE UN ALGORITMO DE OPTIMIZACIÓN PARA EL FUNCIONAMIENTO DE LOS LGV.

Como alternativa, se podría haber desarrollado un *agente* LGV con un algoritmo interno que le permitiera optimizar las tareas de transporte y reposición de boxes dentro del almacén intermedio creado en el espacio continuo del modelo de simulación. Por otro lado, el experimento de optimización se centra en maximizar la utilización del ResourcePool donde están asociados estos *agentes* (ver punto 7.2.2 Definición de la función objetivo). Partiendo de esta premisa, el desarrollar este algoritmo interno dentro de estos *agentes* es completamente compatible con el modelo de simulación actual. Por lo que, este proyecto puede ser objeto de mejora a partir de esta alternativa en futuros proyectos.

6.3. SOLUCIÓN ALTERNATIVA: DESARROLLO EN DETALLE DEL MOVIMIENTO DE LAS PIEZAS CERÁMICAS A LO LARGO DE LA LÍNEA DE PENSADO-SECADO-ESMALTADO.

Dentro del espacio continuo del modelo de simulación se ha modelado únicamente el almacén intermedio en detalle. Para simplificar este entorno, se ha modelado tanto la línea de Pensado-secado-esmaltado, así como la línea de cocción como un único punto dentro del modelado 2D, y a su vez no se ha modelado el movimiento de las baldosas cerámicas a través de este. Como alternativa, se podría haber modelado en detalle estas dos líneas mencionadas anteriormente. Pero puesto que se está modelando un proceso continuo basado en la dinámica de eventos discretos, donde no se van a considerar efectos como cambios de formato, ajustes en la línea de esmaltado, o paradas de mantenimiento; el valor que aportaría esta alternativa al modelo de simulación es irrelevante.

Sin embargo, este modelo de simulación puede ser un punto de partida para crear una versión posterior

para abordar otros detalles. Por ejemplo, se podría realizar una implementación gráfica de la línea de Prensado-secado-esmaltado para poder implementar estrategias de cara a optimizar las labores de mantenimiento y ajuste de la línea.

7. SOLUCIÓN ADOPTADA

En este capítulo se describirá la solución que se ha adoptado para el desarrollo del modelo de simulación, y su posterior experimento de optimización, acorde a los requisitos de diseño expuestos en el Capítulo 5 de esta memoria a partir del proceso de producción cerámico de la empresa Azulejos S.A.. Para información específica acerca de la naturaleza de los tipos de bloques de modelado de procesos, el lector es remitido al Anexo de este documento o al soporte técnico de Anylogic disponible en el propio software o en línea.

Antes de abordar este capítulo, es necesario comprender la naturaleza y las características de los diferentes tipos de bloques de modelado de procesos que se han empleado para desarrollar este modelo de simulación. Como introducción y para obtener información específica acerca de dichos bloques de modelado de procesos, el lector es remitido al Anexo de este documento, o al soporte técnico de Anylogic disponible en el propio software o en línea.

7.1. IMPLEMENTACIÓN EN ANYLOGIC

La Figura 19 muestra el modelo de la línea de fabricación de azulejos descrito en el Capítulo 3 Antecedentes, modelado en el software de simulación Anylogic. El conjunto de bloques de modelado de procesos se ha organizado según las etapas de producción ilustradas en el diagrama de flujo de la Figura 17, las cuales han sido delimitadas unas de otras para mejorar la lectura y comprensión del modelo. Estos componentes se distribuyen de forma lineal. El proceso de producción es inicializado en el bloque de modelado de procesos "source" integrado en la línea de prensado/secado/esmaltado, y finaliza en el bloque "sink2", en la línea de clasificación/embalaje. Existe una línea complementaria adicional que se ramifica desde la línea de Prensado-secado-esmaltado hasta la línea de cocción, comienza desde el bloque de modelado de procesos "enter" y finaliza en el bloque "exit"; y sirve para definir el enrutamiento de los boxes a lo largo del flujo de procesos mediante el recurso LGV. Un recuadro en la sección derecha de la modelo lista los distintos parámetros de funcionamiento y recursos operacionales ligados a los componentes modelados en el diagrama.

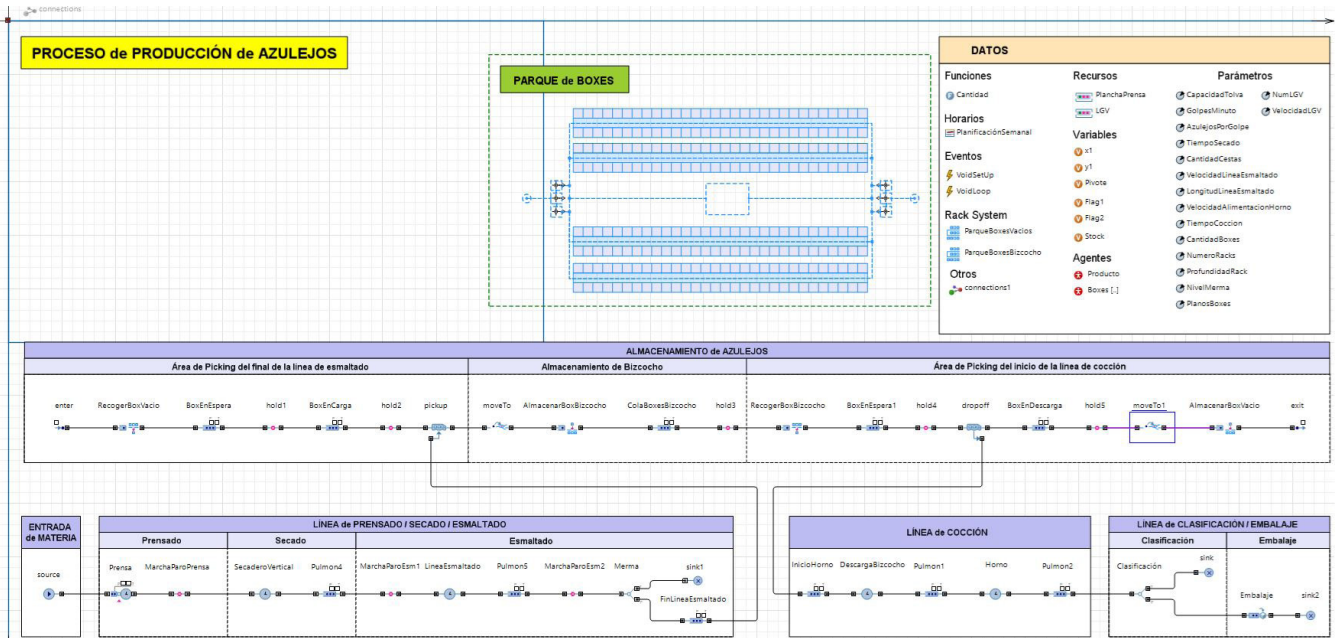


Figura 19. Modelo AnyLogic de la línea de producción cerámica.

A continuación, se describirán las diferentes etapas del modelo definidas en el modelo de simulación, haciendo hincapié en su funcionalidad y parámetros de diseño.

7.1.1. ENTRADA DE MATERIA PRIMA

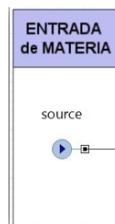


Figura 20. Etapa de entrada de material dentro del flujo de procesos.

La línea de fabricación parte del componente de tipo Source con nombre "source" (ver Figura 20). La finalidad de este elemento es la introducción en la línea de producción de una población de *agentes* de la clase "Producto". El *agente* de la clase "Producto", es un tipo de *agente* que se ha creado para ser inyectado a través del modelo, y que representa las piezas cerámicas que van a circular a lo largo de toda la línea de producción. Además, mediante este *agente* se define como se caracteriza la producción en función de sus parámetros; en la Figura 21 se muestra como se ha modelado el *agente* "Producto" a partir de estos parámetros.

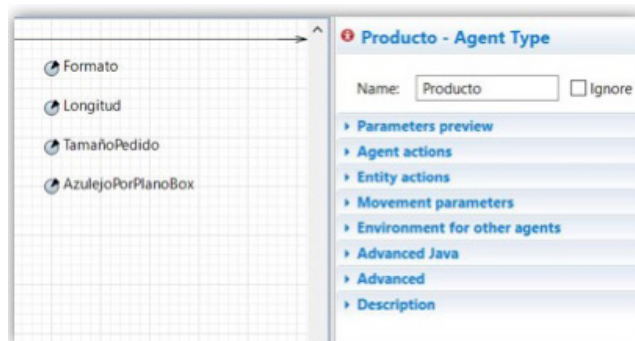


Figura 21. Agente "Producto" definido a partir de parámetros.

Este *agente* "Producto" define un pedido de azulejos (ver punto 3.3 Situación Actual de Azulejos SA): El pedido base definido en la simulación (ver Figura 22) es de 50.000 m² de azulejos planos de un formato de 120 x 120 cm (formato 1.44 m²).

Producto - Producto

Name:

Show name Ignore

Single agent Population of agents

Formato: =

TamañoPedido: =

Longitud: =

AzulejoPorPlanoBox: =

▶ Initial location

▶ Statistics

▶ Advanced

▶ Description

Figura 22. Configuración de los parámetros del agente "Producto".

Por otro lado, se dispone del event llamado "VoidSetUp" del tipo **Condition triggered event** (ver en Anexo II punto 2.4 Events) que se ejecuta a partir de una condición inicial. Este event sólo se ejecuta una vez al iniciar la simulación (dado que su condición es **true**) y sirve para inicializar la simulación antes de que llegue al régimen permanente (ver Figura 23).

VoidSetUp - Event

Name:

Show name Ignore

Visible: yes

Trigger type:

Condition:

▶ Action

▶ Description

Figura 23. Event VoidSetUp.

Dentro de las líneas de código del event VoidSetUp, se hace una llamada al bloque de modelado de procesos "source" desde la función `source.inject()` para que inyecte en el flujo de procesos una cantidad de *agentes* equivalente a la cantidad de azulejos dirimido del tamaño de pedido especificado en el *agente* "Producto" (ver Tabla 9), ya que el bloque "source" está configurado de manera que la llegada de *agentes* está definida a partir de la opción "Calls of inject() function" (ver Figura 25) para que este procedimiento sea posible.

```
//Inyección tierra desde el silo a la Tolva de la prensa
source.inject(Cantidad(Producto.TamañoPedido, Producto.Formato));
```

Tabla 9. Llamada a la función `source.inject()` desde el event "VoidSetUp".

Esta función se apoya a su vez en otra función llamada "Cantidad", la cual calcula cual es la cantidad de azulejos a inyectar en el flujo de procesos a partir de los valores de los campos "TamañoPedido" y "Formato" del agente "Producto" (ver Figura 24).

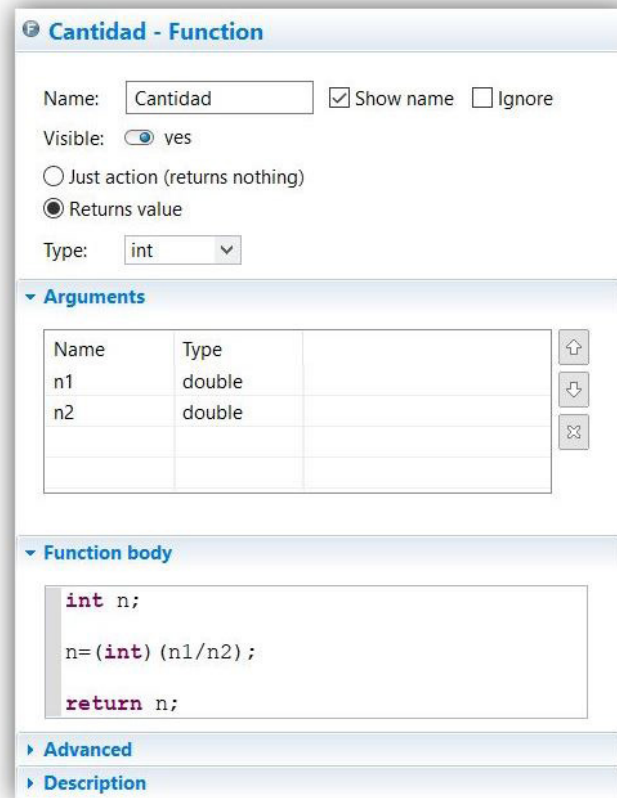


Figura 24. Definición de la función "Cantidad" en el modelo de simulación.

La función "Cantidad" toma los valores de los campos "TamañoPedido" y "Formato", los divide y le devuelve un número entero a la función `source.inject()`.

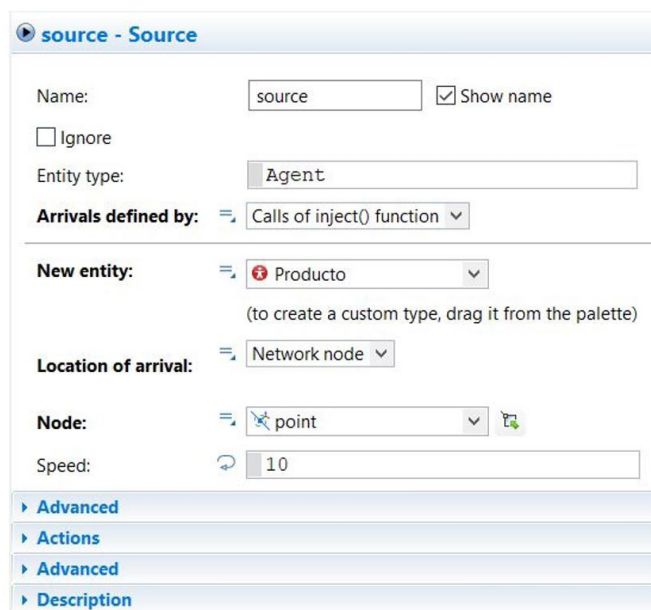


Figura 25. Configuración del bloque de modelado de procesos "source".

La cantidad de piezas introducidas es función del tamaño del pedido (número de azulejos) y del formato de los azulejos (dimensiones), lo que se traduce en una orden de producción de 34721 piezas que instantáneamente avanzan a la siguiente etapa del flujo de procesos.

Cabe añadir que, como se dispone de una Network donde se ha diseñado la distribución en planta del almacén intermedio de bizcocho, se necesita determinar una ubicación física dentro del espacio 2D para todos los bloques de modelado de procesos; y puesto que el comportamiento dinámico de la parte de Prensado-secado-esmaltado no son relevantes, todas ellas se han ubicado dentro de un node de la Network: "point" (ver Figuras 25, 26, 29).

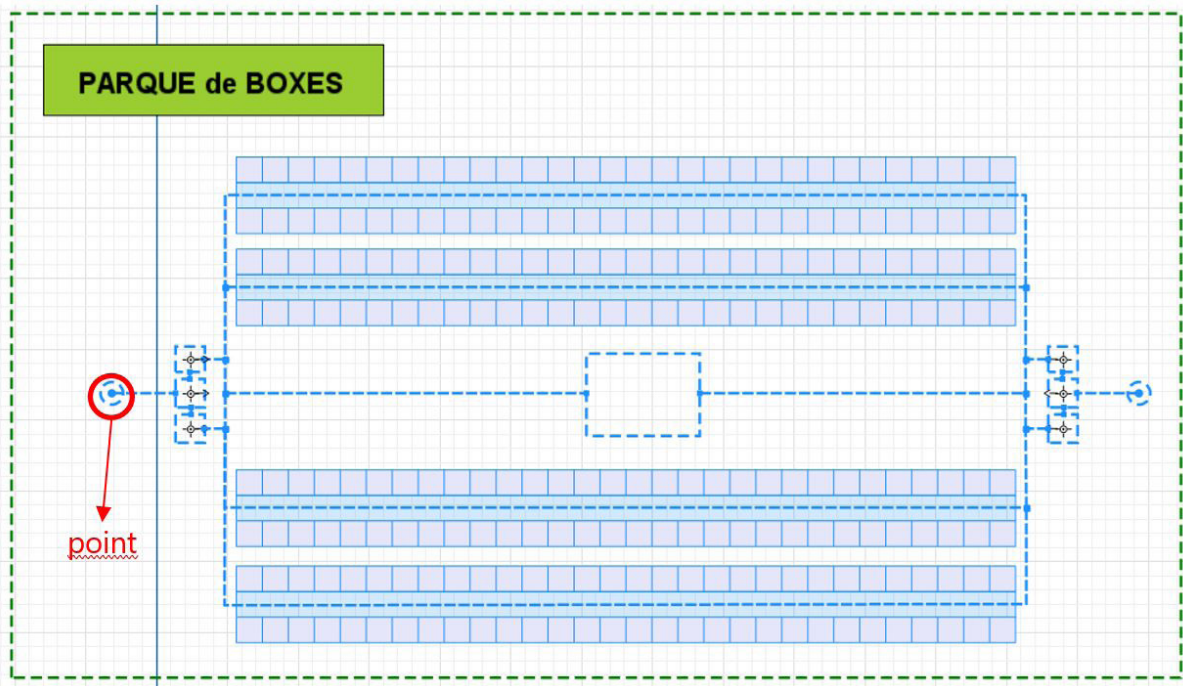


Figura 26. Ubicación de todos los componentes de la línea de Prensado-secado-esmaltado dentro de la Network que modela el almacén intermedio.

7.1.2. LÍNEA DE PRENSADO-SECADO-ESMALTADO

A continuación de este bloque inicial, se ha modelado la línea de Prensado-secado-esmaltado siguiendo la estructura del diagrama de flujo representado en la Figura 18. Esta etapa de fabricación se ha dividido en 3 secciones, (ver Figura 27) en donde se realizan, de forma consecutiva, las operaciones de prensado, secado y esmaltado de piezas.



Figura 27. Línea de prensado-secado-esmaltado.

7.1.2.1. Sección de prensado

La materia prima (cantidad de piezas) introducida por el objeto "source" avanzan hasta la sección de prensado. Ésta esta sección se ha modelado a partir de un bloque de modelado de procesos del tipo Service al que se ha llamado "Prensa" (ver Figura 28), en donde los *agentes* producto quedan almacenados dentro de su Queue interna la espera de ser procesados (**Queue capacity**, ver Figura 29). La capacidad de esta Queue está dimensionada a partir del parámetro "CapacidadTolva" al cual se le ha asignado el mismo valor que al campo "Producto.TamañoPedido".

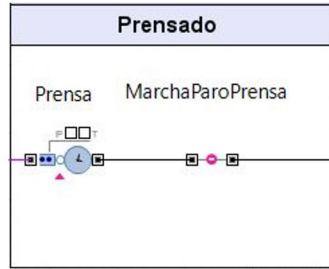


Figura 28. Sección de prensado.

El bloque "Prensa" utiliza un ResourcePool denominado "PlanchaPrensa" para desempeñar sus tareas (**ResourcePool**, ver Figura 29). Este ResourcePool representa a la plancha de la prensa hidráulica que ejecuta la operación de prensado de la etapa.

Se ha definido como un recurso de tipo static (ver Anexo II punto 2.10 ResourcePool) con una capacidad de producción definida a partir del parámetro "AzulejosPorGolpe" (ver Figura 31). Puesto que la capacidad de la plancha sólo permite prensar un azulejo por golpe de plancha (ver punto 3.4 Características de los recursos de la línea de producción) a este parámetro se le ha asignado el valor de 1. Por otra parte, puesto que las prensas hidráulicas trabajan con una única plancha a la vez, se ha configurado el bloque "Prensa" para que solo trabaje con un único recurso "PlanchaPrensa" a la vez (**Number of units**, ver Figura 29).

La actividad de la prensa se modela a través del tiempo de retardo interno del bloque, que representa el tiempo que tarda la prensa en procesar una o varias piezas cerámicas por golpe (**Delay time**, ver Figura 29). Este tiempo de retardo se ha modelado a partir de la inversa del parámetro "GolpesMinuto", el cual representa la tasa de producción de la prensa hidráulica (tiene un valor asignado de 2,671); puesto que de esta manera se obtiene el ciclo de tiempo de la prensa por pieza producida.

Prensa - Service

Name: Show name Ignore

Entity type:

Seize: (alternative) resource sets units of the same pool

Resource pool:

Number of units:

Queue capacity:

Maximum queue capacity:

Delay time:

Send seized resources:

Entity location (queue):

Entity location (delay):

Figura 29. Configuración del bloque de modelado de procesos "Prensa".

- Planificación semanal de la producción.

La planificación semanal de la producción se implementa partir de un bloque Schedule llamado "PlanificaciónSemanal" (ver Figura 30) asociado el recurso "PlanchaPrensa" (ver Figura 31). En este bloque se define un intervalo de tiempo de manera que el bloque "Prensa" solamente tendrá disponible el recurso "PlanchaPrensa" de lunes a viernes.

PlanificaciónSemanal - Schedule

Name: Show name Ignore

Visible: yes

Data

Type: on/off integer real

The schedule defines: Intervals (Start, End) Moments

Duration type: Week Days/Weeks Custom (no calendar mapping)

Default value: on off

Repeat schedule weekly:

Mon	Tue	Wed	Thu	Fri	Sat	Sun	Start	End	Value
✓	✓	✓	✓	✓	-	-	0:00	0:00	on

Action

Figura 30. Configuración de la planificación semanal mediante un bloque *Schedule*.

Ello significa que cuando se inicie la simulación y el contador de tiempo llegue al fin de semana, automáticamente el bloque "Prensa" dejará de funcionar hasta el lunes siguiente.

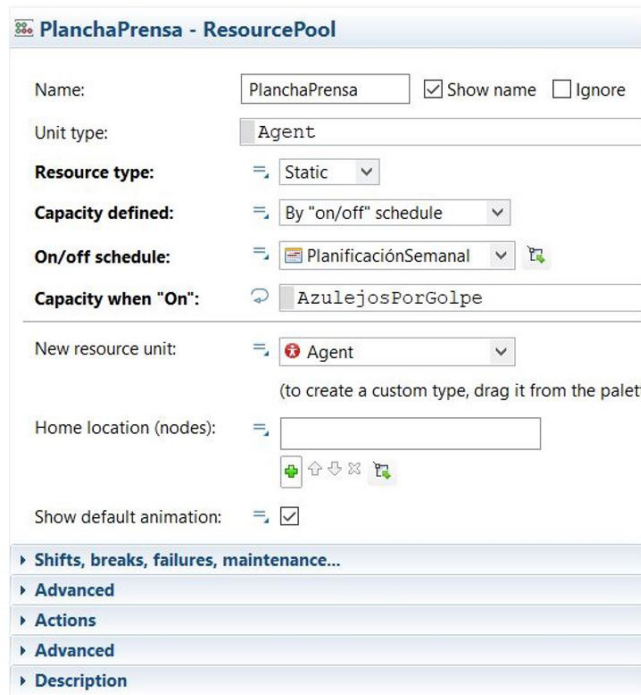


Figura 31. Definición del ResourcePool "PlanchaPrensa".

- **Mecanismo de Marcha-Paro de la prensa.**

A la salida del bloque "Prensa", existe un bloque de modelado de procesos *hold* (ver Anexo II punto 2.17 Hold) con el nombre de "MarchaParoPrensa" (ver Figura 28). La función de este bloque es poder controlar programáticamente la operación de prensado desde el event "VoidLoop".

El event llamado "VoidLoop" es del tipo **Timeout triggered event** (ver en Anexo II punto 2.4 Events) y se ejecuta de manera cíclica al finalizar un periodo de tiempo emulando una rutina de interrupción periódica (ver Figura 32).

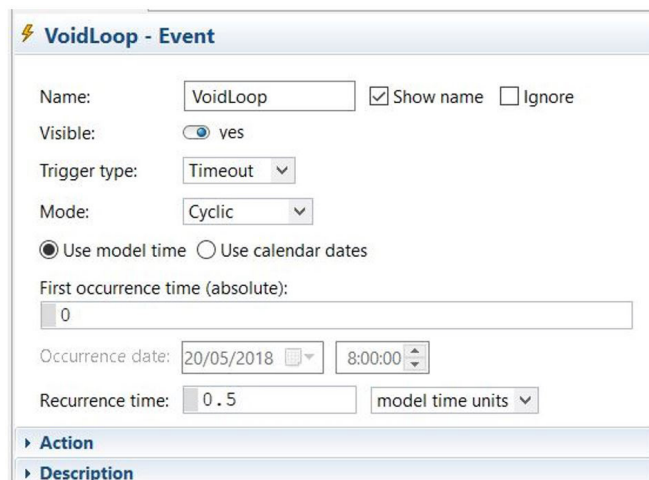


Figura 32. Definición del event "VoidLoop".

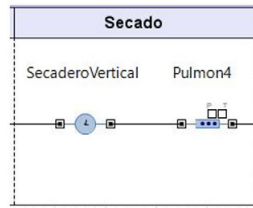


Figura 33. Subetapa de secado.

El tiempo de retardo se ha modelado a partir del parámetro "TiempoSecado"; y la capacidad de su Queue interna está dimensionada a partir del parámetro "CantidadCestas", al cual se le ha asignado la cantidad de cestas de transporte que dispone el secadero vertical utilizado en la planta de Azulejos S.A. (ver punto 3.4 Características de los recursos de la línea de producción).

Figura 34. Definición del delay "SecaderoVertical".

Atendiendo a las características de los recursos de producción, las piezas cerámicas que salen del bloque "SecaderoVertical" entran dentro de la cola de una Queue denominada "Pulmon4", cuya capacidad está definida a partir del parámetro "CantidadCestas" (ver Figura 35). Mientras el flujo de piezas no quede bloqueado, los agentes "Producto" que atravesarán este bloque en un tiempo 0 segundos; mientras que, en el caso contrario, se acumularán hasta que el flujo de piezas se desbloquee.

Figura 35. Definición del bloque "Pulmon4" en Anylogic.

7.1.2.3. Sección de esmaltado

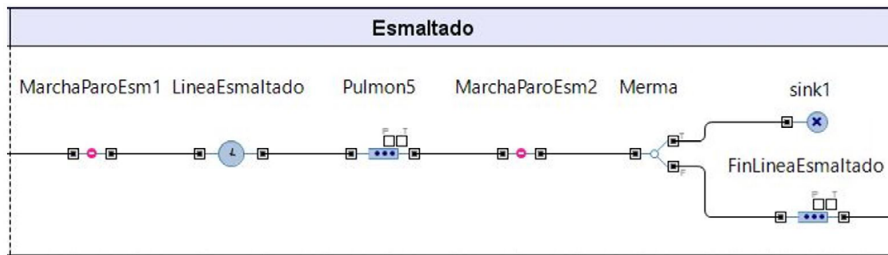


Figura 36. Subetapa de esmaltado.

Al finalizar la operación de secado, los azulejos pasan a la sección de esmaltado. La función principal de sección está definida a partir de un bloque de modelado de procesos del tipo Delay al que se ha llamado “LíneaEsmaltado” (ver Figura 36), en donde el retardo que se introduce, representa el tiempo de ciclo que tardan los azulejos en recorrer la línea de esmaltado (**Delay time**, ver Figura 37). Este periodo de tiempo se ha estimado dividiendo la longitud de la línea de esmaltado entre su velocidad.

Cuando los *agentes* entran en el bloque, quedan almacenados dentro de su Queue interna la espera de ser procesados (**Queue capacity**, ver Figura 36). La capacidad de esta Queue está dimensionada a partir de la función “Cantidad” (ver Figura 24), la cual toma los valores del parámetro “LongitudLineaEsmaltado” y del campo “Producto.Longitud” y le devuelve un número entero que representa la cantidad máxima de azulejos de 120x120 cm lo largo de una cinta de transporte de 120 m. Puesto que los *agentes* entran de manera discreta a este bloque a partir de una tasa de llegada definida en el bloque anterior, cada pieza inicia su temporizador de retardo de manera independiente. De esta manera, se emula el comportamiento de la cinta de transporte que recorre la línea de esmaltado (ver punto 3.1.3.4 Esmaltación y decoración de productos cerámicos); y por otra parte se mantiene una tasa de producción a la salida igual a la tasa de llegada de piezas.

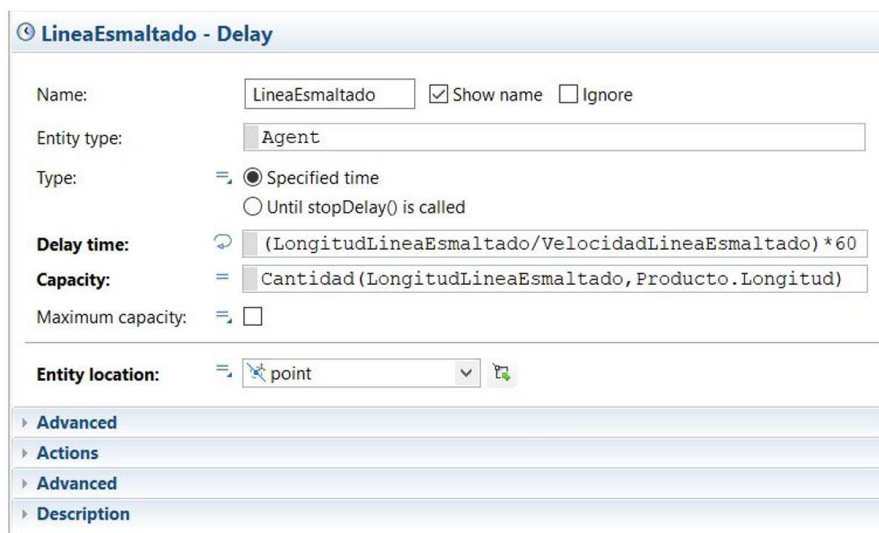


Figura 37. Definición del bloque de modelado de procesos “LineaEsmaltado”.

Atendiendo a las especificaciones de diseño, las piezas cerámicas que salen del bloque “LíneaEsmaltado” entran dentro de la cola de una queue denominada “Pulmon5”, cuya capacidad está definida a partir de la función “Cantidad”, la cual realiza el mismo cálculo que en el bloque “LíneaEsmaltado” (ver Figura 37). Mientras el flujo de piezas no quede bloqueado, los *agentes* “Producto” que atravesarán este bloque en un tiempo 0 segundos; mientras que, en el caso contrario, se acumularán hasta que el flujo de piezas se desbloquee.

Para ello, se pregunta mediante las funciones `BoxesBizcocho.hasSpace()` y `BoxesBizcocho1.hasSpace()` si en el *PalletRack* "BoxesBizcocho" existen celdas vacías. En el caso en el que esta condición se cumpla, se hace una llamada a los bloques "MarchaParoEsm1" y "MarchaParoEsm2" mediante las funciones `MarchaParoEsm1.unlock()` y `MarchaParoEsm2.unlock()` para desbloquear el flujo de *agentes* en la sección de esmaltado. En el caso en que la condición anterior no se cumpla, se hace una llamada a los bloques "MarchaParoEsm1" y "MarchaParoEsm2" mediante las funciones `MarchaParoEsm1.block()` y `MarchaParoEsm2.block()` para bloquear el flujo de *agentes*.

A la salida del bloque "Pulmon5" el flujo de azulejos pasa por un bloque de modelado de procesos del tipo *SelectOutput* (ver Anexo II punto 2.19 *SelectOutput*) denominado "Merma" (ver Figura 36). En este bloque, el 5% (`NivelMerma`) de las piezas pasan a un bloque del tipo *sink* denominado "sink1" para ser eliminados del flujo de procesos (**Probability**, ver Figura 38). Esto se hace para modelar el efecto de la merma a la salida de la línea de esmaltado de los *agentes* que recorren el modelo de simulación.

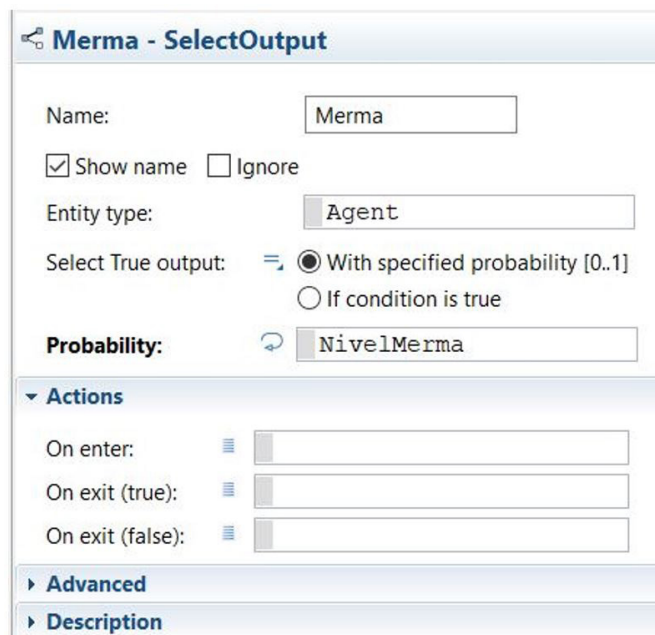


Figura 38. Definición del bloque de modelado de procesos "Merma".

7.1.3. ALMACENAMIENTO INTERMEDIO DE LOS AZULEJOS (I)

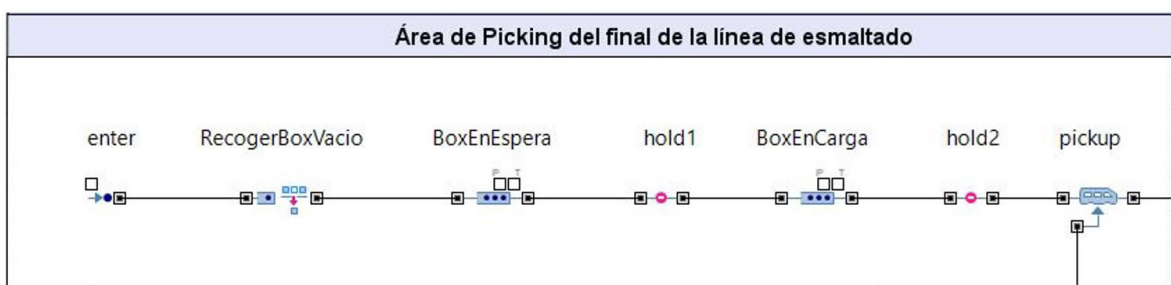


Figura 39. Área de Picking de la línea de esmaltado.

Los *agentes* que continúan por el flujo de procesos, llegan al área de picking de la línea de esmaltado. Esta área de picking se ha modelado mediante un bloque de modelado de procesos del tipo *queue* llamado "FinLineaEsmaltado" (ver figura 36) conectado a un bloque del tipo *Pickup* (ver Anexo II punto 2.16 *Pickup*) (ver figura 39). De manera que, cuando la cantidad de *agentes* en la cola del bloque "FinLineaEsmaltado" es igual al parámetro "PlanosBoxes", estos se cargan en un *agente* del tipo "Box" (ver figura 40).

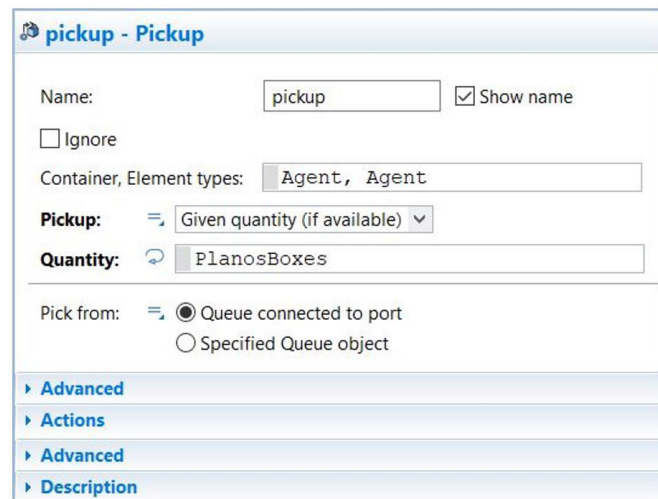


Figura 40. Definición del bloque de modelado de procesos pickup.

7.1.3.1. Modelado de los Boxes en Anylogic

El agente de la clase "Box", es un tipo de *agente* que se ha creado para ser enrutado a través del modelo, y que representa las vagonetas de almacenamiento de bizcocho denominados boxes (ver 3.1.3.5 Almacenamiento intermedio). Este *agente* servirá para almacenar embebidos dentro de sí los *agentes* "Producto" provenientes de la etapa de Prensado-secado-esmaltado mediante el bloque "pickup". Puesto que el *agente* "Box" no tiene ninguna característica dimensional, se ha definido simplemente a partir de una imagen obtenida de la librería de Anylogic (ver figura 42).

Por otro lado, dentro del *agente* "Main" (el *agente* principal donde se ha definido el diagrama de flujo y la los elementos del espacio continuo) se ha definido el *agente* "Boxes" como una población de *agentes* "Box" equivalente al valor del parámetro "CantidadBoxes"; por lo que esta población de *agentes* "Box" está comprendida en un vector del tipo Boxes [...] (ver figura 41). Se ha adoptado esta solución para poder hacer circular los *agentes* "Box" a lo largo del sistema sin que tengan que ser creados, y posteriormente purgados; de esta manera, los *agentes* "Box" recirculan de manera indefinida a lo largo del sistema como lo harían en un sistema real. Además, esta estrategia permite que cada *agente* "Box" tenga una identificación única ($Boxes[i] \forall i \in \{0, CantidadBoxes - 1\}$); ello abre un abanico de posibilidades para plantear estrategias con las cuales gestionar el movimiento de estos *agentes* a lo largo del sistema.

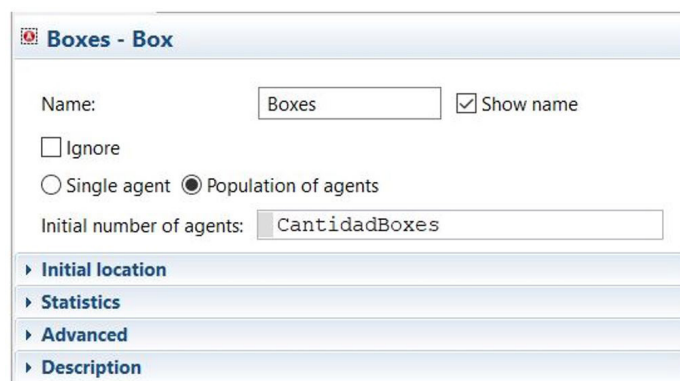


Figura 41. Definición de la población de agentes Boxes.

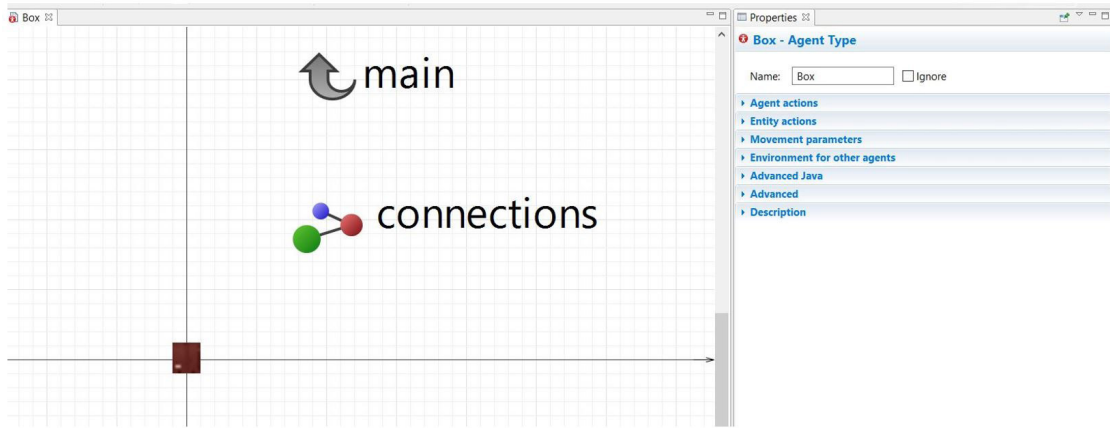


Figura 42. Definición del agente box.

7.1.3.2. Area de Picking del final de etapa Prensado-secado-esmaltado (I)

Atendiendo a las características de los recursos de producción (ver punto 3.4 Características de los recursos de la línea de producción), desde el punto de vista de los *agentes* tipo “Box” el área de picking del final de la línea de esmaltado se ha modelado de la siguiente manera. Cuando el sistema se encuentra funcionando en régimen permanente, siempre hay un *agente* en los bloques tipo *queue* “BoxEnEspera” y “BoxEncarga” (ver figura 39). De manera que, cuando la cantidad de *agentes* en la cola del bloque “FinLineaEsmaltado” es igual al parámetro “PlanosBoxes”, el *agente* tipo “Box” ubicado en el bloque de modelado de procesos “BoxEnCarga” pasa al bloque “pickup”. A continuación, el *agente* que permanece en el bloque “BoxEnEspera” pasa al bloque “BoxEncarga” y desde el bloque “enter” se introduce en el flujo de procesos un nuevo *agente* “Box” que acaba almacenado en el bloque “BoxEnEspera”.

Para poder conseguir este funcionamiento, se han tenido que introducir en el modelado de procesos dos bloques tipo *hold*, “hold1” y “hold2” (ver figura 39). De manera que cuando un *agente* pasa a través del bloque “hold1” se ejecuta la función `hold1.block()` (On Enter ver figura 43); y cuando un *agente* entra en el bloque “BoxEnEspera”, se ejecuta la función `hold2.block()` (On Enter ver figura 44) para bloquear el flujo de *agentes*. De esta manera se queda almacenado un agente en el bloque “BoxEnEspera” y otro en el bloque “BoxEncarga”. Cuando la cantidad de *agentes* en la cola del bloque “FinLineaEsmaltado” es igual al parámetro “PlanosBoxes” se ejecutan las funciones `hold1.unblock()` y `hold2.unblock()` (On Enter ver figura 45) y cada *agente* “Box” avanza una posición en el diagrama de flujo.

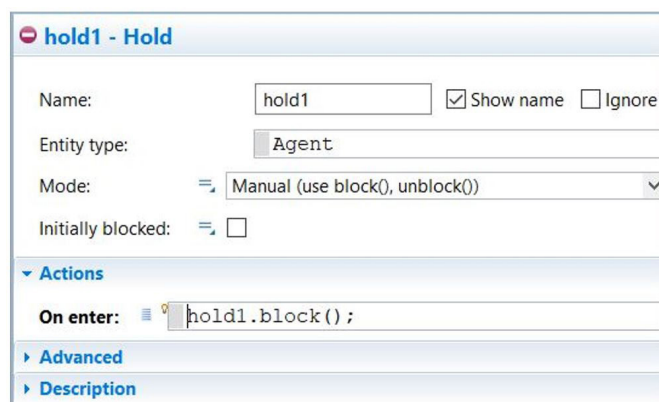


Figura 43. Definición del bloque de modelado de procesos hold1.

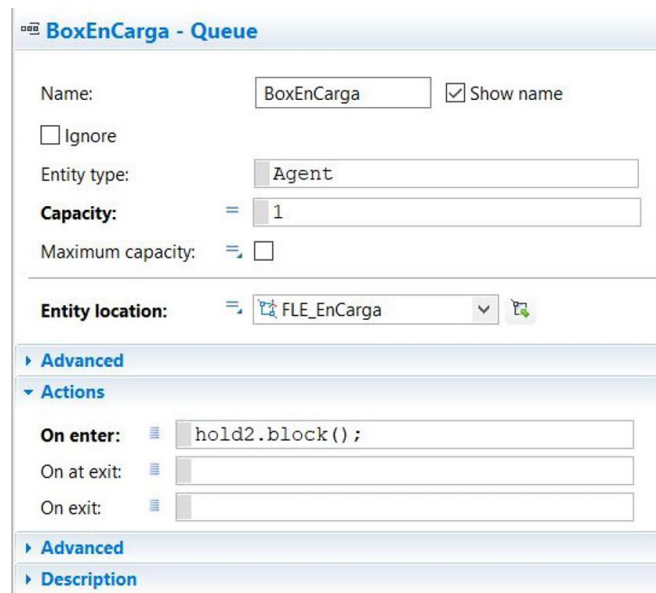


Figura 44. Definición del bloque de modelado de procesos BoxEnCarga.

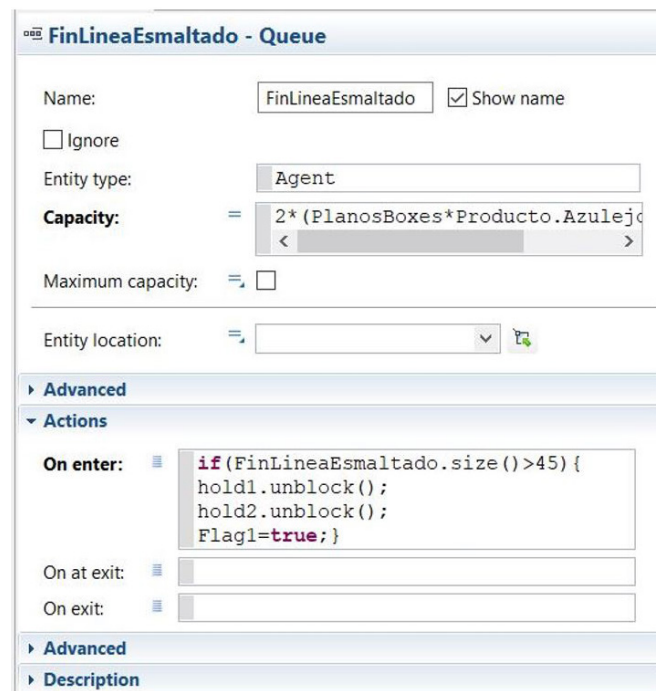


Figura 45. Definición del bloque de modelado de procesos FinLineaEsmaltado.

Por otro lado, se ha determinado una ubicación física para los bloques de modelado de procesos que definen el enrutamiento de los *agentes* “Box” dentro del diagrama de flujo. Esta ubicación se ha definido dentro de la *Network* que modela el almacén intermedio. Para ello, se han creado los *nodes* “FLE_Recepcion”, “FLE_EnCarga” y “FLE_EnPicking” (ver figura 50) y a continuación se han asignado a sus respectivos bloques. Al bloque “BoxEnEspera” se le ha asignado el *node* “FLE_Recepción” (**Entity location** ver figura 46), Al bloque “BoxEnCarga” se le ha asignado el *node* “FLE_EnCarga” (**Entity location** ver figura 45) y al bloque “moveTo” se le ha asignado el *node* “FLE_EnPicking” (**Entity location** ver figura 56).

Y, por último, para poder modelar la ubicación física de los boxes dentro de un almacén intermedio de estas características (ver 3.1.3.5 Almacenamiento intermedio) se ha creado un *RackSystem* denominado “ParqueBoxesVacíos” basado en dos *PalletRack*, “BoxesVacíos1” y “BoxesVacíos2” (ver Anexo II punto 2.6.2

PalletRack) (ver figura 50). Este *RackSystem* tiene la función de almacenar los *agentes* "Box" que no lleven agregados dentro de sí *agentes* "Producto". Estos *PalletRack* están conformados de un pasillo con 30 posiciones de almacenaje de un solo nivel a ambos lados, distribuidas de manera longitudinal; y cuentan con una profundidad suficiente para almacenar un "Box" sin problemas de colisión (ver figura 47).

Figura 46. Definición del bloque de modelado de procesos BoxEnEspera.

Figura 47. Definición del PalletRack BoxesVacios1.

7.1.3.3. Modelado de los LGV en Anylogic

Para poder modelar el movimiento de boxes de un almacén intermedio de estas características (ver 3.1.3.5 Almacenamiento intermedio), se ha tenido que crear un *agente* para emular el comportamiento de un LGV. Este *agente* servirá para almacenar embebidos dentro de sí y transportar los *agentes* "Box" que se encuentre en las distintas ubicaciones de la *Network*. Puesto que el *agente* "LGV" no tiene ninguna característica dimensional, se ha definido simplemente a partir de una imagen obtenida de la librería de Anylogic (ver figura 48).

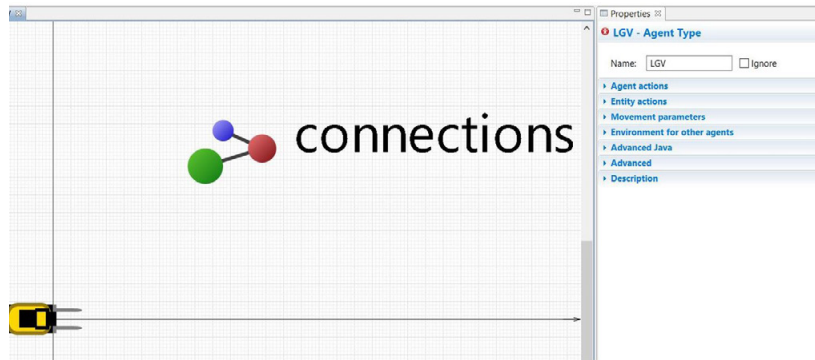


Figura 48. Definición del agente LGV.

Una vez se ha creado este agente, se ha asociado al *ResourcePool* como un recurso tipo *moving* (ver Anexo II punto 2.10 *ResourcePool*) (**New resource unit** ver figura 49) y se le ha asignado una velocidad a partir del parámetro "VelocidadLGV" (**Speed** ver figura 49). Este *ResourcePool* representa el parque de boxes disponible para poder realizar las operaciones de almacenaje intermedio, por lo que, además se ha definido mediante el parámetro "NumLGV" el número de *agentes* LGV disponible (**Capacity** ver figura 49) y su localización física dentro de la *Network* (**Home Location (nodes)** ver figura 49)

En la figura 50 se puede observar la ubicación física dentro de la *Network* del *node* del asociado al *ResourcePool* "LGV", el "node 2".

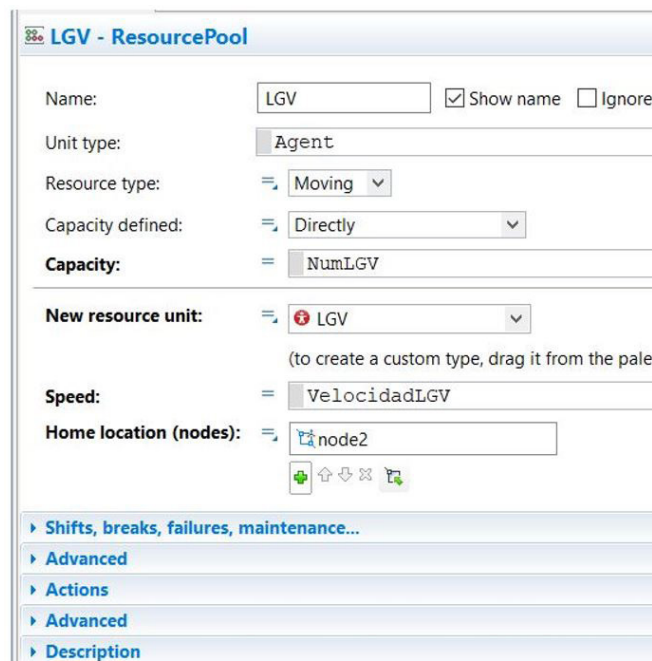


Figura 49. Definición del ResourcePool LGV.

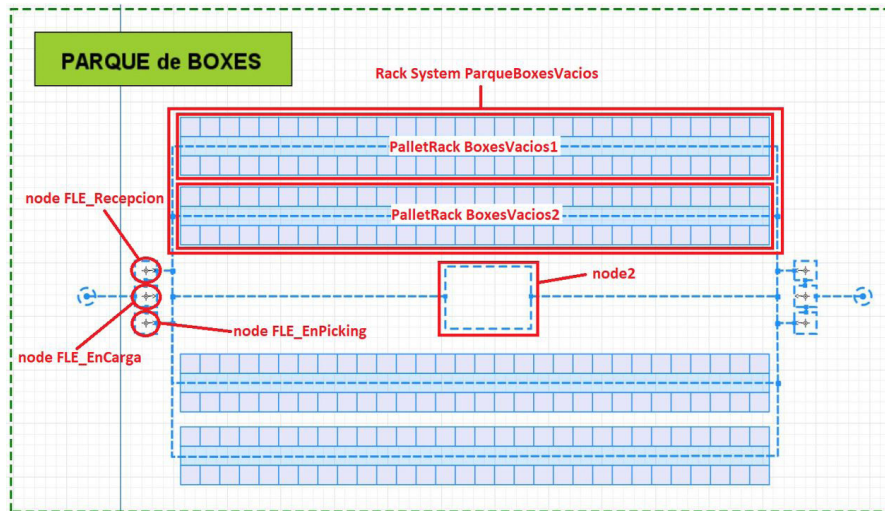


Figura 50. Elementos definidos en el Lay-out la Network.

7.1.3.4. Inicialización del Almacén intermedio

Los *agentes* de la clase "Box" que se han creado están definidos dentro del *agente* "Main" pero no están ubicados dentro de la *Network* "ParqueDeBoxes"; por lo que al inicio de la simulación es necesario crear un procedimiento para poder colocar la población de *agentes* "Box" en sus respectivas posiciones de inicio. Este procedimiento se desarrolla durante la ejecución del *event* "VoidSetUp" (ver punto 7.1.1 Entrada de materia prima) (ver Tabla 7) siguiendo los pasos que se describen a continuación:

- Primero se calcula el número de *agentes* "Box" a introducir en cada *PalletRack* del *RackSystem* "ParqueBoxesVacios" en función del valor del parámetro "CantidadBoxes" y de los parámetros "NumeroRacks" (número de celdas de los *PalletRack*) y "ProfundidadRack" (cantidad posible de elementos a almacenar en las celdas).
- Después, se selecciona un número de *agentes* igual al valor de la variable local "cantidad" de la población de "Boxes". Esta acción se realiza mediante la función `Boxes.get()`; y embebiendo esta función dentro de `BoxesVacios1.put()` se coloca cada *agente* "Box" seleccionado en cada una de las posiciones de las celdas del *PalletRack* "BoxesVacios1"
- A continuación, se da alta en la *Network* "ParqueDeBoxes" a cada uno de los *agentes* que se han introducido dentro del espacio continuo del modelo de simulación mediante la función `Boxes.get().setNetwork()`.
- Por último, se repite este mismo procedimiento en el *PalletRack* "BoxesVacios2" para llenarlo también de *agentes* "Box".

```

//En este bloque de código, se inserta la cantidad
//de boxes con la que se va a realizar la simulación
//en una posición específica del almacén enracado.
int j;

int l;

int k;

int cantidad;

cantidad=((CantidadBoxes/NumeroRacks)-ProfundidadRack)/2;

k=0;

////////////////////////////////////

for (int i=0;i<=cantidad;i++){

for(j=0;j<=1;j++){

//En esta línea de código se pone el agente Boxes[]
//en la posición del almacén
BoxesVacios1.put(j,i,0,true,Boxes.get(k));

//En esta otra el agente Boxes[] se da de alta
//en espacio continuo de la Network
Boxes.get(k).setNetwork(ParqueDeBoxes);

k=k+1;}}

////////////////////////////////////
//Se repite el mismo procedimiento para llenar el
//segundo almacén de Boxes.
for (int i=0;i<=cantidad;i++){

for(j=0;j<=1;j++){
//En esta línea de código se pone el agente Boxes[]
//en la posición del almacén
BoxesVacios2.put(j,i,0,true,Boxes.get(k));

//En esta otra el agente Boxes[] se da de alta
//en el espacio continuo de la Network
Boxes.get(k).setNetwork(ParqueDeBoxes);

k=k+1;}}

|

//Se inicializa el sistema posicionando dos boxes
//vacios en el final de la línea de esmaltado
enter.take(BoxesVacios1.get(0,0,0,0));
enter.take(BoxesVacios1.get(1,0,0,0));
////////////////////////////////////
//enter.take(Boxes.get(Contador));////////////////////////////////////

////////////////////////////////////

```

Tabla 7. Inicialización del sistema: Introducción de boxes en el RackSystem y posicionamiento inicial de los boxes.

7.1.3.5. Área de Picking del final de etapa Prensado-secado-esmaltado (II)

Una vez se ha podido modelar el entorno del Area de Picking del final de etapa Prensado-secado-esmaltado, queda definir movimiento de los *agentes* "Box" desde la perspectiva del *recurso* LGV. Como se ha explicado en el punto 7.1.3.2 Area de Picking del final de etapa Prensado-secado-esmaltado (I), cuando el sistema se encuentra funcionando en régimen permanente, siempre hay un *agente* en los bloques tipo *queue* "BoxEnEspera" y "BoxEncarga". Ello significa que debe haber una alimentación de *agentes* a la entrada de esta parte del proceso.

El problema radica que, al haber planteado un modelo de simulación multiparadigma, al mismo tiempo que el *agente* "Box" tiene que recorrer el flujo de procesos, también tiene que desplazarse a través del espacio continuo de la *Network* "ParqueDeBoxes". Para poder realizar esta acción de manera simultánea, se ha utilizado bloque de modelado de procesos del tipo *RackPick* (ver Anexo II punto 2.14 RackPick) denominado "RecogerBoxVacio" (ver figura 39). Por lo que cuando entre un *agente* desde el flujo de procesos en el bloque "RecogerBoxVacio", un *recurso* LGV se desplazará hasta su posición dentro de la *Network*, lo embeberá dentro de sí, y lo transportará hasta la posición donde se encuentra el bloque "BoxEnEspera"

Figura 51. Definición del bloque de modelado de procesos RecogerBoxVacio (I).

Para poder definir este bloque, se ha tenido que asociar al *RackSystem* desde donde va a recoger los *agentes* "Box" (**Pallet rack / Rack system** ver figura 51) y después se le ha tenido que asignar un node donde de destino donde descargar los agentes (**Node** ver figura 51).

Por otro lado, se le ha asignado el *ResourcePool* LGV (**Moving resource** ver figura 52) para que el bloque de modelado de procesos pueda contar con recursos disponibles con los que realizar las operaciones de transporte de agentes. Este *ResourcePool* se ha definido de manera que se utilice un *recurso* LGV por cada *agente* (**Resource sets** ver figura 52) y que los agentes se embeban de uno en uno en el recurso (**Seize policy** ver figura 52). De manera adicional, se ha establecido una política de gestión de recursos por la cual, cuando se haya entregado el *agente* en la ubicación de destino, o el *recurso* no esté haciendo nada; el este debe volver a la ubicación de su *ResourcePool*.

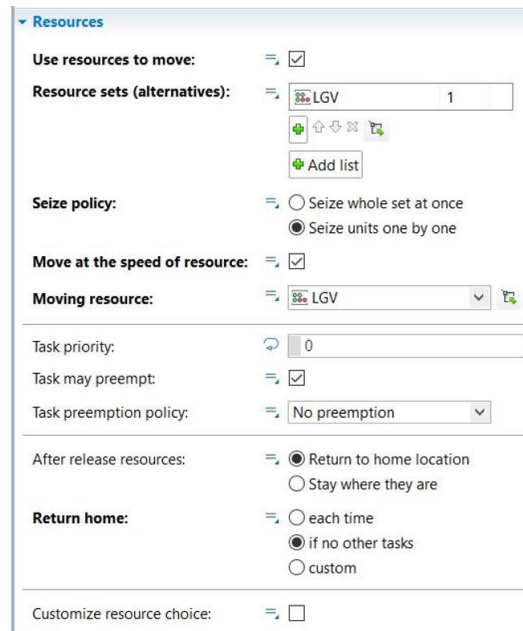


Figura 52. Definición del bloque de modelado de procesos RecogerBoxVacio (II).

Volviendo al entorno del flujo de procesos, para que a este bloque lleguen *agentes* se ha definido a la entrada de esta etapa un bloque de modelado de procesos del tipo *enter* (ver Anexo II Punto 2.13 Enter) denominado "enter". Cuando se hace una llamada a este bloque desde la función `enter.take()`, el bloque toma al *agente* seleccionado y lo enruta a lo largo del flujo de procesos. Esta función se ejecuta desde el *event* "VoidLoop" (ver punto 6.2.2.1 Operación de prensado); donde se ha establecido un procedimiento para que, de manera periódica, el bloque "enter" introduzca *agentes* siempre que se cumplan las condiciones de funcionamiento del proceso en régimen estacionario. A continuación, se expone de manera detallada como se desarrolla este procedimiento dentro de las líneas de código del *event* "VoidLoop" (ver Tabla 7) siguiendo los pasos que se describen a continuación:

- En primer lugar, se pregunta mediante las funciones `BoxesBizcocho.hasSpace()` y `BoxesBizcocho.hasSpace()` si en el *PalletRack* "BoxesBizcocho" (el *PalletRack* donde se almacenan los *agentes* producto que vienen de la línea de Prensado-secado-esmaltado; se explicará en capítulos posteriores) existen celdas vacías. Después se pregunta mediante la función `ParqueBoxesVacios.size()` si en el *RackSistem* "ParqueBoxesVacios" quedan *agentes* "Box" disponibles.
- Más tarde se pregunta mediante la función `BoxesVacios1.size()` si en el *PalletRack* "BoxesVacios1" hay *agentes* "Box" disponibles. En esta línea de comandos, además se comprueba también el estado de la variable local "Stock". Esta variable sirve para forzar a que haya una alternancia a la hora de seleccionar el *PalletRack* desde donde se va a recoger el *agente* "Box". De manera que en cada ciclo de ejecución del *event* "VoidLoop", se ejecutará de manera alternativa, uno de los dos bloques de código correspondientes a cada *PalletRack*.
- A continuación, se comprueba mediante la función `RecogerBoxVacio.queueSize()` que no hay un box en tránsito; se comprueba mediante la función `BoxEnEspera.size()` que no hay un box en el bloque BoxEnEspera; y por último, mediante el valor de la variable *Flag1* se comprueba que el bloque Box en carga ha terminado.
- Una vez realizadas todas estas verificaciones, mediante la función `BoxesVacios2.isFree()` y el código que le precede, se recorren todos los valores de las celdas de posición del *PalletRack* hasta encontrar una que esté ocupada. Cuando se ha encontrado, mediante la función `BoxesVacios1.get()` se comprueba que en esa celda hay un *agente*. Y por último, se embebe esta función dentro de `enter.take()` para enrutar este *agente* dentro del flujo de procesos.

- Por último, más abajo se replica este bloque de programación, pero referenciado al *PalletRack* "BoxesVacios2". Como se ha mencionado antes, gracias a la variable local "Stock" cada que se ejecute el event "VoidLoop" solo se ejecutará uno de los dos bloques de código.

```

//Se pregunta si hay espacio en los PalletRack BoxesBizcocho
if (BoxesBizcocho.hasSpace() || BoxesBizcocho1.hasSpace()){
|
|
//Se pregunta si hay Boxes en el SystemRack ParqueBoxesVacios.
if(ParqueBoxesVacios.size())>0){
|
//Se pregunta si hay Boxes en el PalletRack BoxesVacios.
//La variable Stock sirve para que haya alternancia entre los dos PalletRack a
la
//hora recoger boxes.

if(BoxesVacios1.size())>0 && Stock==0){

//Se comprueba que no hay ya un Box en transito; que no hay ya un box en el
//bloque BoxEnEspera; y que el BoxEnCarga ha terminado de cargar.
if(RecogerBoxVacio.queueSize())<1 && BoxEnEspera.size())<1

&& Flag1==true){
//Mediante este algoritmo, se recorren todas las celdas del PalletRack en busca
de una que no esté vacía.
while(BoxesVacios1.isFree(x1,y1,0)){
x1=x1+1;

if(x1>1){

x1=0;

y1=y1+1;

Stock=1;}}

//Una vez se ha localizado esta celda, se comprueba que realmente almacena un
Box.
if (BoxesVacios1.get(x1,y1,0,0)!=null)
//En caso afirmativo, se enruta este Box en el flujo de procesos.
{enter.take(BoxesVacios1.get(x1,y1,0,0));}
//Por último, se inicializan las variables locales de búsqueda.
y1=0;
x1=0;}
//En el caso en el que quede un box suelto en BoxesVacios1 se reinicia la
variable Stock
if (BoxesVacios1.size())==0 && Stock==0){Stock=1;}

////////////////////////////////////
//Se duplica el algoritmo para utilizar boxes
//desde los dos almacenes
////////////////////////////////////
if(BoxesVacios2.size())>0 && Stock==1){

if(RecogerBoxVacio.queueSize())<1 && BoxEnEspera.size())<1

&& Flag1==true){

while(BoxesVacios2.isFree(x1,y1,0)){

x1=x1+1;

```

```

if(x1>1){
x1=0;

y1=y1+1;

Stock=0;}}

if (BoxesVacios2.get(x1,y1,0,0)!=null){
enter.take(BoxesVacios2.get(x1,y1,0,0));}}

y1=0;
x1=0;}

else {MarchaParoPrensa.block();}}

else {
MarchaParoPrensa.block();

MarchaParoEsm1.block();

MarchaParoEsm2.block();}
////////////////////////////////////
////////////////////////////////////

```

Tabla 8. Rutina de interrupción periódica: Gestión del Área de picking del final de la línea de esmaltado.

Siguiendo con el objeto del desarrollo del punto anterior, al inicio de la simulación, en los bloques "BoxEnEspera" y "BoxEnCarga" no se encuentra ubicado ningún agente "Box". Esto plantea un problema, porque el algoritmo anterior está preparado para funcionar en régimen permanente. Como solución a este problema, al final del event "VoidSetUp" (ver tabla 7) se ejecuta dos veces la función `enter.take()` para enrutar por el flujo de operaciones los dos primeros agentes almacenados en el *PalletRack* "BoxesVacios1".

7.1.4. ALMACENAMIENTO INTERMEDIO DE LOS AZULEJOS (II)

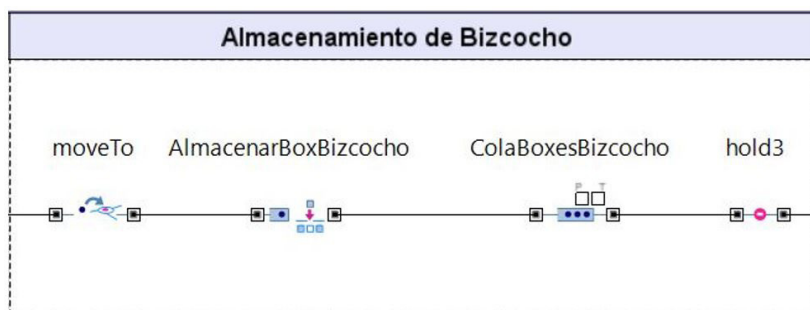


Figura 53. Almacenamiento intermedio de Bizcocho.

Los *agentes* que continúan por el flujo de procesos desde el área de picking del final de la línea de esmaltado llegan a la etapa de almacenamiento intermedio de Bizcocho. Desde el punto de vista del flujo de procesos, el funcionamiento es muy sencillo; los *agentes* recorren los bloques de modelado de procesos hasta llegar a uno del tipo *queue* denominado "ColaBoxesBizcocho", donde permanecen allí hasta que el bloque "hold3" desbloquea el flujo de procesos (ver figura 53). Pero como se ha comentado en el punto anterior, al haber planteado un modelo de simulación multiparadigma, al mismo tiempo que el *agente* "Box" tiene que recorrer el flujo de procesos, también se desplaza a través del espacio continuo del modelo de simulación que se ha diseñado en la *Network* "ParqueDeBoxes".

Partiendo de esta premisa, se ha creado un *RackSystem* denominado "ParqueBoxesBizcocho" basado en dos *PalletRack*, "BoxesBizcocho" y "BoxesBizcocho1" (ver Anexo II punto 2.6.2 *PalletRack*) (ver figura 55). Este *RackSystem* tiene la función de almacenar los *agentes* "Box" que lleven agregados dentro de sí *agentes* "Producto". Estos *PalletRack* están conformados de un pasillo con 30 posiciones de almacenaje de un solo nivel a ambos lados, distribuidas de manera longitudinal; y cuentan con una profundidad suficiente para almacenar un "Box" sin problemas de colisión (ver figura 54).

BoxesBizcocho - Pallet Rack

Name: Ignore

Visible on upper level Lock

Visible: yes

Type: ▾

Number of cells is: Defined explicitly
 Calculated based on the cell width

Number of cells:

Number of deep positions:

Number of levels:

Level height:

▸ Appearance

▸ Position and size

▸ Advanced

▸ Description

Figura 54. Definición de PalletRack BoxesBizcocho.

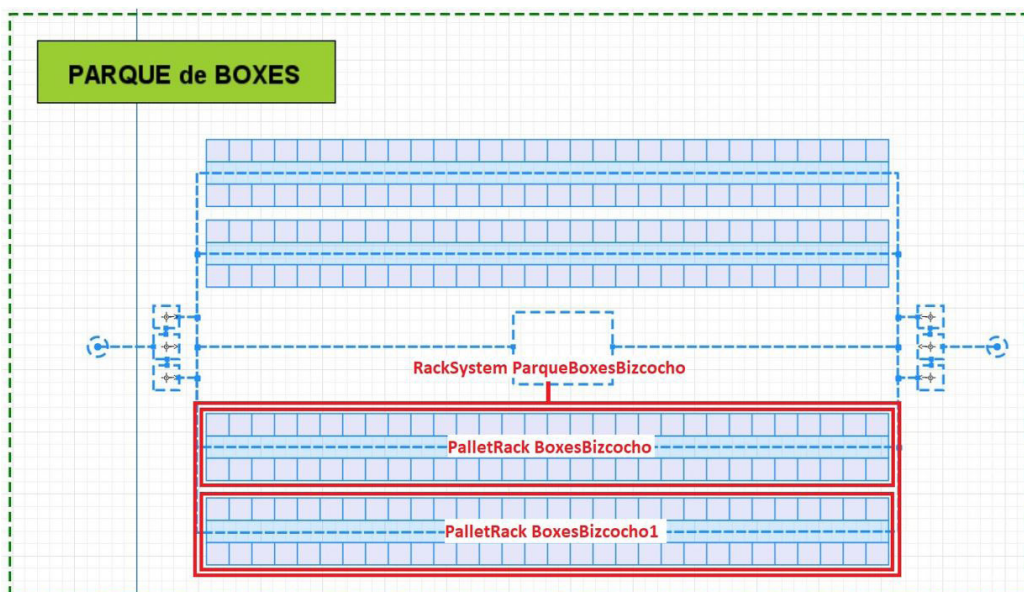


Figura 55. Elementos definidos en el Lay-out de la Network II.

7.1.4.1. Funcionamiento del almacén intermedio de Bizcocho

Volviendo al entorno del flujo de procesos, cuando un *agente* "Box" salga del bloque "PickUp" con *agentes* "Producto" embebidos, pasará al bloque de modelado de procesos del tipo *MoveTo* (ver Anexo II Punto 2.18 *MoveTo*) denominado "moveTo"; y de manera simultánea al node "FLE_EnPicking" (ver figura 56). Una vez el *agente* se encuentra en la nueva ubicación física, este avanzara al bloque "AlmacenarBoxBizcocho".

Este es un bloque de modelado de procesos del tipo *RackStore* que tiene como función transportar al *agente* "Box" a una celda dentro del *RackSystem* "ParqueBoxesBizcocho". Por lo que, en ese momento, un *recurso* LGV se desplazará hasta el *node* "FLE_EnPicking", lo embeberá dentro de sí, y lo transportará hasta su nueva posición dentro del *RackSystem*.

Para poder definir este bloque, se determinó la ubicación dentro de la *Network* desde donde se recogerán los *agentes* (**Node** ver figura 57), y se ha asociado este bloque al *RackSystem* donde se depositarán los *agentes* que vayan a ser transportados. (**Pallet rack / Rack system** ver figura 57). Por otro lado, se le ha asignado el *ResourcePool* LGV (**Moving resource** ver figura 58) para que el bloque de modelado de procesos pueda contar con *recursos* disponibles con los que realizar las operaciones de transporte de *agentes*. Este *ResourcePool* se ha definido de manera que se utilice un *recurso* LGV por cada *agente* (**Resource sets** ver figura 58) y que los *agentes* se embeban de uno en uno en el recurso (**Seize policy** ver figura 58). De manera adicional, se ha establecido una política de gestión de recursos por la cual, cuando se haya entregado el *agente* en la ubicación de destino, o el *recurso* no esté haciendo nada; el este debe volver a la ubicación de su *ResourcePool*.

moveTo - MoveTo

Name: Show name

Ignore

Entity type:

Entity: moves to
 is placed (jumps) to

Destination:

Node:

... with offset:

Set agent's speed:

▶ Actions

▶ Advanced

▶ Description

Figura 56. Definición del bloque de modelado de procesos moveTo.

AlmacenarBoxBizcocho - RackStore

Name: Show name

Ignore

Entity type:

Pallet rack / Rack system:

The cell is:

Choose cell closest to:

Move entity to:

Elevation time per level:

Use delay:

Entity location (queue):

Figura 57. Definición del bloque de modelado de procesos AlmacenarBoxBizcocho I.

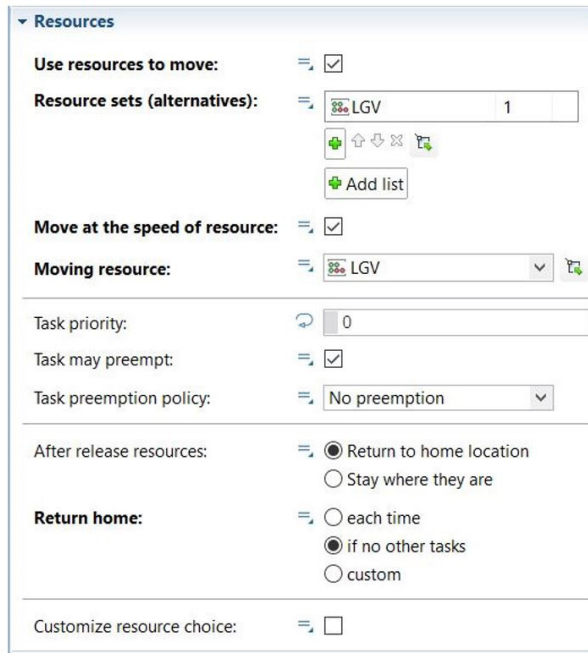


Figura 58. Definición del bloque de modelado de procesos AlmacenaBoxBizcocho II.

Una vez los *agentes* son transportados al *RackSystem*, de manera simultánea avanzan por el flujo de procesos, y llegan al bloque tipo *queue* “ColaBoxesBizcocho” (ver figura 53). Puesto que a la salida de este bloque hay un bloque de modelado de procesos tipo *hold* denominado “hold3”, el flujo de *agentes* se detiene tanto en el bloque “ColaBoxesBizcocho”, como en el *RackSystem*. El bloque “hold3” sólo se desbloqueará cuando se haga se le haga una llamada desde la función `hold3.unlock()`, esto simulará una orden de pedido desde la siguiente etapa del proceso de producción.

7.1.5. ALMACENAMIENTO INTERMEDIO DE LOS AZULEJOS (III)

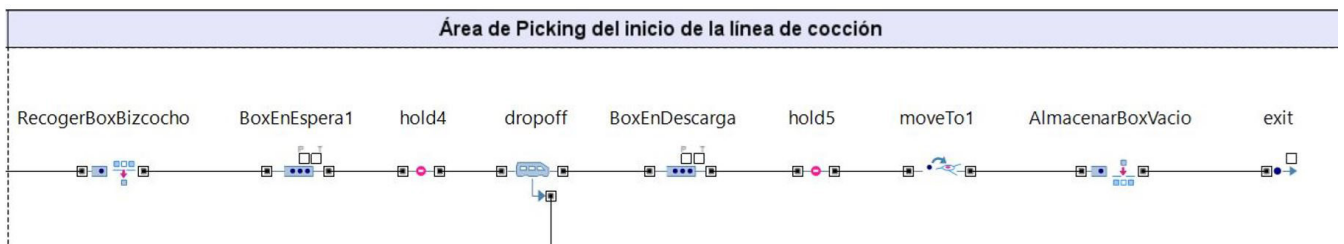


Figura 59. Área de Picking del final del final del inicio de la línea de cocción.

Los *agentes* que continúan por el flujo de procesos, llegan al Área de picking del inicio de la línea de cocción. Esta área de picking se ha modelado mediante un bloque de modelado de procesos del tipo *queue* llamado “BoxEnEspera1” (ver figura 60), conectado a un bloque del tipo *Dropoff* (ver Anexo II punto 2.1.6 Dropoff) (ver figura 62); que tiene conectado a su salida otro bloque del tipo *queue* llamado “BoxEnDescarga” (ver figura 61). De manera que, cuando el *agente* “Box” pasa por el bloque *Dropoff*, desagrega los *agentes* “Producto” que tiene embebidos a través de su línea de descarga. Esta línea de descarga está conectada a su vez un bloque de modelado de procesos del tipo *queue* llamado “InicioHorno” (ver figura 63), desde donde se dispensan los *agentes* “Producto” a la siguiente etapa del flujo de procesos.

Una vez ha tenido lugar la descarga de los *agentes* embebidos, el *agente* “Box” avanza hasta el siguiente bloque de modelado de procesos, y permanece allí hasta que la siguiente etapa haya consumido los *agentes* desagregados.

BoxEnEspera1 - Queue

Name: Show name

Ignore

Entity type:

Capacity: =

Maximum capacity:

Entity location:

▶ **Advanced**

▶ **Actions**

▶ **Advanced**

▶ **Description**

Figura 60. Definición del bloque de modelado de procesos BoxEnEspera1.

BoxEnDescarga - Queue

Name: Show name

Ignore

Entity type:

Capacity: =

Maximum capacity:

Entity location:

▶ **Advanced**

▶ **Actions**

▶ **Advanced**

▶ **Description**

Figura 61. Definición del bloque de modelado de procesos BoxEnDescarga.

dropoff - Dropoff

Name: Show name

Ignore

Container, Element types:

Dropoff:

Condition:

▶ **Advanced**

▼ **Actions**

On enter:

On dropoff:

On exit:

On dropoff exit:

▶ **Advanced**

▶ **Description**

Figura 62. Definición del bloque de modelado de procesos Dropoff.

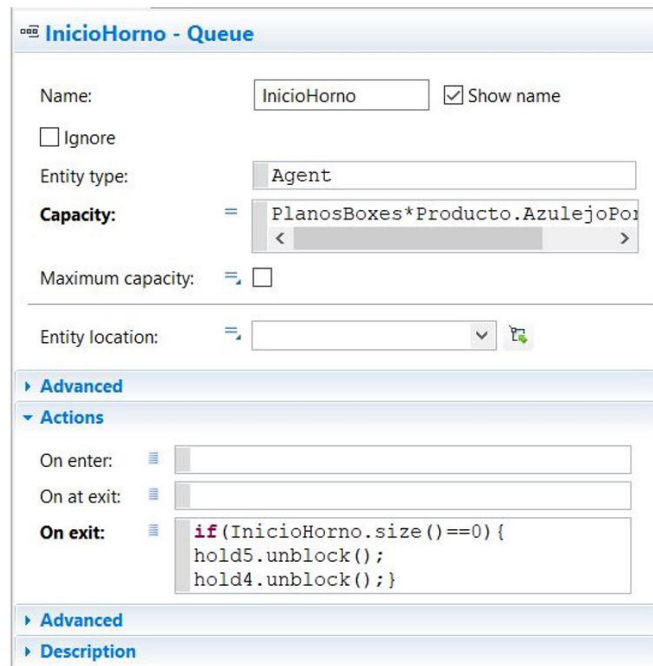


Figura 63. Definición del bloque de modelado de procesos InicioHorno.

7.1.5.1. Área de Picking del inicio de la línea de cocción.

Atendiendo a las especificaciones de diseño (ver punto 3.4 Características de los recursos de producción), el área de picking del inicio de la línea de cocción se ha modelado de la siguiente manera. Cuando el sistema se encuentra funcionando en régimen permanente, siempre hay un *agente* en los bloques “BoxEnEspera1” y “BoxEnDescarga” (ver figura 59). De manera que, cuando la siguiente etapa ha consumido los *agentes* desagregados del *agente* “Box”, este pasa al bloque “AlmacenarBoxVacio” a la espera de ser almacenado en la *RackSystem* “ParqueBoxesVacios”.

Para poder conseguir este funcionamiento, se han tenido que introducir en el flujo de procesos dos bloques tipo *hold*, “hold4” y “hold5” (ver figura 59). De manera que, cuando un *agente* pasa a través del bloque “hold4”, se ejecuta la función `hold4.block()` (On Enter ver figura 64); y cuando un *agente* entra en el bloque “dropoff”, se ejecuta la función `hold5.block()` (On Enter ver figura 62) para bloquear el flujo de *agentes*. De esta manera, se queda almacenado un *agente* en el bloque “BoxEnEspera1”, y otro en el bloque “BoxEnDescarga”. Cuando la cantidad de *agentes* en la cola del bloque “InicioHorno” es igual a 0 se ejecutan las funciones `hold4.unblock()` y `hold5.unblock()` (On Enter ver figura 63) y cada *agente* “Box” avanza una posición en el diagrama de flujo.

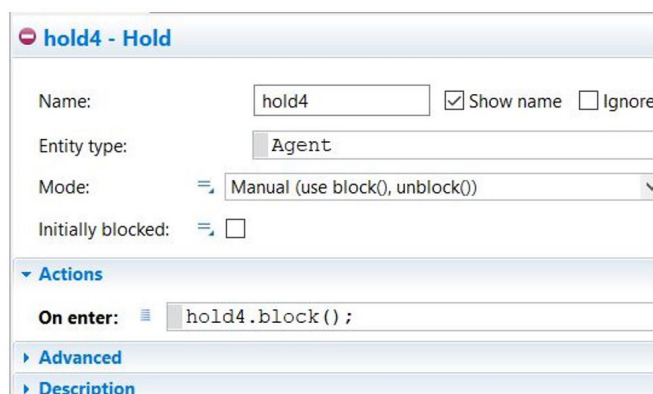


Figura 64. Definición del bloque de modelado de procesos hold4.

Por otro lado, se ha determinado una ubicación física dentro de la *Network* "ParqueDeBoxes" para los bloques de modelado de procesos que definen el enrutamiento de los *agentes* "Box" dentro del diagrama de flujo. Para ello, se han creado los *nodes* "H_Recepcion", "H_EnDescarga" y "H_EnPicking" (ver figura 65) y a continuación se han asignado a sus respectivos bloques. Al bloque "BoxEnEspera1" (**Node** ver figura 61); Al bloque "BoxEnCarga" se le ha asignado el *node* "H_EnDescarga" (**Entity location** ver figura 62) y al bloque "moveTo1" se le ha asignado el *node* "H_EnPicking" (**Entity location** ver figura 68).

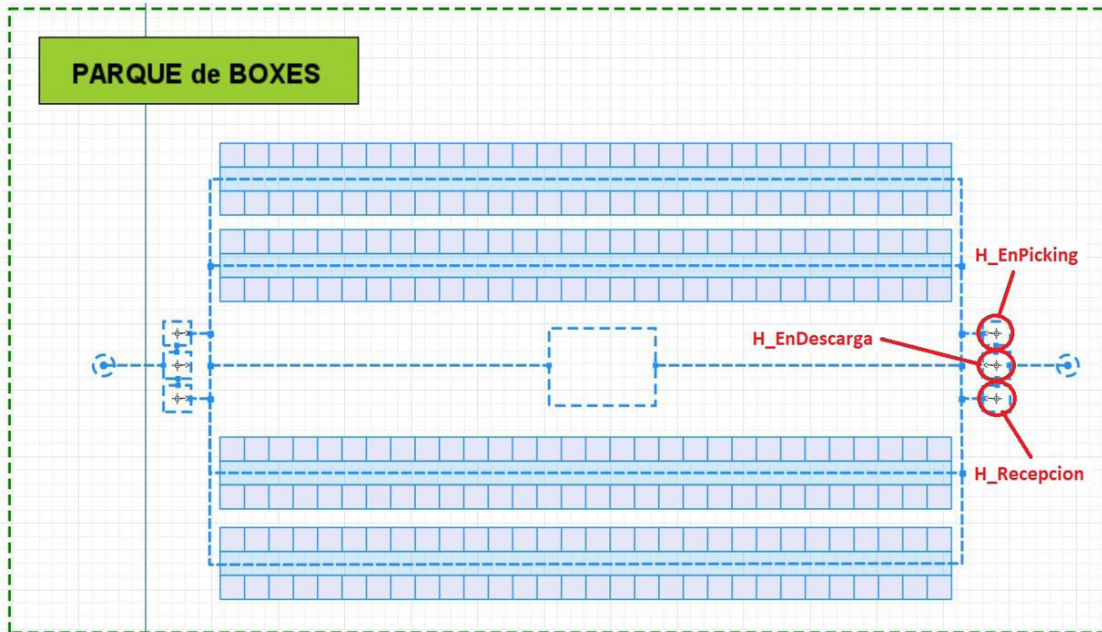


Figura 65. Elementos definidos en el Lay-out de la Network (II).

7.1.5.2. Entrada de agentes en el Área de Picking del inicio de la línea de cocción.

Como se ha comentado en el punto anterior, el bloque "hold3", detiene el flujo de *agentes* desde el bloque "ColaBoxesBizcocho", y se sólo desbloquea cuando se le haga una llamada desde la función `hold3.unlock()`. Por lo que es necesario desarrollar un procedimiento para ejecutar esta función cuando se requiera. Este cometido se lleva a cabo desde el event "VoidLoop" (ver tabla 8).

Para ello, se pregunta mediante la función `RecogerBoxBizcocho.queueSize()` si hay un *agente* que esté siendo transportado desde el *RackSystem* "ParqueBoxesBizcocho" hasta el bloque "BoxEnEspera1"; se pregunta mediante la función `BoxEnEspera1.size()` si hay un *agente* en el bloque "BoxEnEspera1"; y mediante la variable "Flag2", se fuerza a que los *agentes* avancen por el flujo de procesos de uno en uno. Cuando todas estas condiciones se cumplen, se hace una llamada al bloque "hold3" desde la función `hold3.unlock()` y un *agente* avanza desde el bloque "ColaBoxesBizcocho" hasta el bloque de modelado de procesos tipo *RackPick* "RecogerBoxBizcocho" (ver figura 66).

El bloque "RecogerBoxBizcocho" tiene la función de trasladar al *agente* hasta el bloque "BoxEnEspera" tanto por el flujo de procesos, como por el espacio continuo de la *Network*. Por lo que, para poder definir este bloque, se le ha asociado el *RackSystem* "ParqueBoxesBizcocho". (**Pallet rack / Rack system** ver figura 66) y después se le ha asignado un *node* donde de destino donde descargar los *agentes* (**Node** ver figura 66).

Por otro lado, se le ha asignado el *ResourcePool* LGV (**Moving resource** ver figura 67) para que el bloque pueda contar con *recursos* disponibles con los que realizar las operaciones de transporte de *agentes*. Este *ResourcePool* se ha definido de manera que se utilice un *recurso* LGV por cada *agente* (**Resource sets** ver figura 67) y que los *agentes* se embeban de uno en uno en el recurso (**Seize policy** ver figura 67). De manera

adicional, se ha establecido una política de gestión de recursos por la cual, cuando se haya entregado el agente en la ubicación de destino, o el recurso no esté haciendo nada; el este debe volver a la ubicación de su ResourcePool.

Volviendo al análisis del enrutamiento del agente "Box" a lo largo del flujo de procesos, al atravesar el bloque "hold3", se ejecuta la función `hold3.block()`; y se niega la variable "Flag2". Por otro lado, cuando el agente sale el bloque "RecogerBoxBizcocho" la variable "Flag2" vuelve a ser verdadera. Llegados a este punto, el flujo de procesos se bloquea a la espera de que se cumplan las condiciones para poder dejar pasar a más agentes.

```
//Se pregunta si hay espacio en los PalletRack BoxesBizcocho

//Se pregunta si hay Boxes en el SystemRack ParqueBoxesVacios.

//Se pregunta si hay Boxes en el PalletRack BoxesVacios.
//La variable Stock sirve para que haya alternancia entre los dos PalletRack a la
//hora recoger boxes.

//Se comprueba que no hay ya un Box en transito; que no hay ya un box en el
//bloque BoxEnEspera; y que el BoxEnCarga ha terminado de cargar.

//Mediante este algoritmo, se recorren todas las celdas del PalletRack en busca de
una que no esté vacía.

//Una vez se ha localizado esta celda, se comprueba que realmente almacena un Box.
//En caso afirmativo,se enruta este Box en el flujo de procesos.
//Por último, se inicializan las variables locales de búsqueda.

//En el caso en el que quede un box suelto en BoxesVacios1 se reinicia la variable
Stock
////////////////////////////////////
//Se duplica el algoritmo para utilizar boxes
//desde los dos almacenes
////////////////////////////////////

////////////////////////////////////
////////////////////////////////////
```

```

////////////////////////////////////
////////////////////////////////////
//Se transporta el Box con el bizcocho cargado a la
//a la zona de recepción de la línea de cocción,
//a la entrada del horno.
    if(RecogerBoxBizcocho.queueSize()<1 && BoxEnEspera1.size()<1
    && Flag2==true){
    hold3.unblock();}
    else {hold3.block();}
////////////////////////////////////
////////////////////////////////////
    
```

Tabla 8. Rutina de interrupción periódica: Gestión del Área de picking del inicio de la línea de cocción.

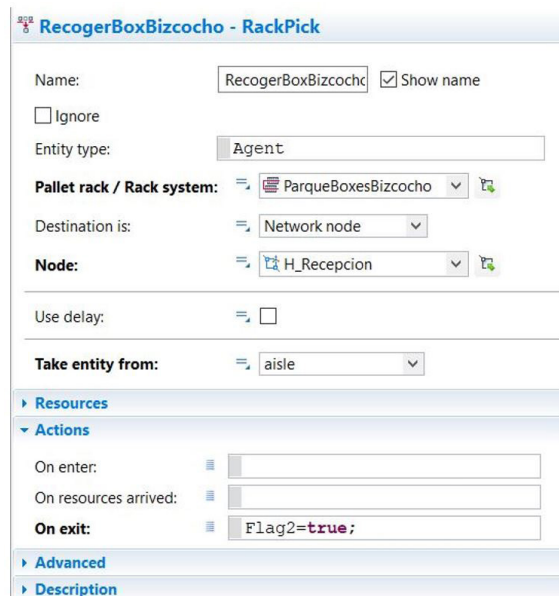


Figura 66. Definición del bloque de modelado de procesos RecogerBoxBizcocho (I).

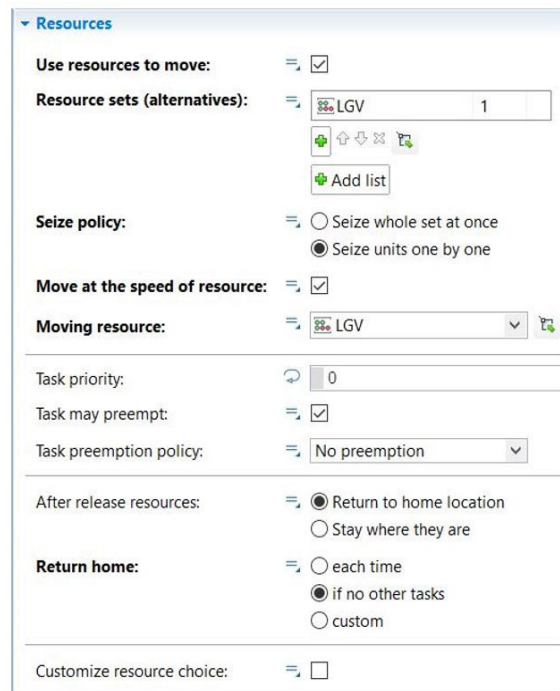


Figura 67. Definición del bloque de modelado de procesos RecogerBoxBizcocho (II).

7.1.5.3. Salida de agentes en el Área de Picking del inicio de la línea de cocción.

Como sea comentado a en este punto, el agente "Box" permanece retenido en el bloque "BoxEnEspera" hasta que en el bloque "InicioHorno" se hayan consumido los agentes desagregados por el bloque "dropoff". Por lo que, cuando se de esa condición, pasará al bloque de modelado de procesos del tipo MoveTo denominado "moveTo1"; y de manera simultánea al node "H_EnPicking" (ver figura 65). Una vez el agente se encuentra en la nueva ubicación física, este avanzara al bloque "AlmacenarBoxVacio".

Este es un bloque de modelado de procesos del tipo RackStore que tiene como función transportar al agente "Box" a una celda dentro del RackSystem "ParqueBoxesVacios". Por lo que, en ese momento, un recurso LGV se desplazará hasta el node "H_EnPicking", lo embeberá dentro de sí, y lo transportará hasta su nueva posición dentro del RackSystem.

Para poder definir este bloque, se determinado la ubicación dentro de la Network desde donde se recogerán los agentes (Node ver figura 69), y se ha asociado este bloque al RackSystem donde se depositarán los agentes que vayan a ser transportados. (Pallet rack / Rack system ver figura 69). Por otro lado, se le ha asignado el ResourcePool LGV (Moving resource ver figura 70) para que el bloque pueda contar con recursos disponibles con los que realizar las operaciones de transporte de agentes. Este ResourcePool se ha definido de manera que se utilice un recurso LGV por cada agente (Resource sets ver figura 70) y que los agentes se embeban de uno en uno en el recurso (Seize policy ver figura 70). De manera adicional, se ha establecido una política de gestión de recursos por la cual, cuando se haya entregado el agente en la ubicación de destino, o el recurso no esté haciendo nada; el este debe volver a la ubicación de su ResourcePool.

Figura 68. Definición del bloque de modelado de procesos moveTo1.

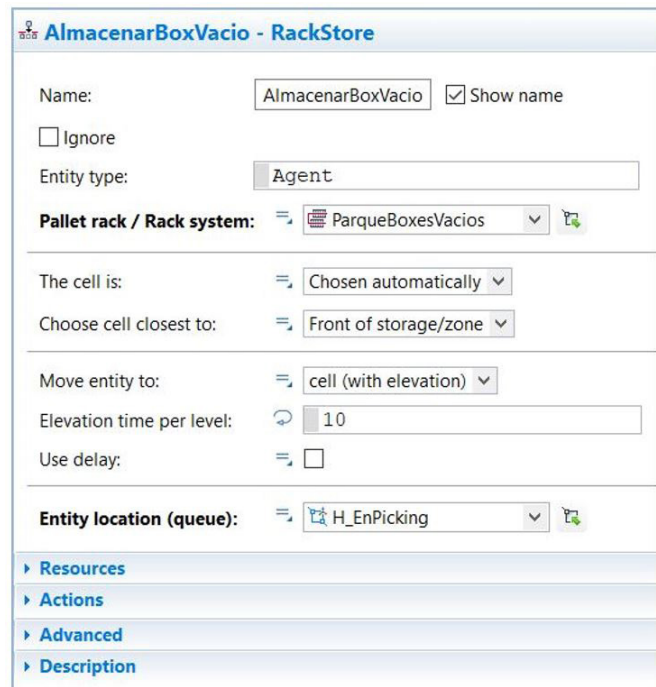


Figura 69. Definición del bloque de modelado de procesos AlmacenarBoxVacio (I).

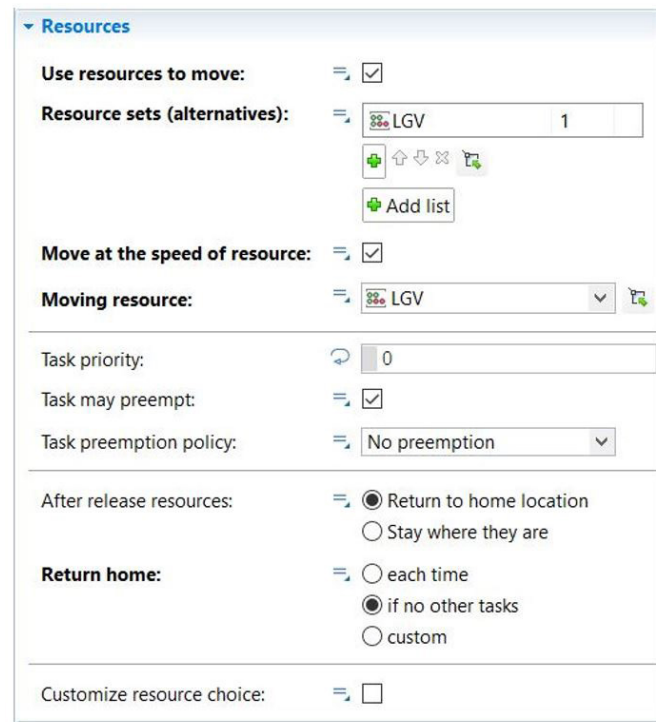


Figura 70. Definición del bloque de modelado de procesos AlmacenarBoxVacio (II).

Por último, una vez el agente haya llegado al RackSystem “ParqueBoxesVacios”, de manera simultánea avanzará por el flujo de procesos hasta llegar al bloque de modelado de procesos tipo Exit denominado “exit”. Llegados a este punto, este bloque retirará al agente del flujo de procesos; y este permanecerá dentro del RackSystem a la espera de ser enrutado de nuevo por el bloque “enter”.

7.1.6. LÍNEA DE COCCIÓN.

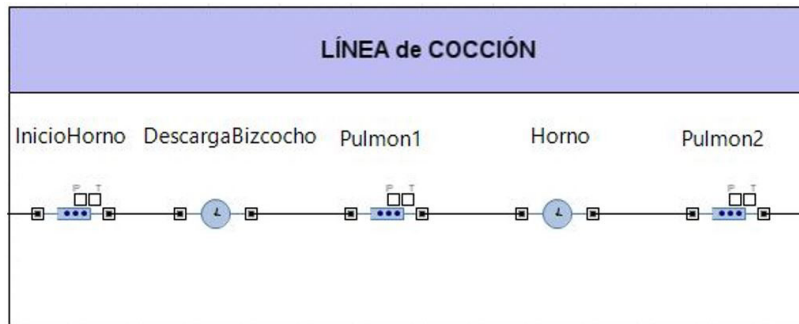


Figura 71. Línea de cocción.

Como se ha comentado en el punto anterior, al inicio de la línea de cocción, hay un bloque de modelado de procesos tipo *queue* que almacena dentro de sí los *agentes* "Producto" desagregados de los *agentes* "Box" que pasan por el bloque "dropoff". Este bloque tiene conectado a su salida un bloque de modelado de procesos tipo *Delay* denominado "DescargaBizcocho" (ver figura 72) que actúa como un dispensador de *agentes*; y delimita la tasa de llegada de estos al siguiente bloque. Esta tasa de llegada se modela a partir del parámetro "VelocidadAlimentacionHorno" (Delay, ver Figura 72); puesto que de esta manera se obtiene el tiempo entre llegada de piezas al horno.

Figura 72. Definición del bloque de modelado de procesos DescargaBizcocho.

Una vez el *agente* ha salido de este acaba en otro bloque del tipo *Delay* denominado "Horno". Este bloque, como su propio nombre indica, modela un horno de solera con rodillos (ver punto 3.4 Características de los recursos de la línea de producción).

Por otro lado, es necesario modelar tanto el ciclo de cocción como la longitud del horno. Para modelar el ciclo de cocción, se ha creado el parámetro "TiempoCocción" (Delay Time, ver figura 73), el cual delimita el tiempo que permanece un *agente* dentro del bloque "Horno". Por otro lado, teniendo en cuenta que la longitud de cada pieza es de 1.2 m, se ha modelado la longitud del horno limitando la capacidad de *agentes* a 200 unidades (Capacity, ver figura 73). De esta manera, considerando el caso en el cual WIP del horno esté lleno, y considerando que los *agentes* se dispondrán dentro de el de manera consecutiva; la longitud mínima desde la primera pieza hasta la última será de 240 metros.

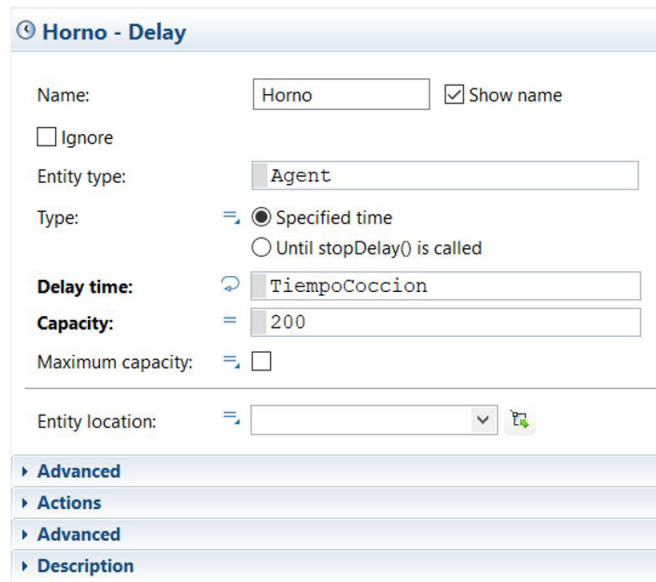


Figura 73. Definición del bloque de modelado de procesos Horno.

7.1.6.1. Pulmones en la línea de cocción.

Atendiendo las características de los recursos de la línea de producción, cuando se caracteriza la distribución en planta de Azulejos S.A. se especifica que existen dos pulmones; uno a la entrada y otro a la salida del horno (ver punto 3.4 Características de los recursos de la línea de producción). Para poder satisfacer este requisito, se han añadido dos bloques tipo *queue* (uno a la entrada y otra salida del bloque “Horno”), denominados “Pulmon1” y “Pulmon2” respectivamente.

Estos bloques de modelado de procesos no tienen ningún efecto cuando se realiza la simulación, ni interactúan de manera efectiva con ningún objeto presente en el modelado de procesos. Pero su presencia en el modelo es interesante porque permite que, en un futuro, cuando se realicen versiones de este modelo más complejas, se disponga de un mecanismo de absorción de variaciones en el flujo de materiales, que además está presente en la realidad.

7.1.7. LÍNEA DE CLASIFICACIÓN Y EMBALAJE.

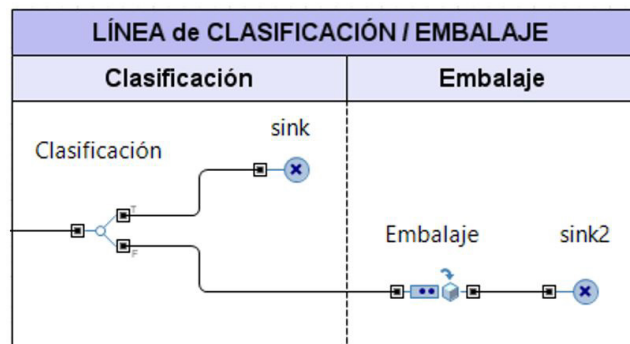


Figura 74. Línea de clasificación y embalaje.

Los *agentes* que continúan por el flujo de procesos, llegan a la Línea de clasificación y embalaje. Como se ha comentado anteriormente, en los requisitos de diseño no se especifica el dimensionamiento de la línea de clasificación y embalaje; y solo se llega a describir a nivel cualitativo. Por lo que se entiende que el

comportamiento de esta sección a nivel productivo es irrelevante, y que su papel solo es funcional.

Partiendo de esta premisa, el planar, el calibre y la máquina de visión artificial (ver punto 3.1.3.7 Clasificación y embalaje) que componen la línea de Clasificación, se han modelado mediante un bloque de modelado de procesos del tipo *SelectOutput* denominado "Clasificación". En este bloque, el 5% (NivelMerma) de los *agentes* pasan a un bloque *sink* denominado "sink" para ser eliminados del flujo de procesos (**Probability**, ver Figura 75). Ello modela la clasificación de piezas cerámicas que se haría a la salida del horno por parte de estos 3 equipos, y que eliminaría de la línea de producción las piezas consideradas defectuosas.

Por otro lado, el valor del parámetro "NivelMerma" corresponde al valor especificado en los requisitos de diseño (ver Tabla 6).



Figura 75. Definición del bloque de función Clasificación.

A continuación, los *agentes* siguen recorriendo el flujo de procesos, llegan a la etapa de embalaje. Como se ha comentado anteriormente, el cometido de esta etapa es meramente funcional, y el único requisito es que los *agente* queden empacados en grupos de 25 unidades (ver Punto 5.2 Caracterización de la distribución en planta de Azulejos S.A). Para modelar esta etapa del proceso productivo, se ha optado por utilizar un bloque de modelado de procesos del tipo *Batch* (ver Anexo II, Punto 2.19 Batch) denominado "Embalaje". De manera que como tamaño de lote (**Batch Size**, ver figura 76).

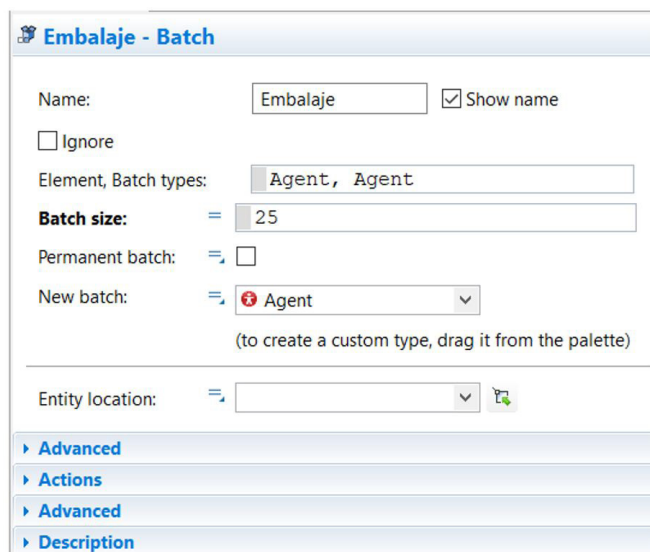


Figura 76. Definición del bloque de modelado de procesos Embalaje.

El nuevo *agente* generado es del tipo "Agent" (genérico). No se ha creído conveniente crear una nueva categoría de *agentes* para representar los lotes de producto acabado; ya que el bloque de modelado de procesos donde van a acabar estos agentes, es del tipo *Sink* (sink2 ver figura 74), lo que significa que serán purgados del sistema.

7.2. IMPLEMENTACIÓN DEL EXPERIMENTO DE OPTIMIZACIÓN EN EL MODELADO DE PROCESOS.

En los apartados anteriores se ha presentado y definido con el mayor grado de detalle posible las soluciones adoptadas en el modelo de simulación de una línea de fabricación industrial de azulejos cerámicos. Los parámetros que delimitan el funcionamiento de los elementos incluidos en dicha línea (velocidades, dimensiones y tasas) se han escogido de tal forma que reflejen de manera precisa las condiciones que se han establecido en el Capítulo 5. Requisitos del modelo de simulación.

En este punto, se procederá a desarrollar un experimento de optimización (ver Anexo I Experimento de optimización en Anylogic) que permita para maximizar la utilización del *ResourcePool* "LGV" (ver punto 7.1.3.3 Modelado de los LGV en Anylogic).

7.2.1. DISEÑO DEL INTERFAZ DE OPTIMIZACIÓN.

Con el fin de realizar el experimento propuesto, se ha confeccionado una ventana de optimización donde introducir los parámetros que caracterizan el proceso, realizar las iteraciones implícitas en el experimento, y donde poder mostrar los resultados que se han alcanzado. En primer lugar, se ha creado el experimento de optimización desde donde implementará dicha ventana (ver figura 77).

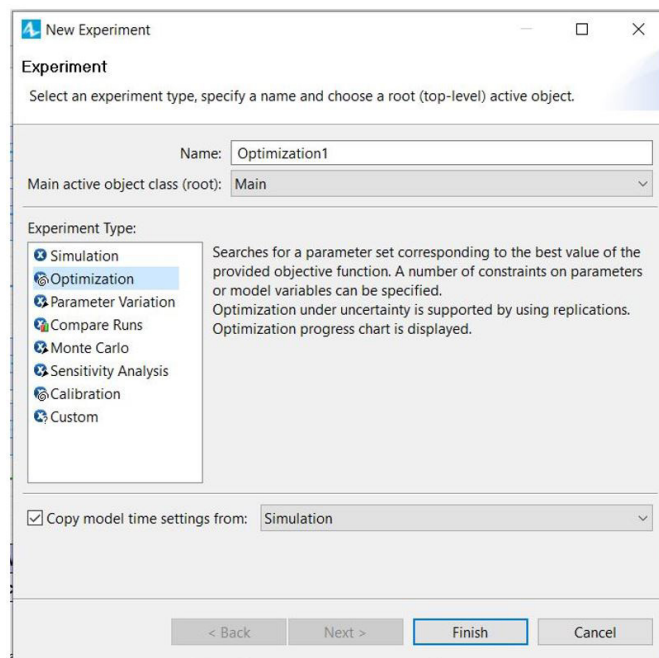


Figura 77. Ejemplo de creación del experimento de optimización.

Una vez se ha creado este experimento, aparece ya la ventana de optimización que se crea por defecto. Como se observa en la figura 78, el experimento carga automáticamente todos los parámetros del sistema junto su valor correspondiente.

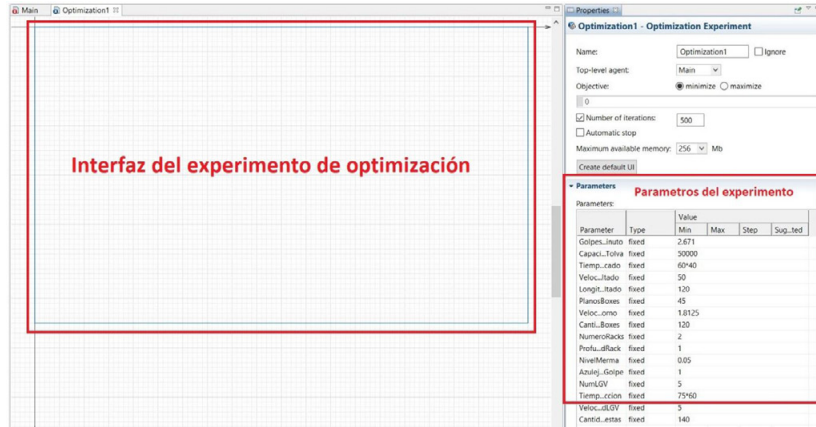


Figura 78. Ventana del experimento de optimización.

Llegados a este punto, se crea la interfaz de configuración del experimento (ver figura 80). En el lado izquierdo de la ventana, se halla una tabla recogiendo el conjunto de parámetros definidos en el sistema. Esta tabla contiene además los objetos que permitirán la visualización de dichos valores durante el experimento (los objetos con forma de interrogante). A modo de ejemplo, se muestra la estructura interna de uno de estos objetos en la figura 79.

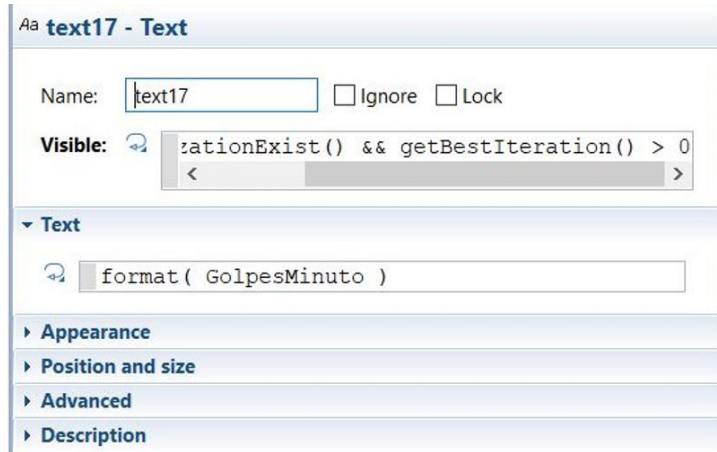


Figura 79. Objeto tipo "text" definido en el interfaz de optimización.

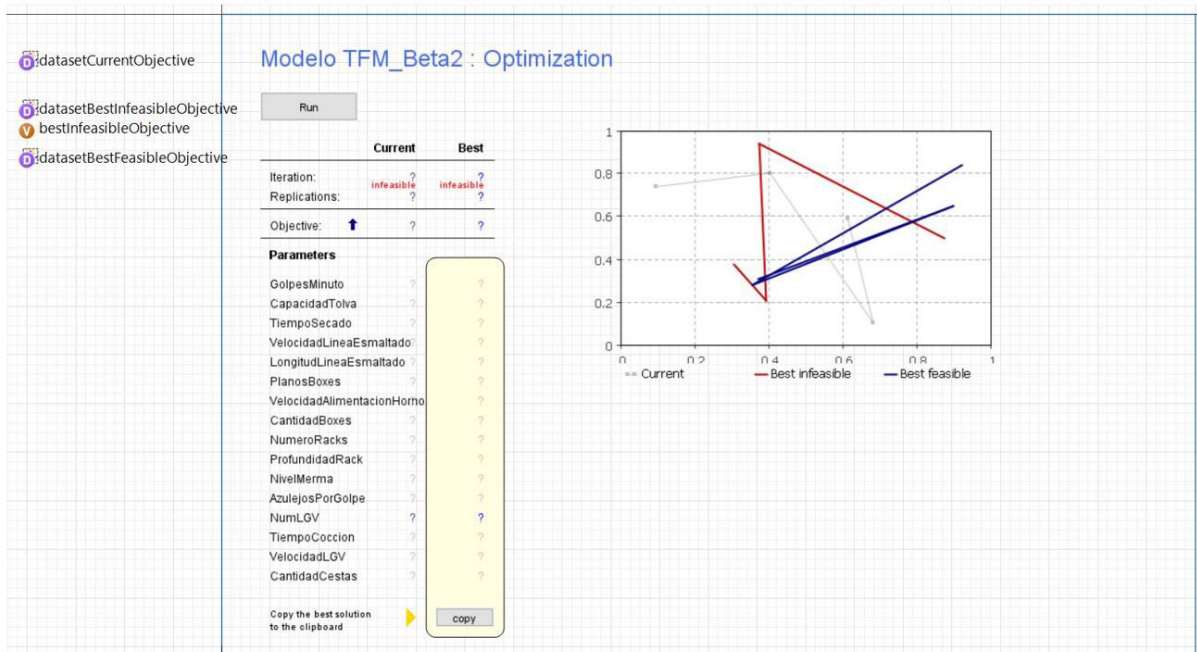


Figura 80. Interfaz del experimento de optimización por defecto.

Estos objetos se dividen en dos columnas distintas de la tabla, bajo los títulos de "Current" (del inglés, actual) y "Best" (del inglés, mejor). En la columna "Current" se muestran los valores de la iteración en curso, mientras que en la columna "Best" se da el valor de los parámetros pertenecientes a la iteración que mejor se ha ajustado a los criterios de optimización.

En el lado derecho de la ventana (ver figura 80), hay habilitado un gráfico para trazar la evolución de la función objetivo de la optimización en función de la iteración en curso. En el área del gráfico se trazan tres curvas diferentes:

- Una curva gris con el valor actual de la función objetivo.
- Una curva azul con el valor óptimo de la función objetivo para la iteración en curso.
- Una curva roja con el valor óptimo de la función objetivo no satisfaciendo los requerimientos de optimización impuestos por el usuario.

Vinculado a este gráfico, hay una tabla en la parte superior izquierda de la ventana. En esta tabla se representa el valor de la función objetivo en función de la iteración donde se ha calculado (Objective, ver figura 80); así como el número de iteraciones que sean llevado a cabo. De manera que, al finalizar el experimento, en la columna "Best" aparecerá la iteración que ha maximizado la función objetivo.

7.2.2. DEFINICIÓN DE LA FUNCIÓN OBJETIVO.

La optimización es llevada a cabo con el objetivo de maximizar la utilización del *ResourcePool* "LGV" definido en el modelo de simulación. Por lo que la función objetivo a maximizar que se ha utilizado es `root.LGV.utilization()` (Objective, ver figura 81).

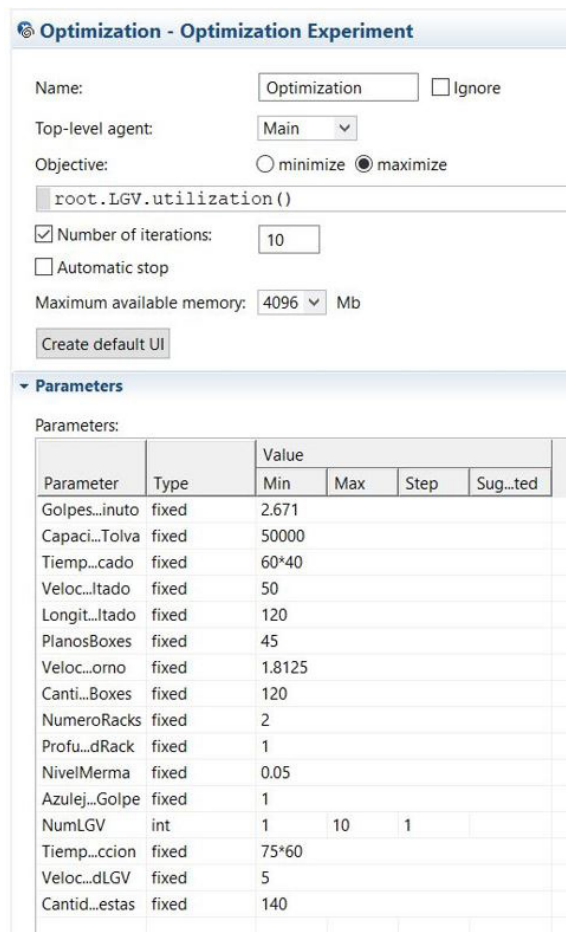


Figura 81. Ventana de propiedades del experimento de optimización.

Por otro lado, la capacidad de este *ResourcePool* se define a partir del parámetro "NumLGV", el cual define el número de *recursos* LGV disponibles (**Capacity** ver figura 49). Por lo que, para definir el experimento, esta variable se ha acotado entre los valores 1 (cantidad mínima para desarrollar la tarea de transporte) y 10 (cantidad de recursos muy sobredimensionada para realizar las tareas de transporte) (**Parameters**, ver figura 81). En consecuencia, la optimización se realiza en un proceso de 10 iteraciones (Number of iterations, ver figura 81), donde el valor de la variable cambia en una unidad por iteración (empezando con un único *recurso* en la iteración 1, hasta 10 *recursos* en la iteración 10).

Del mismo modo, la utilización del recurso LGV se acota dentro de un rango de valores. Esto permite estudiar los casos donde no se pueda disponer del 100% de la utilización del recurso (por ejemplo, los efectos de paradas de mantenimiento que se puedan dar en este tipo de vehículos). En un principio, se dejará por defecto en 1 (utilización del 100%).

Requirements

Requirements (are tested after a simulation run to determine whether the solution is feasible):

Enabled	Expression	Type	Bou...
<input checked="" type="checkbox"/>	root.LGV.utilization()	<=	1.0

Figura 82. Restricción implementada en el experimento de optimización.

7.3. EXPERIMENTO DE SIMULACIÓN DEL MODELO

En este capítulo se procederá a exponer el comportamiento del modelo de la línea de producción desarrollado en el capítulo anterior cuando este se encuentra en funcionamiento. Se describirá como avanzan los *agentes* a lo largo del flujo de procesos mientras avanza la simulación. De esta manera, se podrá observar cómo evoluciona el WIP de *agentes* en los distintos bloques de modelado de procesos que componen las distintas etapas del flujo de procesos.

Por otro lado, se describirá cual es el comportamiento de los *recursos* LGV durante el desarrollo de la simulación. De esta manera se podrá observar cuál es su dinámica junto con el resto de *agentes* que se encuentran dentro del modelo de simulación.

7.3.1. INICIO DE LA SIMULACIÓN DEL MODELO

En este punto describe el procedimiento necesario para iniciar el proceso de simulación. En primer lugar, para que el comportamiento de la simulación se ajuste al modelo de planta propuesto en el Capítulo 5 Requisitos del modelo de simulación, es necesario inicializar el valor de los parámetros de la simulación (ver figura 83). En la tabla 9 se recoge el valor que se ha asignado a los parámetros del modelo de simulación según este propósito.

Como se puede observar en la tabla 9, el valor inicial del parámetro que modela la velocidad de los LGV (VelocidadLGV) tiene asignado el valor de 5 (este valor, interpretado por el sistema, equivale a 5 m/s). Esto significa que el inicio de la simulación se realizará acorde al Caso 1 propuesto en el punto 5.2 Experimento de optimización. Por lo que este parámetro será el encargado de modificar las condiciones iniciales del experimento para poder simular los casos planteados en este punto.

Una vez se ha configurado el experimento ajustando los valores de los parámetros del modelo, desde los parámetros de la simulación, se determina la fecha de inicio de la simulación (ver figura 84). En este caso, como fecha propuesta, se ha planteado un lunes de una semana laboral cualquiera sin días festivos de por medio

(lunes 4/03/2019) a las 00:00:00 horas. Por último, se carga el experimento, y desde el interfaz del inicio de la simulación, se ejecuta la simulación (ver figura 85).

Parámetros de la simulación	
Nombre	Valor
CapacidadTolva	50.000
GolpesMinuto	2,671
AzulejosPorGolpe	1
TiempoSecado	60*40
CantidadCestas	140
VelocidadLineaEsmaltado	50
LongitudLineaEsmaltado	120
VelocidadAlimentacionHorno	1,8125
TiempoCocción	75*60
CantidadBoxes	120
NumeroRacks	2
NivelMerma	0,05
PlanosBoxes	45
NumLGV	5
VelocidadLGV	5

Tabla 9. Valor de los parámetros de la simulación.

DATOS

<p>Funciones</p> <ul style="list-style-type: none"> Cantidad <p>Horarios</p> <ul style="list-style-type: none"> PlanificaciónSemanal <p>Eventos</p> <ul style="list-style-type: none"> VoidSetUp VoidLoop <p>Rack System</p> <ul style="list-style-type: none"> ParqueBoxesVacios ParqueBoxesBizcocho <p>Otros</p> <ul style="list-style-type: none"> connections1 	<p>Recursos</p> <ul style="list-style-type: none"> PlanchaPrensa LGV <p>Variables</p> <ul style="list-style-type: none"> x1 y1 Pivote Flag1 Flag2 Stock <p>Agentes</p> <ul style="list-style-type: none"> Producto Boxes [...] 	<p style="text-align: center;">Parámetros</p> <ul style="list-style-type: none"> CapacidadTolva GolpesMinuto AzulejosPorGolpe TiempoSecado CantidadCestas VelocidadLineaEsmaltado LongitudLineaEsmaltado VelocidadAlimentacionHorno TiempoCoccion CantidadBoxes NumeroRacks ProfundidadRack NivelMerma PlanosBoxes
-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Figura 83. Parámetros del modelo de simulación.

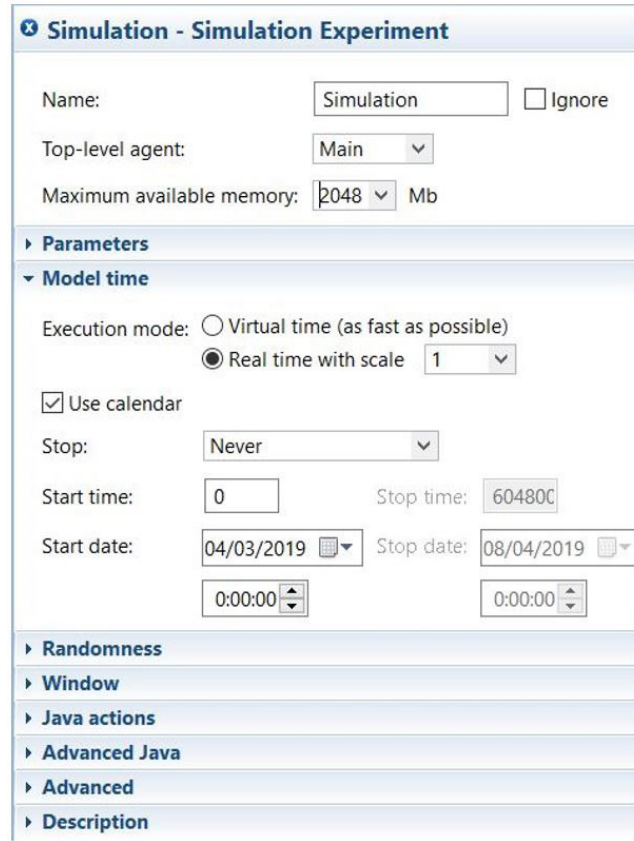


Figura 84. Fecha del inicio de la simulación.



Figura 85. Interfaz del inicio de la simulación.

7.3.2. COMPORTAMIENTO DEL MODELO DE SIMULACIÓN EN RÉGIMEN TRANSITORIO.

Una vez se ha iniciado la simulación, se observa que el bloque "source" ha inyectado el modelo de simulación 34.722 *agentes* y que estos han avanzado hasta la *queue* del bloque "Prensa" (ver figura 85).

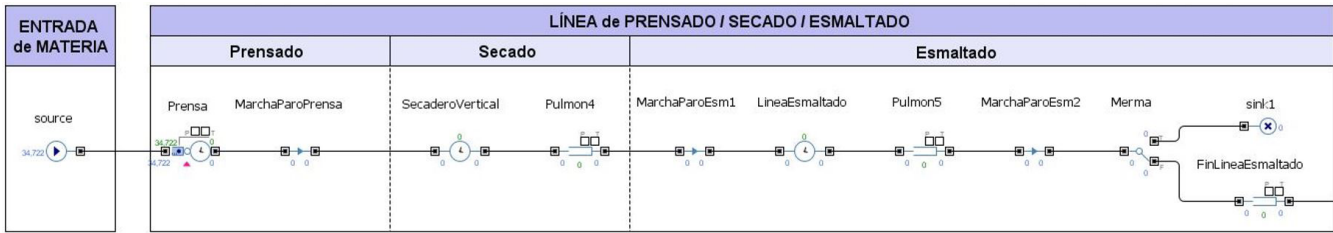


Figura 85. Proceso de llenado de la tolva de la prensa.

Por otro lado, si el lector se dirige al espacio continuo de la *Network* "ParqueDeBoxes" (ver figura 86), se observa que la población de *agentes* "Box" están colocados en sus respectivas posiciones de inicio dentro de los *PalletRack* que componen el *RackSystem* "ParqueBoxesVacios". También se observa que la población de *recursos* "LGV" aparece ubicada en el "node 2" de la *Network* (ver figura 50). Puesto que no se ha planteado ninguna ubicación específica para estos *recursos* basada en *Attractors*, su ubicación dentro del *node* se determina de manera aleatoria.

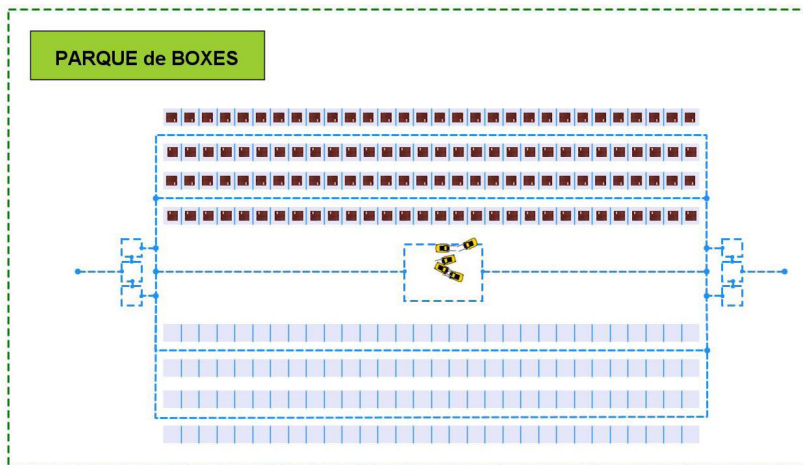


Figura 86. Espacio continuo de la Network al inicio de la simulación.

Si se sigue escrutando el espacio continuo de la *Network* "ParqueDeBoxes", se observa que al inicio de la simulación dos *recursos* "LGV" se desplazan al *RackSystem* "ParqueBoxesVacios" para transportar dos *agentes* "Box" al Área de Picking del final de la línea de esmaltado (ver figura 87).

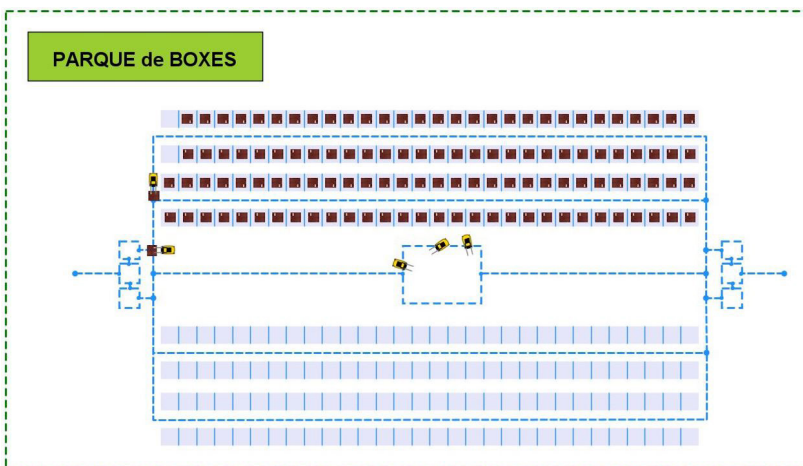


Figura 87. Dos recursos "LGV" transportando 2 agentes "Box" al Área de Picking del final de la Línea de Esmaltado.

En el momento en el que estos dos *agentes* han sido depositados en sus respectivos *nodes*, los *recursos* "LGV" vuelven a su localización inicial (ver figura 88). En ese momento, si el lector dirige la mirada al flujo de procesos del Área de Picking del Final de la Línea es Esmaltado, podrá observar que hay posicionado un *agente* "Box" en el bloque "BoxEnEspera" y otro en el bloque "BoxEnCarga".

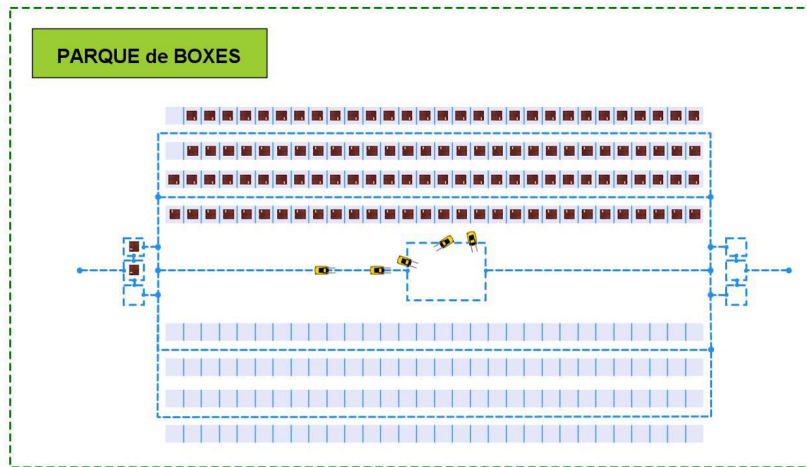


Figura 88 Recursos "LGV" volviendo a su posición inicial dentro de la Network.



Figura 89. Estado de los bloques de de modelado de procesos "BoxEnEspera" y "BoxEnCarga" al inicio de la simulación.

7.3.3. COMPORTAMIENTO DEL MODELO DE SIMULACIÓN EN RÉGIMEN PERMANENTE (PERIODO DE ENTRE SEMANA).

En este punto se mostrará cual es el comportamiento de los bloques de modelado de procesos que componen el flujo de procesos cuando trabajan en régimen permanente. El comportamiento de los bloques vendrá determinado por el WIP que acumulan durante el desarrollo de la simulación. Estos WIP debe ser equivalentes a las medidas de flujo básicas calculadas anteriormente (ver punto 3.5.1 Medidas de flujo básicas de las operaciones de producción), pero la verificación de los resultados obtenidos en este punto, se analizarán en el capítulo siguiente.

Por otro lado, también se mostrará cual es el comportamiento dinámico de los elementos que componen el espacio continuo de la *Network* "ParqueDeBoxes", para observar cual ha sido el resultado de la estrategia que se ha seguido para modelar el comportamiento de la planta de Azulejos S.A.

- LÍNEA DE PRENSADO-SECADO-ESMALTADO.

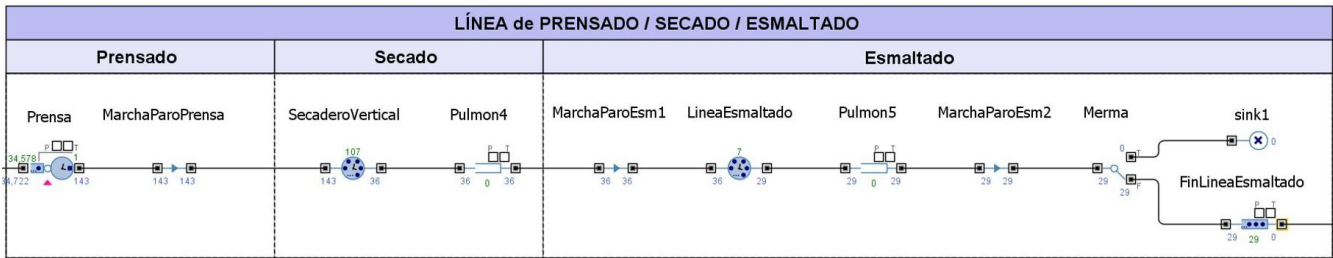


Figura 90. Estado de una simulación en t = 04-03-2019 00-53-37 AM (Línea de Prensado-Secado-Esmaltado).

Continuando con el desarrollo de la simulación, se observa que los *agentes* producto avanzan sin problemas a lo largo del flujo de procesos. Si se detiene la simulación en alrededor de t = 04/03/2019 53:35 AM (ver figura 90), el flujo de procesos muestra que los WIP de los bloques "SecaderoVertical" y "LineaDeEsmaltado" se han estabilizado en 107 y 7 *agentes* respectivamente. Por otro lado, también se observa que el número de *agentes* embebidos en el bloque "FinLineaEsmaltado" está incrementándose.

- ÁREA DE PICKING DEL FINAL DE LÍNEA DE ESMALTADO.

Si se detiene la simulación en alrededor de t = 04/03/2019 1:00:22 AM (ver figura 90) se observa que 45 *agentes* "producto" que se encontraban en el bloque "FinLineaEsmaltado" han avanzado al bloque "pickup"; y que el *agente* "Box" que se encontraba en el bloque "BoxEnCarga" ha avanzado hasta "AlmacenarBoxBizcocho". Si llegados a este punto se presta atención, se llega a la conclusión de que en el bloque "AlmacenarBoxBizcocho" se encuentra un *agente* "Box" con 45 *agentes* "Producto" embebidos dentro de sí.

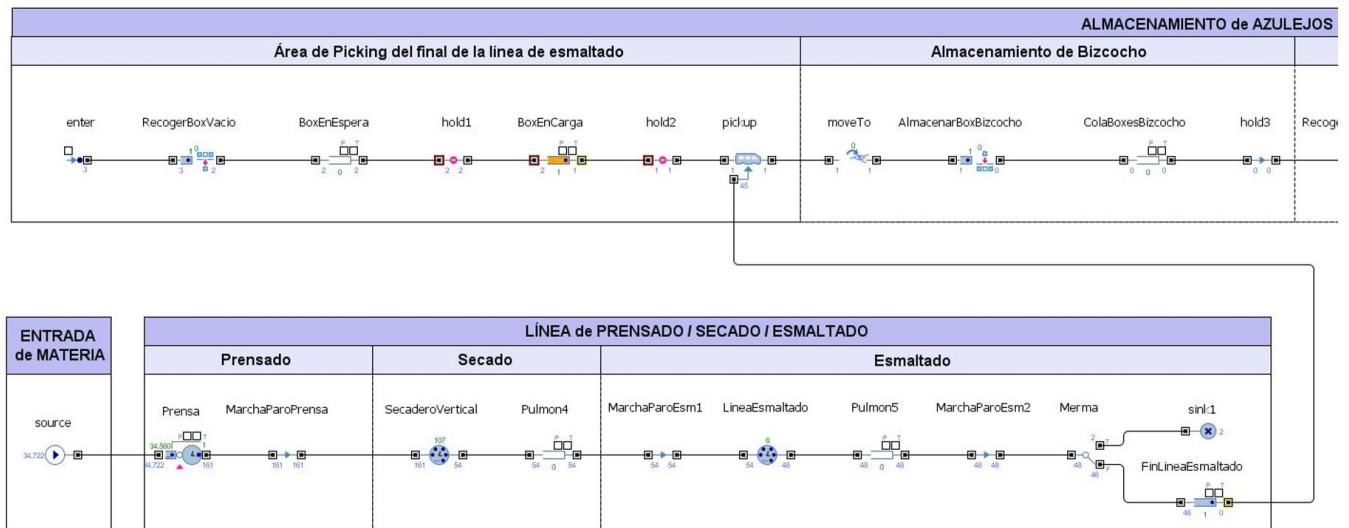


Figura 91. Estado de una simulación en t = 04-03-2019 1-00-22 AM.

Por otro lado, también se observa que el *agente* que se encontraba en el bloque "BoxEnEspera" ha avanzado al bloque "BoxEnCarga". Por lo que, si se dirige la mirada al espacio continuo de la *Network* "ParqueDeBoxes" (ver figura 92), se observa que los *agentes* Box que se encuentran en el Área de Picking del final de la línea de esmaltado, también han avanzado de posición dentro de la *Network*.

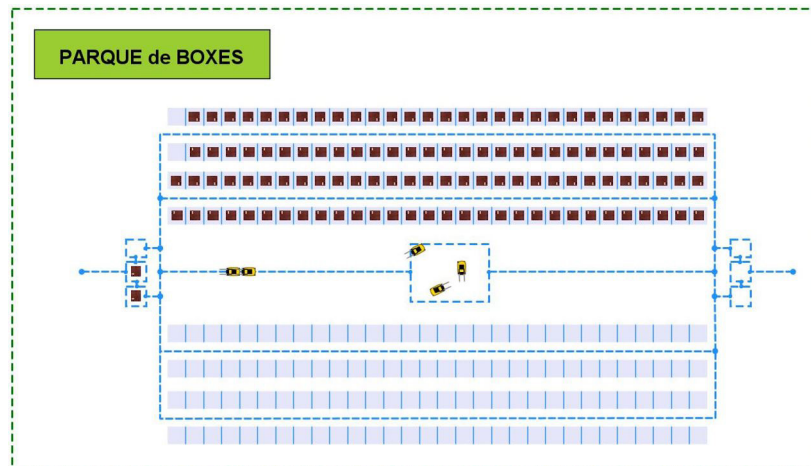


Figura 92 Estado de una simulación en t = 04-03-2019 1-01-12 AM (Espacio continuo de la Network).

- ALMACENAMIENTO DE BIZCOCHO.

En ese momento dos *recursos* "LGV" se han puesto en movimiento (ver figura 92). Uno de esos "LGV" se dirige Área del Final de la línea de esmaltado, su intención es transportar al *agente* "Box" que contiene los *agentes* "Producto" embebidos al *RackSystem* "ParqueBoxesBizcocho". El otro por su parte, se dirige al *RackSystem* "ParqueBoxesVacios" con intención de transportar un *agente* "Box" al Área de Picking del final de la línea de esmaltado.

Más adelante, cuando el *agente* "Box" ha sido depositado en el *RackSystem* "ParqueBoxesBizcocho", un tercer "LGV" sale a su encuentro con intención de transportarlo al Área de Picking del inicio de la línea de cocción (ver figura 93).

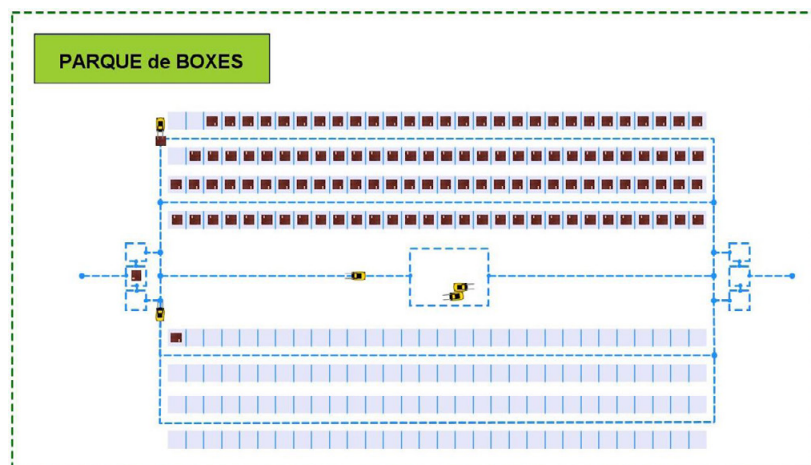


Figura 93. Estado de una simulación en t = 04-03-2019 1-02-12 AM (Espacio continuo de la Network).

Por último, el "Box" es depositado en el Área de Picking del inicio de la línea de cocción y todos los "LGV" vuelven a su posición de inicio dentro de la *Network* (ver figura 94).

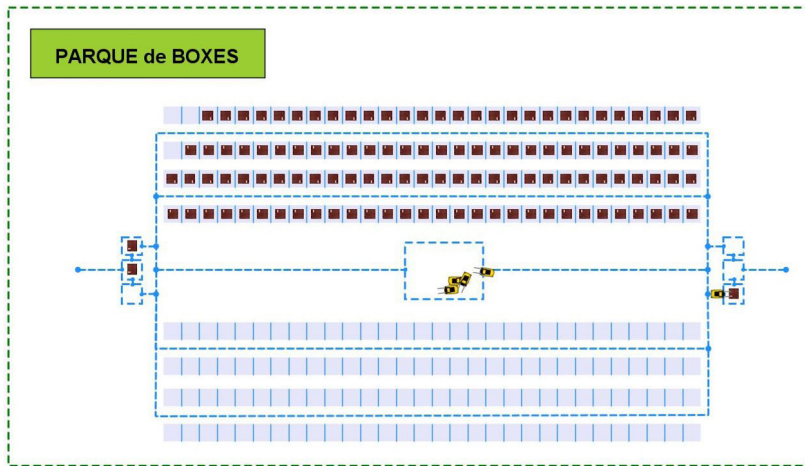


Figura 94. Estado de una simulación en t = 04-03-2019 1-05-16 AM (Espacio continuo de la Network).

- ÁREA DE PICKING DEL INICIO DE LA LÍNEA DE COCCIÓN I.

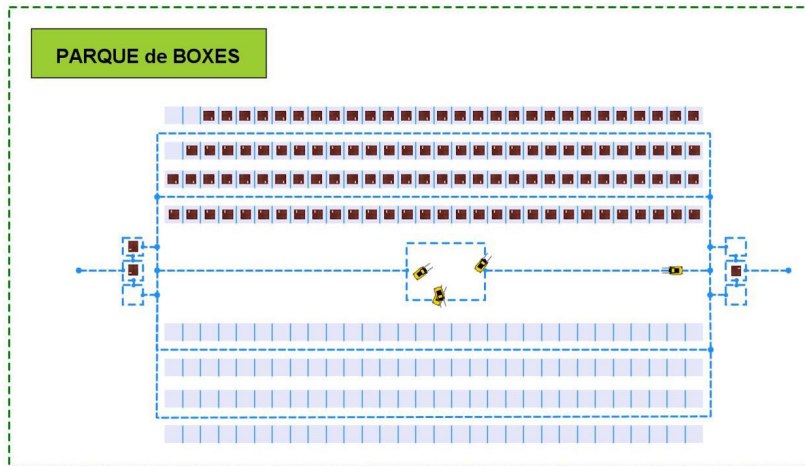


Figura 95. Estado de la simulación en t = 04-03-2019 1-06-29 AM (Espacio continuo de la Network).

En el instante en el que el agente "Box" ha sido depositado en el *node* "H_Recepcion", este automáticamente avanza de posición hasta el *node* "H_EnDescarga" (ver Figura 95). Si en ese instante el lector dirige la mirada al entorno del flujo de procesos, podrá observar que en ese momento hay un agente en el bloque "BoxEnDescarga" (ver Figura 96).

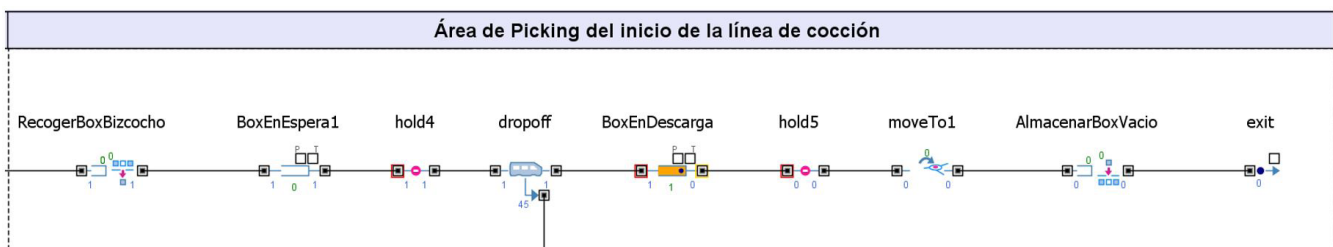


Figura 96. Estado de la simulación en t = 04-03-2019 1-06-29 AM.

Si se continúa con la simulación se observa que, en el instante en el que hay disponible un agente en el RackSystem "ParqueBoxesBizcocho", un recurso "LGV" sale a su encuentro para transportarlo al Área de Picking

del inicio de la línea de cocción (ver Figura 97). Una vez está allí, deposita el agente de nuevo en el *node* "H_Recepcion" (ver Figura 98).

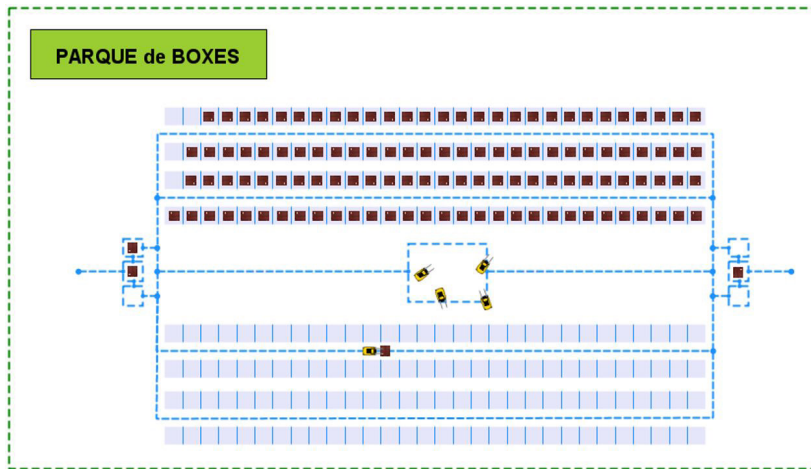


Figura 97. Estado de la simulación en t = 04-03-2019 1-22-56 AM (Espacio continuo de la Network).

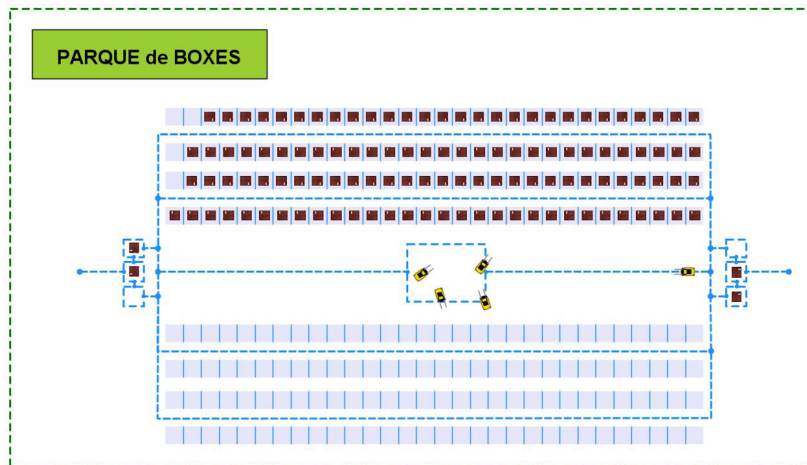


Figura 98. Estado de la simulación en t = 04-03-2019 1-24-35 AM (Espacio continuo de la Network).

Si en ese instante el lector dirige la mirada al entorno del flujo de procesos, podrá observar que en ese momento hay un agente en el bloque "BoxEnEspera1" y otro en el bloque "BoxEnDescarga" (ver Figura 99).

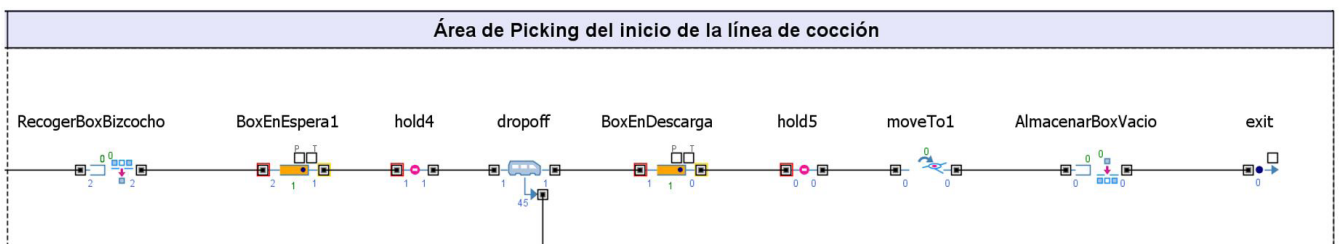


Figura 99. Estado de la simulación en t = 04-03-2019 1-24-35 AM.

- LÍNEA DE COCCIÓN.

Si se retoma la simulación en $t = 04/03/2019\ 1:06:29\ AM$ y se dirige la mirada a la Línea de cocción, se puede observar que los *agentes* "Producto" que se hallaban embebidos en el *agente* "Box", mediante el bloque "dropoff" (ver Figura 96), se han desagregado y han avanzado hasta el bloque "InicioHorno" (ver Figura 100). Puesto que esta etapa tiene un $CT=0$, un *agente* ya ha avanzado al bloque "DescargaBizcocho" de manera instantánea.

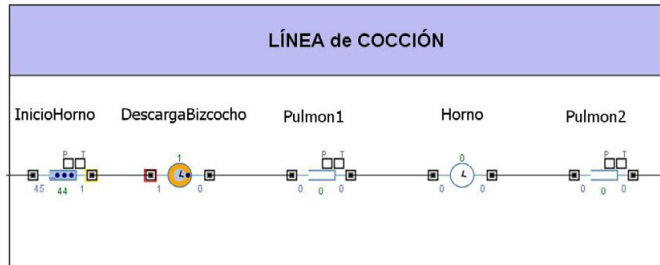


Figura 100. Estado de la simulación en $t = 04-03-2019\ 1-06-29\ AM$.

Continuando con el desarrollo de la simulación, se observa que en $t = 04/03/2019\ 2:21:46\ AM$, se observa que el WIP del bloque "Horno" se ha estabilizado en 136 *agentes*, y que un *agente* ya ha avanzado hacia la siguiente etapa del flujo de procesos (ver Figura 101).

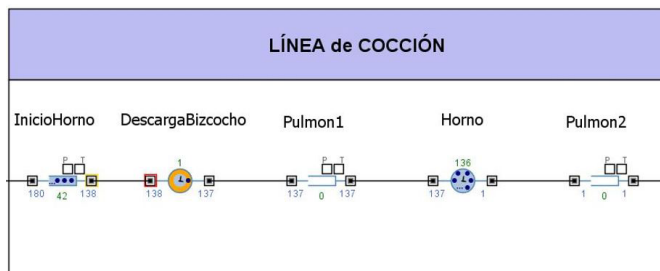


Figura 101. Estado de la simulación en $t = 04-03-2019\ 2-21-46\ AM$.

- LÍNEA DE CLASIFICACIÓN Y EMBALAJE.

Si en $t = 2:21:46$ el lector dirige la mirada a la Línea de Clasificación y Embalaje (ver Figura 102), se observa que el *agente* ha abandonado la etapa de la Línea de cocción, ha atravesado la sección de clasificación y se encuentra en el bloque "Embalaje" a la espera de más unidades para completar un bloque.

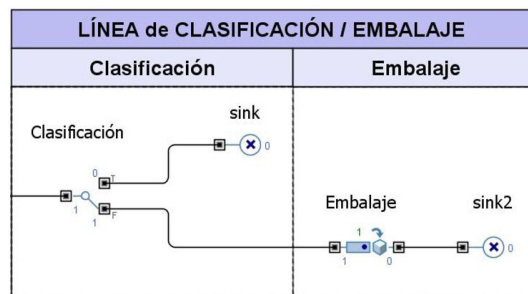


Figura 102. Estado de la simulación en $t = 04-03-2019\ 2-21-46\ AM$.

Cuando la simulación llega a $t = 04/03/2019\ 2:36:16\ AM$ se observa que el bloque "Embalaje" ha generado un nuevo *agente* del tipo "Agent" basado en 25 *agentes* "Producto" embebidos. A continuación, este nuevo

agente ha avanzado al bloque "Sink2" y ha sido purgado del sistema. Por otro lado, si se pone el punto de mira sobre el comportamiento de la sección de clasificación, se observa que el bloque "Clasificación" ha purgado del sistema 2 agentes de los 27 que han pasado a través de el Llegados a este momento (ver Figura 103).

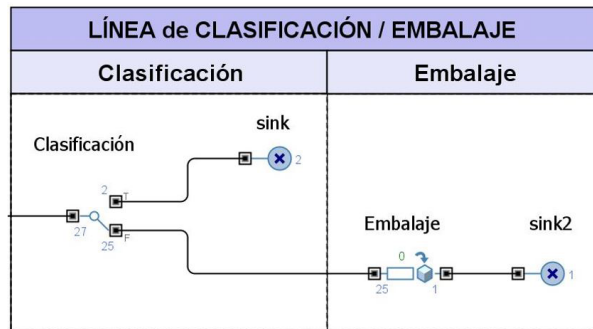


Figura 103. Estado de la simulación en t = 04-03-2019 2-36-16 AM.

- ÁREA DE PICKING DEL INICIO DE LA LÍNEA DE COCCIÓN II.

Si se retoma la simulación en t = 04/03/2019 1:31:37 AM y se dirige la mirada al Área de Picking del inicio de la línea de cocción (ver Figura 101), se observa que el agente "Box" que se encontraba en el bloque "BoxEnDescarga" ha avanzado hasta el Bloque "AlmacenarBoxVacío". Esto se debe a que en el bloque "InicioHorno" se han consumido todos los agentes "Producto" que estaban almacenados. Como resultado, el agente "Box" que se encontraba en el node "H_EnDescarga" ha avanzado al node "H_EnPicking" y el agente que se encontraba en "H_EnEspera" ha avanzado a "H_EnDescarga" (ver Figura 104).

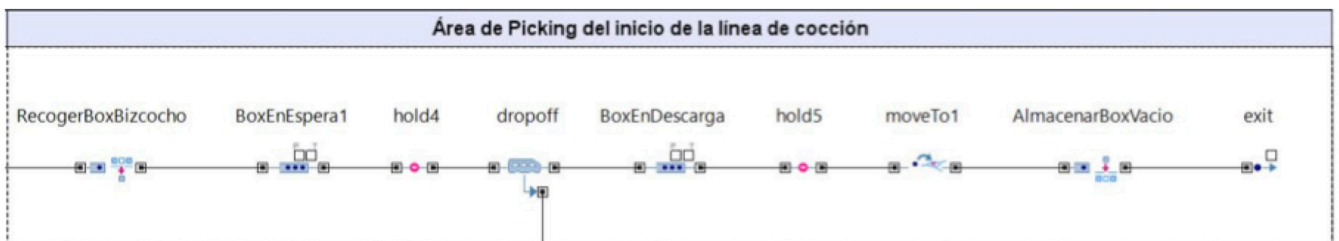


Figura 104. Estado de la simulación en t = 04-03-2019 1-31-37 AM .

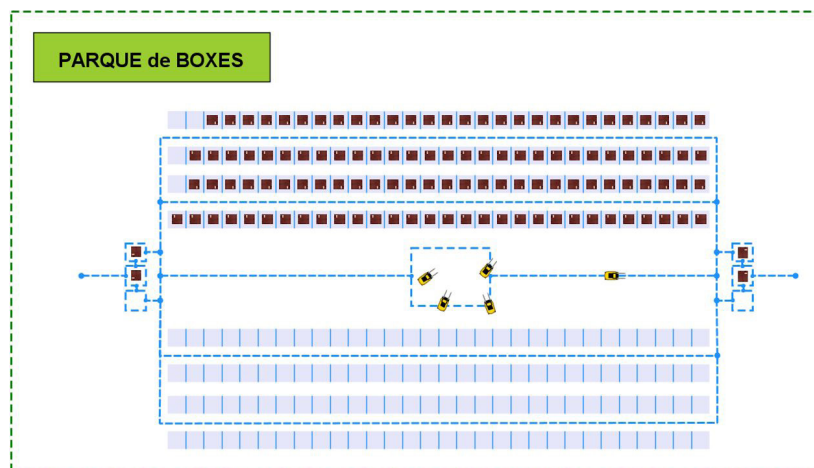


Figura 105. Estado de la simulación en t = 04-03-2019 1-31-37 AM (Espacio continuo de la Network).

En ese mismo instante, un *recurso* LGV ha salido a su encuentro con intención de transportarlo al *RackSystem* "ParqueBoxesVacios" (ver figura 105). Si se avanza la simulación en tiempo $t = 04/03/2019$ 1:34:39 AM, Se observa que el "LGV" ha depositado al *agente* "Box" en *RackSystem* "ParqueBoxesVacios" y que vuelve a su posición de origen dentro de la *Network* (ver Figura 106).

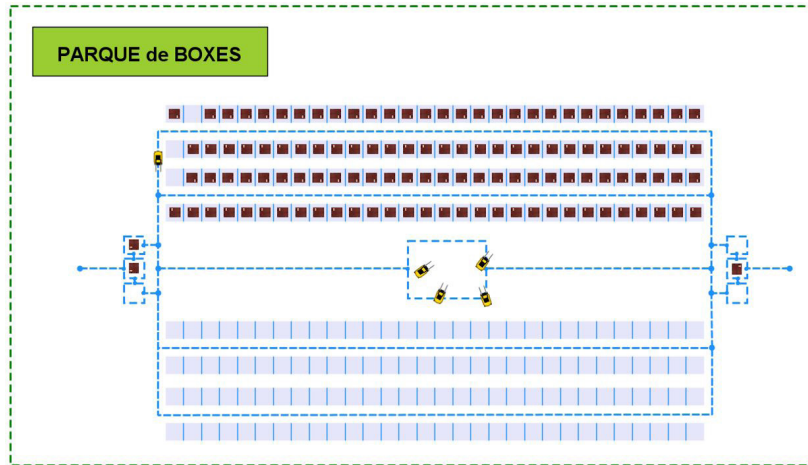


Figura 106. Estado de la simulación en $t = 04-03-2019$ 1-34-39 AM (Espacio continuo de la Network).

7.3.4. COMPORTAMIENTO DEL MODELO DE SIMULACIÓN EN RÉGIMEN PERMANENTE (PERIODO DE FIN DE SEMANA)

Si se avanza la simulación a $t = 09/03/2019$ 00:03:41 AM se observa que el *RackSystem* "ParqueBoxesBizcocho" está prácticamente lleno, y el *RackSystem* "ParqueBoxesVacios" está prácticamente vacío (ver Figura 107).

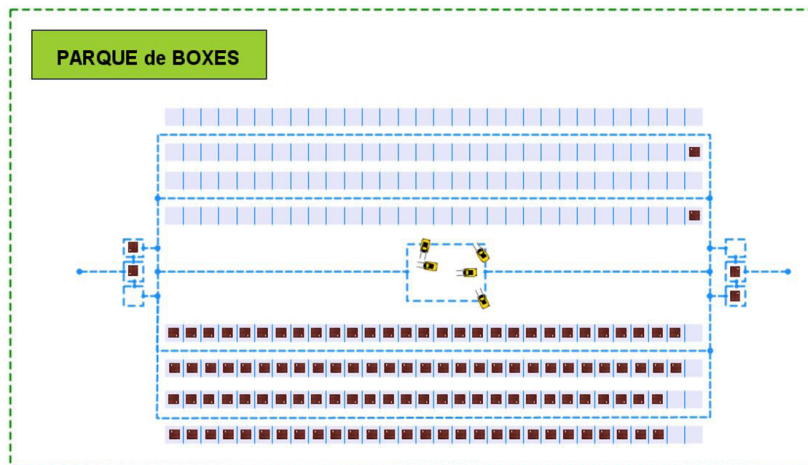


Figura 107. Estado de la simulación en $t = 09-03-2019$ 00-02-41 AM (Espacio continuo de la Network).

Por otro lado, si se continua con el desarrollo de la simulación y se llega a $t = 11/03/2019$ 00:00:34 AM se observa que el *RackSystem* "ParqueBoxesBizcocho" está completamente vacío y que solamente quedan *agentes* en el Área de Picking del inicio de la Línea de cocción dando alimentación a la Línea de cocción (ver Figura 108).

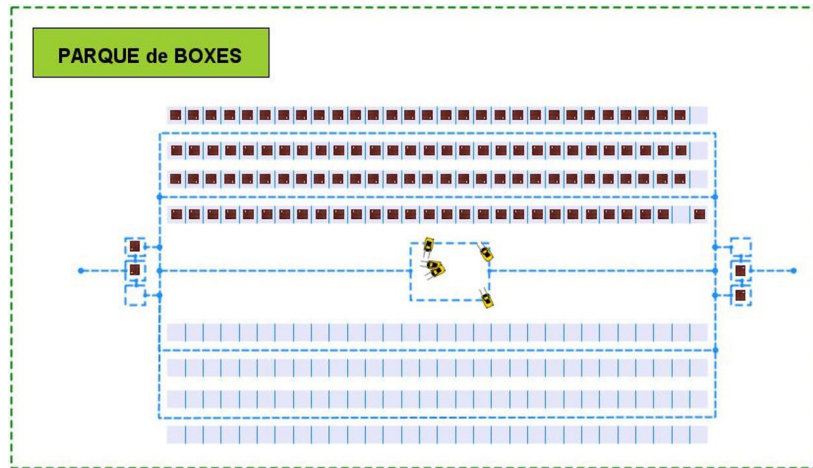


Figura 108. Estado de la simulación en lunes, t = 11-03-2019 00-00-34 AM (Espacio continuo de la Network).

7.3.5. FINALIZACIÓN DE LA SIMULACIÓN.

La simulación finaliza en t = 16/03/2019 05:31:22 PM. Si se escruta el entorno del flujo de procesos de la simulación, se observa que todos los WIP de los bloques de modelado de procesos están a 0; salvo un pequeño remanente en "FinLineaEsmaltado" y "Embalaje" (ver Figura 109). Por otro lado, el bloque "sink2" da cuenta de los lotes de *agentes* que se han producido a lo largo de la simulación, 1252.

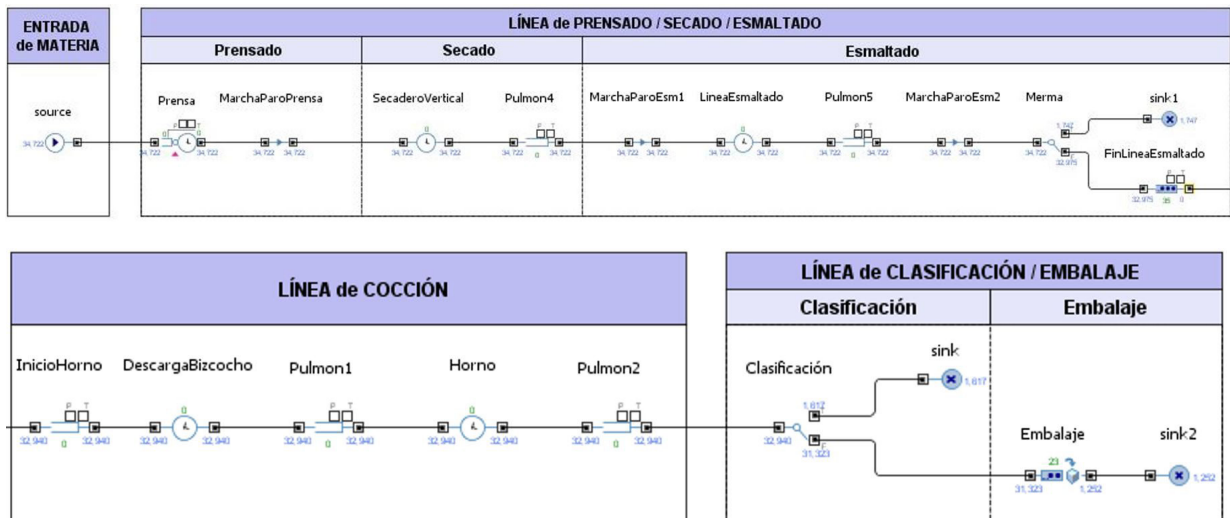


Figura 109. Estado de los bloques de modelado de procesos finalizada la simulación.

Por otro lado, si el lector dirige la mirada al espacio continuo de la *Network*, podrá observar que la posición de los *agentes* "Box" es la misma que se da cuando finaliza el régimen transitorio del inicio de la simulación (ver Figura 110).

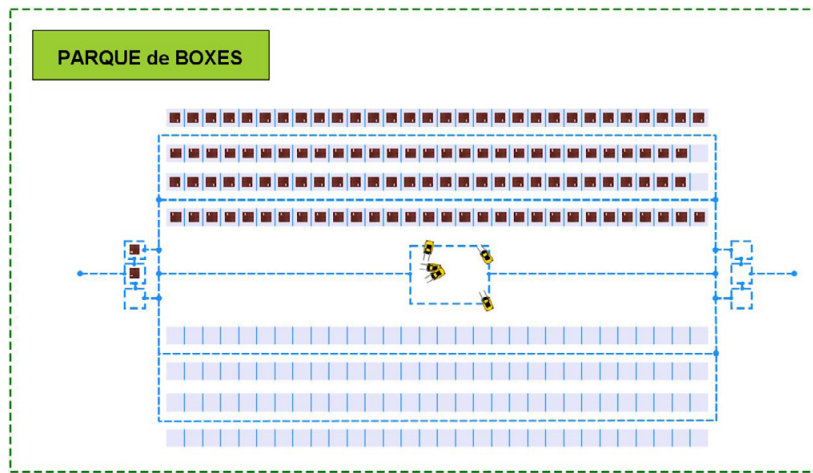


Figura 110. Estado del espacio continuo de la Network finalizada la simulación.

7.3.6. EXPERIMENTO DE OPTIMIZACIÓN

En este punto se procederá ejecutar el experimento de optimización planteado el Punto 6.3. Implementación del experimento de optimización en el modelado de procesos. En este experimento se desarrollarán los tres casos planteados en el Punto 5.2.1 Condiciones iniciales del experimento de optimización. Una vez se haya ejecutado cada caso, se reportarán los datos y la información que arroja el experimento.

- CASO 1: VELOCIDAD DE LOS LGV 5 M/S.

Puesto que en la simulación desarrollada en los puntos anteriores el valor del parámetro “VelocidadLGV” tenía un valor asignado de 5, no es necesario inicializar la simulación modificando el valor de este parámetro (ver Punto 7.3.1 Inicio de la simulación del modelo). Por lo que, en este caso, ya se puede dar paso al inicio de la simulación sin cambiar nada (ver Figura 111).

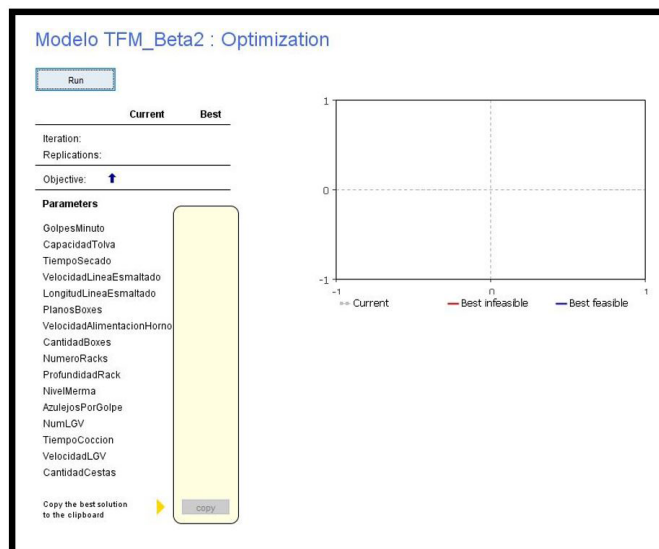


Figura 111. Experimento del Caso 1 inicializado.

Una vez que se ha ejecutado el experimento de optimización el interfaz arroja los resultados que se han obtenido. Para las condiciones iniciales del experimento dadas a partir de los parámetros con los que se ha inicializado la simulación y para una velocidad de 5 m/s, la cantidad de *agentes* “LGV” que maximizan la utilización de este recurso es de 2 (ver Figura 112). Por otro lado, se observa que la máxima utilización que se puede alcanzar con estos dos recursos es 1, lo que significa que su utilización es máxima.

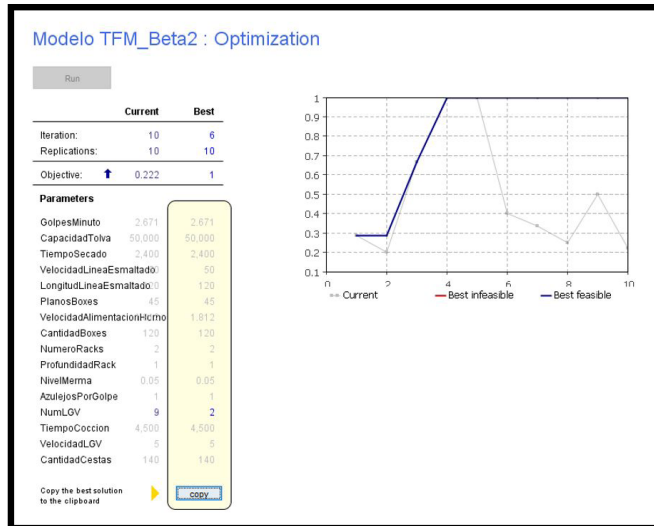


Figura 112. Resultados del experimento de optimización del Caso 1.

- CASO 2: VELOCIDAD DE LOS LGV 7,5 M/S.

Para poder ejecutar el experimento de optimización planteando este caso, se ha modificado el valor del parámetro "VelocidadLGV" para asignarle el valor de 7,5 y a continuación se ha ejecutado el experimento.

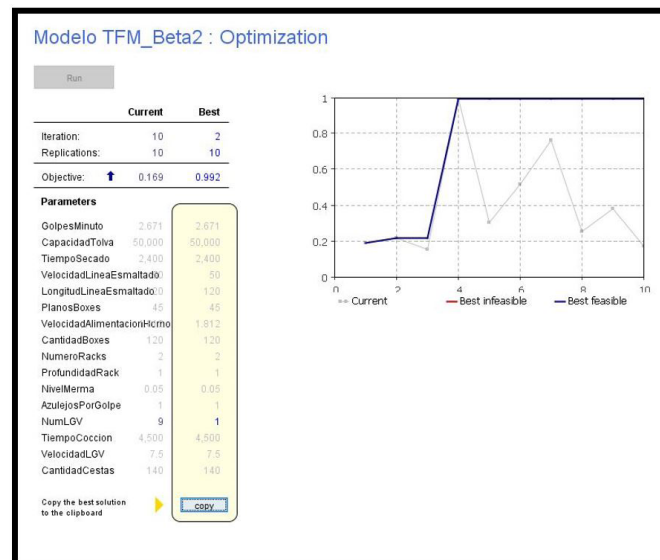


Figura 113 Resultados del experimento de optimización del Caso 2.

En este caso, para las condiciones iniciales del experimento dadas a partir de los parámetros con los que se ha inicializado la simulación y para una velocidad de 7,5 m/s, la cantidad de agentes "LGV" que maximizan la utilización de este recurso es de 1 (ver Figura 112). Por otro lado, se observa que la máxima utilización que se puede alcanzar con este único recurso es 1, lo que significa que su utilización es máxima.

- CASO 3: VELOCIDAD DE LOS LGV 5 M/S CONSIDERANDO PARADAS POR RECARGA DE BATERÍAS.

Para poder ejecutar el experimento de optimización planteando este caso, se ha modificado el valor del parámetro "VelocidadLGV" para asignarle el valor de 5. Por otro lado, se ha limitado la utilización máxima acotando el valor de la restricción implementada en el experimento de optimización en 0.8333 (ver figura 113). Una vez se han implementado estos cambios, se ha ejecutado el experimento de optimización.

Requirements

Requirements (are tested after a simulation run to determine whether the solution is feasible):

Enabled	Expression	Type	Bound
<input checked="" type="checkbox"/>	root.LGV.utilization()	<=	0.8333

Figura 113. Restricción implementada en el experimento de optimización del Caso 3.

En este caso, para las condiciones iniciales del experimento dadas a partir de los parámetros con los que se ha inicializado la simulación, para una velocidad de 5 m/s y limitando la utilización máxima en 0.8333; la cantidad de *agentes* "LGV" que maximizan la utilización de este recurso es de 3 (ver Figura 114). Por otro lado, se observa que la máxima utilización que se puede alcanzar con estos tres recursos es 0,667.

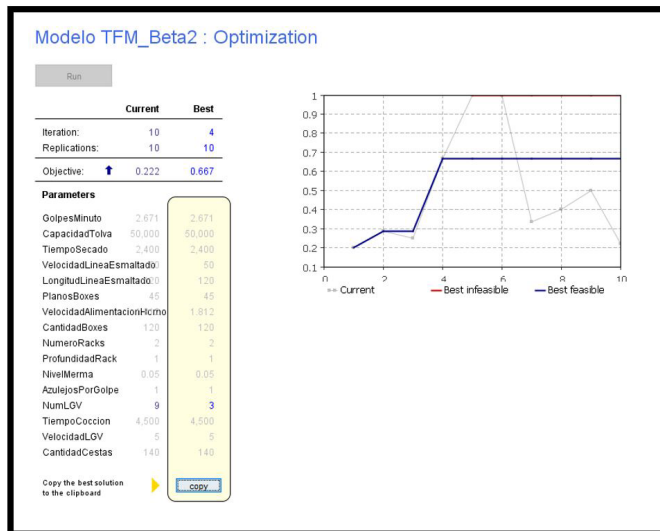


Figura 114. Resultados del experimento de optimización del Caso 3.

8. VERIFICACIÓN DE RESULTADOS

En este capítulo se procederá a analizar los resultados obtenidos en el capítulo anterior sobre el comportamiento del modelo de simulación. En primer lugar, se analizará cual ha sido el desempeño del modelo de simulación para sacar conclusiones sobre la cohesión de los distintos actores en el entorno multiparadigma. A continuación, se estudiarán las medidas de WIP arrojadas por los distintos bloques de modelado de procesos durante el desarrollo de la simulación en régimen estacionario. Y por último se sacarán conclusiones en relación a los casos estudiados en el experimento de simulación.

8.1. DESARROLLO DEL EXPERIMENTO DE SIMULACIÓN DEL MODELO.

Una vez se ha iniciado la simulación, se observa que el bloque de modelado de procesos "source" ha inyectado el modelo de simulación 34.722 *agentes* y que estos han avanzado hasta la *queue* del bloque "Prensa" (ver figura 85). Teniendo en cuenta que el tamaño de pedido son 50.000 metros cuadrados y que el formato de las piezas es de 120x120 cm, el número de *agentes* inyectados en el modelo de simulación concuerda con las condiciones iniciales planteadas (ver Capítulo 3 Antecedentes). Por otro lado, se observa que la población de *agentes* "Box" están colocados en sus respectivas posiciones del espacio continuo de la *Network*, y que dos *recursos* "LGV" han transportado dos *agentes* "Box" a sus respectivas posiciones del Área de Picking del final de la Línea de Esmaltado (ver figura 87). Después de analizar este comportamiento, se puede afirmar que es correcto y que concuerda perfectamente con el planteamiento realizado en el 7.1.3.4. Inicialización del Almacén intermedio.

Más adelante, cuando el sistema alcanza el régimen permanente, se observa que los *agentes* "Producto" avanzan sin problemas a lo largo del flujo de procesos siguiendo el enrutamiento propuesto a lo largo del Capítulo 7 Solución adoptada. Por otro lado, también se observa que el movimiento de los *agentes* "Box" concuerda con el planteamiento desarrollado en los Puntos 7.1.3. Almacenamiento intermedio de los azulejos (I), 7.1.4. Almacenamiento intermedio de los azulejos (II) y 7.1.5. Almacenamiento intermedio de los azulejos (III). Esto significa que, a nivel dinámico, el modelo de simulación se ha comportado acorde al planteamiento de diseño, por lo que se puede afirmar que su funcionamiento en régimen permanente es correcto y que no presenta fallos.

Por último, cuando la simulación finaliza, se observa que todos los WIP de los bloques de modelado de procesos están a 0, salvo un pequeño remanente en los bloques "FinLineaEsmaltado" y "Embalaje" (ver Figura 109). Ello es indicativo de que el sistema ha consumido todos los *agentes* producto que han sido inyectados al inicio de la simulación, y que no han quedado *agentes* por procesar. También se observa que todos los *agentes* "Box" quedan almacenados en el *RackSystem* "ParqueBoxesVacíos" y que hay dos que siguen posicionados en el Área de Picking del Final de la Línea de Esmaltado. Analizando la situación final de estos *agentes*, se puede afirmar que una vez se inicia la simulación, el modelo siempre se mantiene en régimen permanente; incluso hasta cuando está ya ha terminado, y no hay más *agentes* que procesar. Esto es un indicativo de la robustez que presenta el modelo; pues si una vez finalizada la simulación se volviera a inyectar en el modelo un conjunto adicional de *agentes*, este se pondría a trabajar de nuevo sin ningún problema.

8.1.1. DESARROLLO DE LA SIMULACIÓN DURANTE EL PERIODO DE FIN DE SEMANA

Si se observa cual es el comportamiento del espacio continuo de la *Network* cuando se alcanza $t = 09/03/2019\ 00:02:41\ AM$ (ver Figura 107), se aprecia que todos los *agentes* "Box" menos 2 se encuentran en el *RackSystem* "ParqueBoxesBizcocho". Ello se debe a que, aunque el bloque "Prensa" haya parado, aún hay que procesar el WIP acumulado en el bloque "SecaderoVertical" y "LineaEsmaltado". Si se tiene en cuenta que en el momento de que el bloque "Prensa" deja de trabajar en estos dos bloques hay un WIP acumulado de entre 112 y 114 *agentes*, y que la capacidad de los *agentes* "Box" es de 45 unidades, con este remanente de *agentes* acumulado aún se llenan estos dos últimos "Box" que quedan.

Por otro lado, si se continua con el desarrollo de la simulación y se llega a $t = 11/03/2019\ 00:00:34\ AM$ se observa que el *RackSystem* "ParqueBoxesBizcocho" está completamente vacío y que solamente quedan *agentes* en el Área de Picking del inicio de la Línea de cocción dando alimentación a la Línea de cocción (ver Figura 108).

Viendo esta situación, el lector puede llegar a la conclusión de que bloque "Horno" se puede quedar en algún momento sin suministro mientras se espera a los *agentes* que salen del bloque "Prensa" lleguen al almacén intermedio. Pero en esta situación, el bloque "Horno" aún tiene un *agentes* "Box" en descarga y otro en espera, lo que le permite seguir alimentándose de *agentes* mientras el *RackSystem* "ParqueBoxesBizcocho" se mantiene a la espera de recibir nuevos *agentes* "Box". Si se continúa con la simulación, se observa que durante un breve periodo de tiempo el bloque "Horno" se queda sin suministro, aunque es algo aceptable porque el WIP de este bloque únicamente desciende a 125 *agentes* (ver Figura 115). En sistemas reales, el pulmón que está a la entrada del horno lo alimentaría hasta que se reestableciera el suministro.

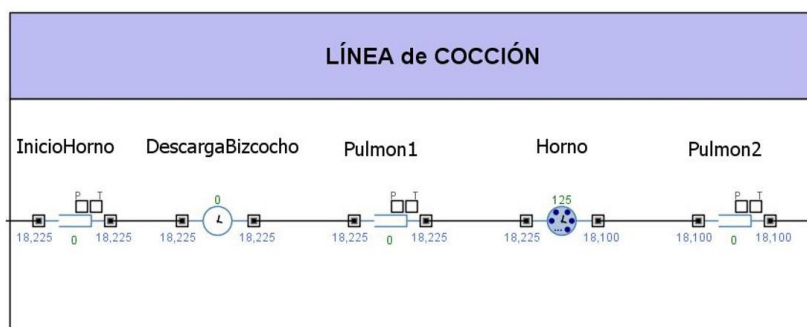


Figura 115. Estado de la simulación en $t = 08-03-2019\ 00:47:45\ AM$.

Este pequeño margen es un resultado aceptable de la simulación. Cuando el WIP de la línea de Prensado-Secado-Esmaltado se vacía, queda un pequeño remanente de piezas en el bloque "FinLineaEsmaltado". Puesto que las piezas cerámicas producidas en el periodo de entre semana por la línea de Prensado-Secado-Esmaltado, no son múltiplos de 45 Siempre quedará un pequeño remanente de piezas a la espera de completar un agente "Box" (ver Figura 116). Si se suma el WIP del horno con este pequeño remanente, el resultado es 142, suficiente para mantener la continuidad de la producción.

Esto se debe a que el sistema está ajustado al límite; en sistemas reales se suele reservar un inventario acumulado de piezas como stock de seguridad para mantener la continuidad del funcionamiento del horno frente a fenómenos intempestivos.

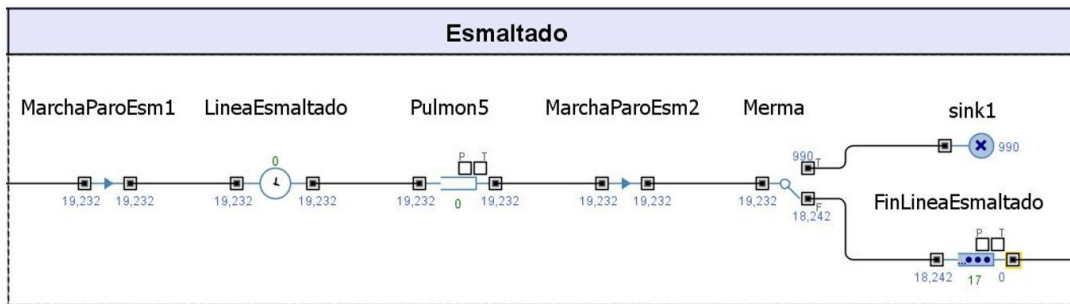


Figura 116. Estado de la simulación en t= 06/03/2019 06:57:40 AM.

8.2. MEDIDAS DE FLUJO BÁSICAS DE PRODUCCIÓN DE LOS BLOQUES QUE COMPONEN EL FLUJO DE PROCESOS DURANTE EL PERIODO DE ENTRE SEMANA.

A continuación, se recogen en una tabla las medidas de WIP arrojados por los distintos bloques de modelado de procesos durante el desarrollo de la simulación en régimen estacionario. En esa misma tabla, también se recogen las medidas de flujo básicas calculados en el Capítulo 3 Antecedentes (ver tabla 15). Como se puede observar las medidas de flujo básicas coinciden con los que se dan en el modelo, por lo que se puede afirmar que la confección del modelo de simulación es correcta y se ajusta la caracterización especificada en el Capítulo 3 Antecedentes (ver Punto 3.4 Características de los recursos de la línea de producción). En este aspecto, se puede afirmar que el modelo queda validado.

Bloque de modelado de procesos	WIP teórico (agentes)	WIP en régimen permanente (agentes)
Prensa	34722	34722
SecaderoVertical	106,84	[106 – 107]
LineaEsmaltado	6,41	[6-7]
FinLineaEsmaltado	45	45
InicioHorno	45	45
Horno	135,937	[135 - 136]
Embalaje	25	25

Tabla 10. Indicadores de producción de los bloques de modelado de procesos que componen los procesos productivos del modelo de simulación.

En relación a lo expuesto en este punto y en el anterior, se puede afirmar que según el criterio de verificación de los resultados determinado en el Punto 5.1 Verificación del modelo de simulación; los resultados que arroja el experimento de simulación del modelo desarrollado en el presente proyecto son correctos, por lo que queda verificado que el modelo queda válido.

8.3. ANÁLISIS DE LOS RESULTADOS DEL EXPERIMENTO DE OPTIMIZACIÓN.

En este punto se procederá a analizar los resultados del experimento de optimización reportados en el Punto 7.3.6. Experimento de optimización. Se analizarán los resultados de los 3 casos que se han desarrollado según el planteamiento del Punto 5.2.1. Condiciones iniciales del experimento de optimización.

8.3.1. CASO 1: VELOCIDAD DE LOS LGV 5 M/S.

Una vez se ha ejecutado este experimento, se observa que la cantidad de *agentes* "LGV" que maximizan la utilización de este recurso es de 2 (ver Figura 112). Si se realiza una simulación cambiando el valor del parámetro "NumLGV" de 5 a 2, se observa que la simulación se desarrolla sin problemas. En cambio, si la cantidad se reduce a 1, se observa que en ocasiones los "LGV" se salen de los *paths* y atraviesan la *Network* para poder transportar a tiempo los *agentes* "Box" (ver Figura 116), esto es un indicativo de que con un solo recurso "LGV" la simulación no se desarrolla con normalidad.

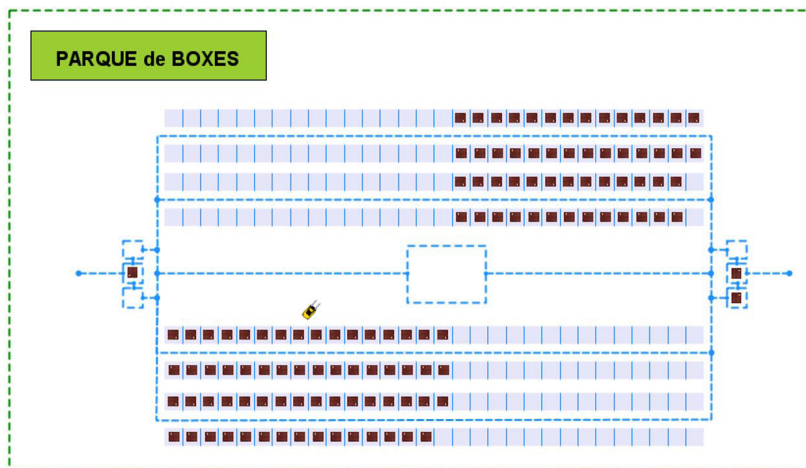


Figura 116. Funcionamiento anormal de la simulación.

8.3.2. CASO 2: VELOCIDAD DE LOS LGV 7.5 M/S.

Una vez se han modificado las condiciones iniciales del experimento y la velocidad asignada al *recurso* "LGV" es de 7,5 m/s; cuando se ejecuta el experimento, la cantidad de *agentes* "LGV" que maximizan la utilización de este recurso es de 1 (ver Figura 112). Es interesante observar, que, al aumentar la velocidad de desplazamiento de este *recurso* a lo largo de la *Network*, se necesiten menos unidades para alcanzar la utilización máxima. Esto es un indicativo claro de que el experimento de optimización se está comportando de manera correcta, y arroja resultados lógicos.

Para validar este resultado, se ha ejecutado una simulación variando el valor del parámetro "NumLGV" de 2 a 1, y se observa que el comportamiento de la simulación es correcto, y no presenta ningún fallo.

8.3.3. CASO 3: VELOCIDAD DE LOS LGV 5 M/S CONSIDERANDO PARADAS POR RECARGA DE BATERÍAS.

En este último caso, se han modificado las condiciones iniciales del experimento y la velocidad asignada al recurso LGV vuelve a ser de 5 m/s, pero en cambio la utilización máxima de este recurso se limita a 0.8333. Como resultado, se obtiene que la cantidad de *agentes* "LGV" que maximizan la utilización de este recurso es de 3 (ver Figura 114). Por otro lado, también se observa que la máxima utilización que se puede alcanzar con estos tres recursos es 0,667. Esto es un indicativo claro de que, a grosso modo, siempre va a haber 2 recursos en uso, y otro a la espera.

Analizando este resultado, se puede intuir que una estrategia para contemplar las paradas de recarga de baterías en un sistema real de estas características, es tener 3 LGV y realizar rotaciones entre ellos. De manera que, uno siempre esté en carga y los otros estén funcionando.

PLIEGO DE CONDICIONES

**DESARROLLO Y APLICACIÓN DE UN MODELO DE SIMULACIÓN PARA
MEJORAR EL ALMACENAMIENTO INTERMEDIO DE UNA PLANTA DE
PRODUCCIÓN CERÁMICA**

UNIVERSITAT JAUME I

Máster en Ingeniería Industrial

Autor: Carlos Cantos González

Directores: Fernando Romero Subirón

Pedro Rosado Castellano

CASTELLÓN, JULIO 2021

ÍNDICE. PLIEGO DE CONDICIONES.

1. Condiciones sobre los medios materiales 111

2. Condiciones del personal técnico 111

En este documento se recoge el conjunto de condiciones materiales y técnicas que aseguran la correcta implantación del presente proyecto en una empresa del sector cerámico que se ajuste a la situación actual de Azulejos S.A. y por otro lado garantizan su correcto desempeño dentro de esta.

1. CONDICIONES SOBRE LOS MEDIOS MATERIALES

Primeramente, se introduce en la Tabla 11 aquellos elementos materiales imprescindibles para poder llevar a cabo el desempeño de la herramienta desarrollada en el presente proyecto por un usuario final. Debido al carácter de éste, los materiales requeridos se centrarán en las herramientas básicas para la definición del modelo computacional y su posterior simulación.

Concepto	Descripción	Especificación
Disco duro	Espacio disponible	1.5 Gb
Memoria RAM	Espacio disponible	8 Gb
Procesador	CPU de dos núcleos	Sin especificar
Sistema Operativo	Entorno de software necesario para manipular un ordenador	Microsoft Windows 10 x64
Licencia AnyLogic	Permiso para el empleo del software de modelado y simulación.	AnyLogic 8.3.3. Professional

Tabla 11. Elementos materiales del proyecto.

2. CONDICIONES DEL PERSONAL TÉCNICO

En este punto se describirán las condiciones necesarias que debe tener el personal técnico encargado, tanto de la manipulación del modelo computacional, como de implantar los resultados en el entorno de la empresa.

- Formación académica:
 - Grado en tecnologías industriales o similar
 - Master en ingeniería industrial
 - Curso formativo acerca del software de modelado y simulación AnyLogic.
- Experiencia profesional mínima:
 - Dos años de experiencia en la empresa dentro del entorno de producción.

PRESUPUESTO

**DESARROLLO Y APLICACIÓN DE UN MODELO DE SIMULACIÓN PARA
MEJORAR EL ALMACENAMIENTO INTERMEDIO DE UNA PLANTA DE
PRODUCCIÓN CERÁMICA**

UNIVERSITAT JAUME I

Máster en Ingeniería Industrial

Autor: Carlos Cantos González

Directores: Fernando Romero Subirón

Pedro Rosado Castellano

CASTELLÓN, JULIO 2021

ÍNDICE. PRESUPUESTO.

1. Especificación de los elementos materiales 119
2. Estudio de costes 119

En este documento se presenta el presupuesto del proyecto descrito en la memoria. Para ello, el capítulo se ha estructurado en dos partes. En primer lugar, se listan los elementos que deben adquirirse, junto a su respectivo precio unitario y la cantidad requerida. Seguidamente, se desglosa la inversión del proyecto computándose el costo total realizado.

1. ESPECIFICACIÓN DE LOS ELEMENTOS MATERIALES

En la Tabla 12 se listan el conjunto de elementos necesarios para el desarrollo del proyecto, las cantidades requeridas en cada caso, así como la unidad a partir de la cual se medirá el coste asociado a las mismas.

Concepto	Descripción	Cantidad	Unidad
Ordenador	Aparato para el desarrollo del modelo de la línea de producción cerámica y su simulación.	1	€
Licencia MS Windows 10 x64	Permiso para el empleo del sistema operativo necesario para manipular el ordenador	1	€
Licencia AnyLogic	Permiso para el empleo del software de modelado y simulación.	1	€
Curso de formación	Curso formativo acerca del software de modelado y simulación AnyLogic.	40 h	€/ h
Mano de obra	Trabajo dedicado a formarse en el software de simulación, al desarrollo del modelo y a la simulación del mismo.	480 h	€/ h

Tabla 12. Especificación de los elementos materiales.

2. ESTUDIO DE COSTES

Los costes relacionados al desarrollo del proyecto descrito en este documento y, por tanto, equivalentes a la inversión a realizar, se resumen en la Tabla 13. Se incluyen en esta tabla los elementos descritos en el apartado anterior y se excluyen todos aquellos gastos que derivan de la solución obtenida de la simulación, al ser muy específicos al escenario de la empresa o caso estudiado.

Concepto	Cantidad	Coste unitario	Coste
Ordenador	1 unidad	780 €/ ud.	780 €
Licencia MS Windows 10 x64	1 unidad	100 €/ ud.	100 €
Licencia AnyLogic	1 unidad	13950 €/ ud.	13.950 €
Curso de formación	40 h	90 €/ h	3.600 €
Mano de obra	480 h	45 €/ h	21.600 €
			40.030 €

Tabla 13. Coste total del desarrollo del proyecto.

Con todo ello, dicho coste total invertido asciende a un total de cuarenta y cuatromil cuatrocientos treinta euros (44.030 €).

ANEXO

**DESARROLLO Y APLICACIÓN DE UN MODELO DE SIMULACIÓN PARA
MEJORAR EL ALMACENAMIENTO INTERMEDIO DE UNA PLANTA DE
PRODUCCIÓN CERÁMICA**

UNIVERSITAT JAUME I

Máster en Ingeniería Industrial

Autor: Carlos Cantos González

Directores: Fernando Romero Subirón

Pedro Rosado Castellano

CASTELLÓN, JULIO 2021

ÍNDICE. ANEXO.

1. Modelado basado en agentes de Anylogic	127
1.1. Agentes en Anylogic	127
1.2. Parámetros y variables	128
1.3. Funciones	129
1.4. Eventos	129
1.4.1. Tipos de eventos	130
1.5. Schedule	130
1.6. Network	131
1.6.1. Atractor	132
1.6.2. PalletRack	133
1.6.2.1. RackSystem	134
1.7. Source	135
1.8. Sink	136
1.9. Delay	136
1.10. Queue	137
1.11. ResourcePool	137
1.12. Service	138
1.13. Enter	139
1.14. Exit	140
1.15. RackPick	140
1.16. Pickup	143
1.17. Dropoff	144
1.18. Hold	145
1.19. Move To	145
1.20. SelectOutput	146
1.21. Batch	147

1. MODELADO BASADO EN AGENTES DE ANYLOGIC

1.1. AGENTES EN ANYLOGIC.

Los *agentes* son los principales bloques de construcción del modelo AnyLogic; es una unidad de diseño de modelos que puede tener comportamiento, memoria (historial), tiempo, contactos, etc. Los *agentes* pueden representar cosas muy diversas: vehículos, unidades de equipo, proyectos, productos, ideas, organizaciones, inversiones, terrenos, personas en diferentes roles, etc (ver Figura 117).



Figura 117 Ejemplos de tipos de Agentes en Anylogic.

Dentro de un *agente* se pueden definir variables, eventos, gráficos de estado, stock de sistemas dinámicos y diagramas de flujo. También se pueden integrar otros *agentes* y añadir diagramas de flujo de procesos; se pueden definir tantos tipos de *agentes* en el modelo como diferentes tipos de *agentes* haya. Además, El agente puede representar a un solo *agente* o a una *población de agentes*; dicha población representa un conjunto de *agentes* del mismo tipo.

El diseño de un *agente* normalmente comienza con la identificación de sus atributos, comportamiento e interfaz con el mundo externo. En el caso de un gran número de *agentes* con conexiones dinámicas (como las redes sociales), los *agentes* pueden comunicarse mediante funciones de llamada.

El estado interno y el comportamiento del *agente* pueden ser implementados de varias maneras. El estado del *agente* puede ser representado por un número de variables, por el mapa de estado, etc. El comportamiento puede ser por así decirlo pasivo (por ejemplo, hay agentes que sólo reaccionan a la llegada de mensajes o a llamadas de función y no tienen su propio tiempo), o activo, cuando la dinámica interna (timeouts o procesos de dinámica de sistemas) del *agente* hace que actúe. En este último caso, lo más probable es que los *agentes* tengan objetos *event* y/o *statechart* dentro.

Por otra parte, los *agentes* pueden estar embebidos en otros *agentes* a cualquier nivel deseado. Los objetos embebidos son instancias de otros tipos de *agentes*, embebidas por cada instancia de este tipo de *agente*. La integración es el paso para crear la jerarquía del modelo.

Por defecto, los objetos embebidos se muestran simplemente como un icono . Pero si se ha dibujado un icono para este tipo de *agente*, este icono se mostrará en lugar del predeterminado. Además de las formas marcadas como iconos, el icono también incluye puertos públicos y variables.

1.2 PARÁMETROS Y VARIABLES.



Se pueden definir los datos del agente utilizando parámetros (ver Figura 118) y variables (ver Figura 119). Los parámetros se utilizan normalmente para definir algunas características del objeto modelado. Son útiles cuando las instancias de objetos tienen el mismo comportamiento descrito en la clase, pero difieren en algunos valores de parámetros. Generalmente se utilizan parámetros numéricos, mientras que se puede definir un parámetro de cualquier clase Java. Por otra parte, el agente puede contener variables; las variables se utilizan generalmente para almacenar los resultados de la simulación del modelo, o para modelar algunas características de los objetos cambiando con el tiempo.

Existe una clara diferencia entre variables y parámetros. Una variable representa el estado de un modelo y puede cambiar durante la simulación, mientras que un parámetro se utiliza comúnmente para describir objetos estáticamente. Un parámetro es normalmente una constante en una sola simulación, y se cambia sólo cuando se necesita ajustar el comportamiento del modelo.

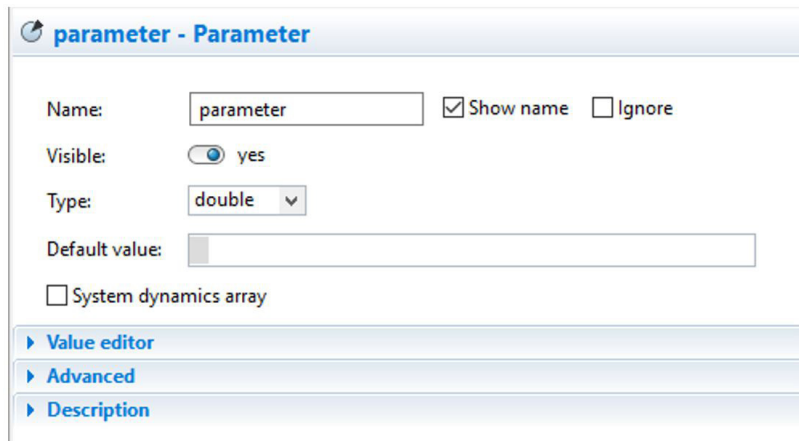


Figura 118. Un Parámetro definido en Anylogic.

Se utiliza una variable en lugar de un parámetro si necesita modelar alguna unidad de datos que cambia continuamente con el tiempo; y se utiliza un parámetro en lugar de una variable si sólo necesita modelar algún parámetro de un objeto modificado sólo en determinados momentos de tiempo. Todos los parámetros y variables son visibles y modificables a lo largo de la ejecución del modelo, por lo que puede simplemente ajustar su modelo cambiándolo en tiempo de ejecución.

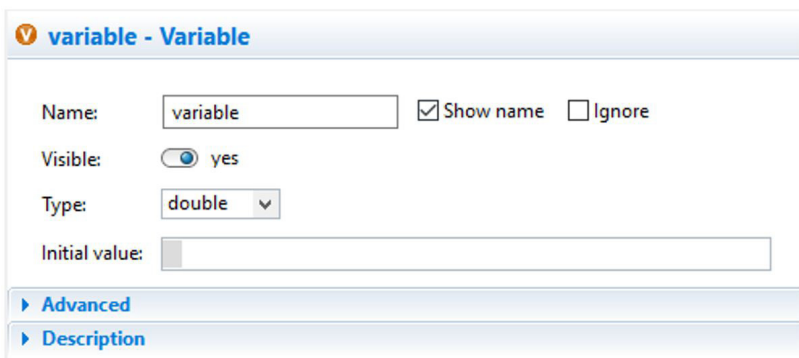
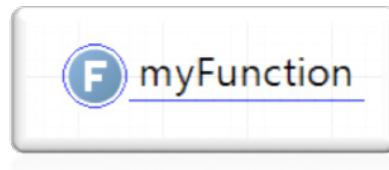


Figura 119. Una Variable definida en Anylogic.

1.3. FUNCIONES.



AnyLogic permite definir e implementar funciones en los modelos de simulación (ver Figura 120). Cuando se define una función, esta devolverá el valor de una expresión cada vez sea llamada desde el modelo. Las funciones son útiles cuando se necesita reutilizar un mismo algoritmo en varios lugares del modelo, además, puesto que se escriben en Java, tienen todas las ventajas de este lenguaje, tales como operadores condicionales y cíclicos, campos y demás.

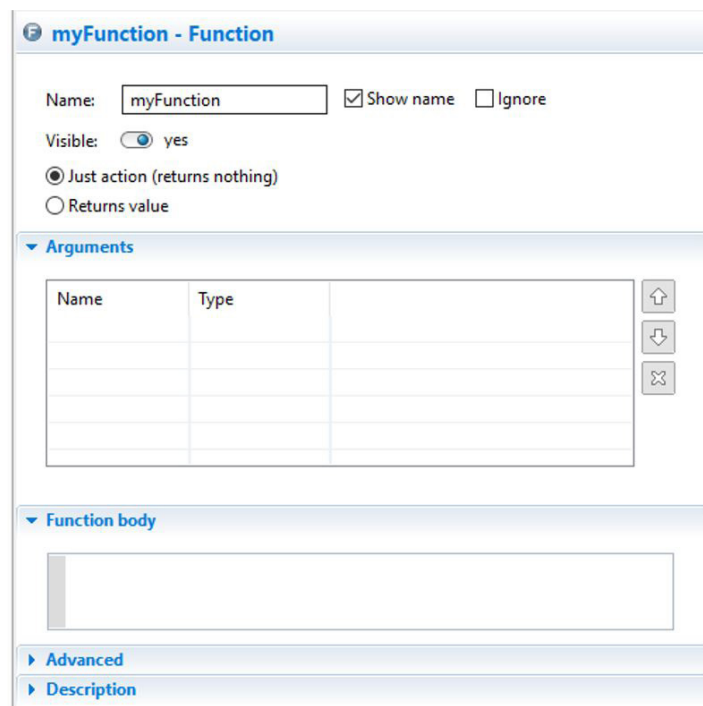
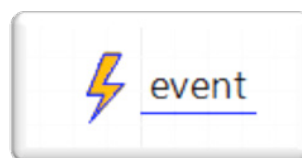


Figura 120. Una función definida en Anylogic.

Para constituir una función se pueden definir algunos argumentos opcionales para permitir el paso de algunos datos de entrada adicionales necesarios para los cálculos de dicha función. Si se ha definido algún parámetro para la función, se le debe pasar los valores de parámetro cada vez que se llame a la función. Los valores se deben proporcionar en el orden en que se definen en la entrada de argumentos de la función.

1.4. EVENTOS.



Los *Eventos* son la forma más sencilla de programar alguna acción en el modelo (ver Figura 121). Por lo tanto, los *eventos* se utilizan comúnmente para modelar retardos y tiempos de espera. A veces se hace lo mismo usando transiciones cronometradas en los gráficos de estado, pero los *eventos* pueden ser más eficientes.

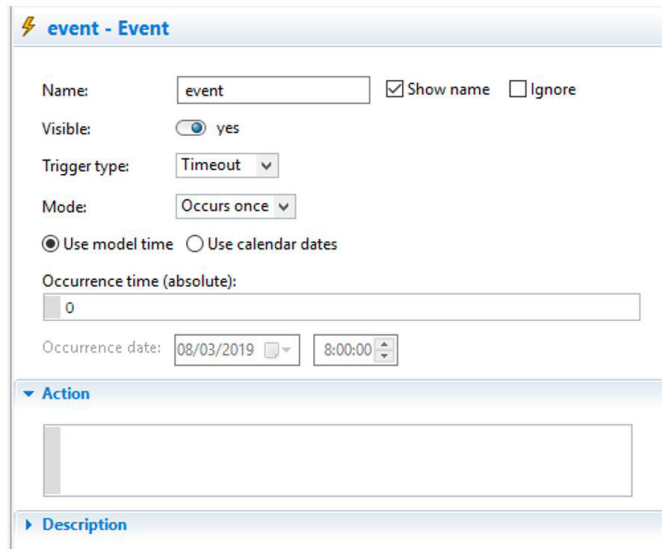


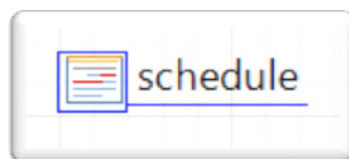
Figura 121. Un evento definido en Anylogic.

1.4.1. TIPOS DE EVENTOS.

Hay tres tipos de eventos (ver **Triger type** Figura 121):

- **Timeout triggered event:** Consiste en un *evento* que se inicia al finalizar un tiempo de espera. Se utiliza cuando se necesita programar una acción en un momento determinado (o en una fecha determinada) El *evento* se produce exactamente en al finalizar el tiempo de espera programado después de que se iniciado la cuenta atrás. El Timeout triggered event tiene aún más características: puede especificar que expire una sola vez o cíclicamente, o que sea totalmente controlado por el usuario.
- **Condition triggered event:** Es un *evento* que se ejecuta a partir de una condición. Se utiliza cuando se desea supervisar una determinada condición y ejecutar una acción cuando esta condición se convierte en verdadera.
- **Rate triggered event:** Este *evento* es desencadenado a partir de una la tasa de llegada. Se utiliza para modelar una tasa de llegada de eventos a partir de una distribución estadística (Ej: Distribución de Poisson). Con frecuencia es necesario modelar la tasa de llegadas: por ejemplo, llegadas de clientes en sistemas de colas, llegadas de transacciones en modelos de red basados en servidores, etc.

1.5. SCHEDULE.



AnyLogic proporciona a los usuarios un *Schedule*, un elemento especial que permite definir cómo cambia un valor en el tiempo de acuerdo con el patrón definido (opcionalmente cíclico) (ver Figura 122).

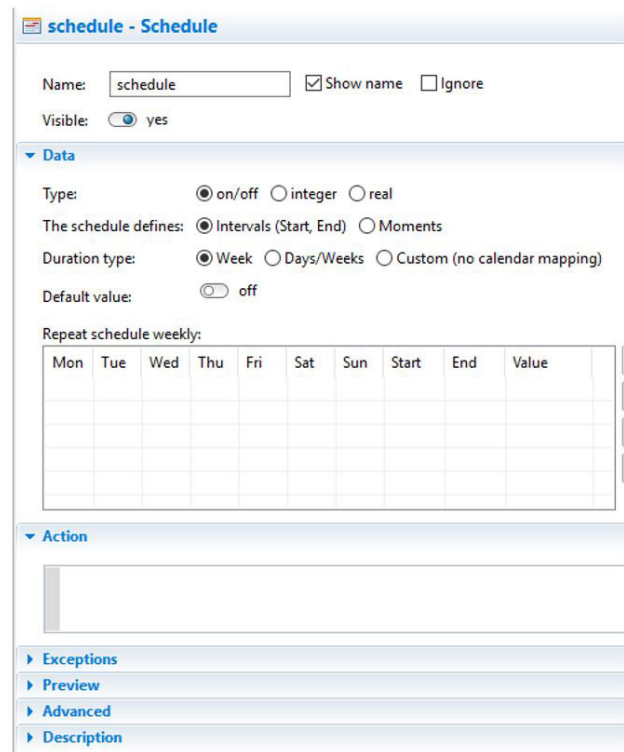


Figura 122. Un schedule definido en Anylogic.

El *Schedule* se utiliza frecuentemente para:

- Planificar la gestión de un conjunto de recursos definido mediante el elemento *ResourcePool*.
- Definir una programación para la generación de agentes dentro de un modelado de procesos a partir del bloque *Source*.

El *Schedule* se puede definir mediante intervalos o momentos de tiempo discretos. Se utilizan intervalos cuando se necesita modelar cómo un valor cambia continuamente en el tiempo (normalmente de acuerdo con un patrón cíclico); por ejemplo, los intervalos se utilizan para definir los horarios de trabajo para los turnos de los trabajadores, el patrón cíclico de la tasa de llegada de peatones/agentes, etc. Por otra parte, se utilizan momentos cuando se necesita definir una secuencia de momentos clave de tiempo y algunos valores correspondientes a estos momentos en particular (o realizar algunas acciones).

Además, puede asociar acciones con momentos clave del *Schedule* (ya sea intervalos de tiempo de conmutación o momentos de tiempo, dependiendo del modo de programación), escribiendo código Java en la sección Acción de las propiedades de la programación. De modo que se puede utilizar el *Schedule* como una herramienta perfecta para programar un número ilimitado de acciones en el futuro (probablemente, asignadas a las fechas del calendario en particular).

1.6. NETWORK.

Una *Network* es un conjunto de nodos interconectados con segmentos (ver Figura 124). Se crea una *Network* cuando se necesita un modelado basado en un Lay-Out. Típicamente se utiliza cuando los procesos modelados se desarrollan en un espacio físico determinado e incluyen el movimiento de agentes y recursos.

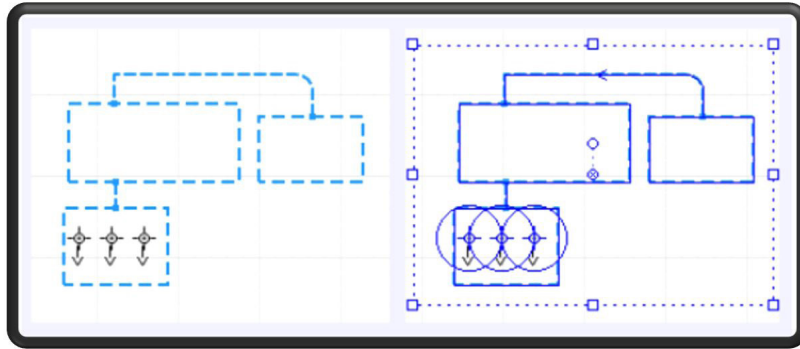


Figura 123. Ejemplo de una *Network* modelada en Anylogic.

La topología de la *Network* se define dibujando elementos específicos (*nodes* y *paths*). En la red, un *node* define el lugar donde los *agentes* pueden permanecer, mientras que los *paths* que conectan los *nodes* definen gráficamente las rutas que los *agentes* pueden tomar cuando se mueven de un nodo a otro (ver Figura 124).

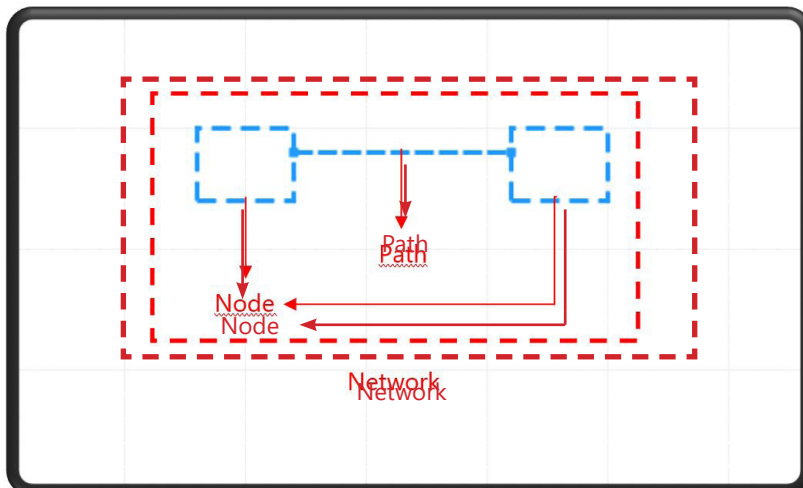


Figura 124. Elementos básicos que conforman una *Network*.

El movimiento se realiza siempre a lo largo de la trayectoria más corta entre los *nodes* de origen y destino. Los *agentes* y las unidades de recursos pueden tener velocidades individuales y estas velocidades pueden cambiar dinámicamente; por ejemplo, se puede ajustar diferentes velocidades para la carretilla elevadora cargada y la descargada. Se supone que los segmentos tienen capacidad ilimitada y que los *agentes* que se mueven a lo largo de un segmento no interfieren.

1.6.1. ATRACTOR.

Un *Attractor* permite controlar la ubicación del agente dentro de un *node* rectangular o un *node* poligonal. La flecha del atractor define la orientación de los *agentes* ubicados en un atractor, esto es generalmente importante cuando se tiene animación 2D o 3D y se desea que las formas de animación del agente se enfrenten de la manera exacta (por ejemplo, se puede desear que el cliente se enfrente a un cajero automático) (ver Figura 125).

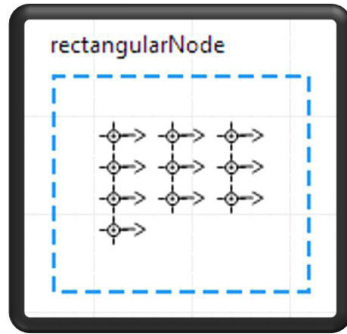


Figura 125. *Attractors* dentro de un nudo.

1.6.2. PALLETRACK

El elemento *PalletRack* se utiliza para definir gráficamente zonas de almacenamiento.

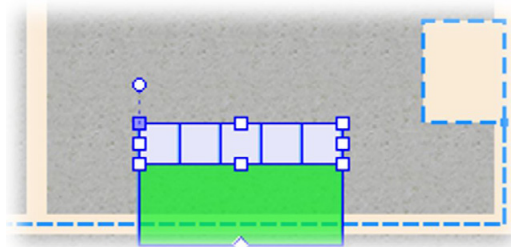


Figura 126. Ejemplo del elemento *PalletRack* definido en Anylogic.

En la Figura 126 se muestra un ejemplo de cómo queda definido el elemento *PalletRack* en Anylogic y en la Figura 127 se muestra su representación en 3D a partir de una animación. Como se observa, se ha creado un conjunto de *nodes* y *paths* a lo largo de un pasillo dado, y de esta manera se pueden gestionar los *agentes* almacenados en las ubicaciones de al lado del pasillo del *PalletRack* independientemente del nivel en el que se encuentren.

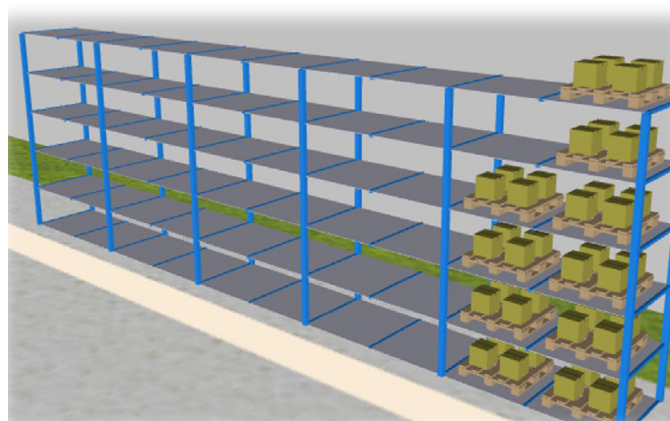


Figura 127. Representación del *PalletRack* en 3D.

El *PalletRack* puede tener una de estas tres configuraciones:

- Un pasillo, una fila de racks
- Dos pasillos, una fila de racks
- Un pasillo, dos filas de racks

En la Figura 128 se ilustra como quedan definidos en el espacio 2D de una Network de Anylogic las distintas configuraciones del elemento *PalletRack* descritas anteriormente.

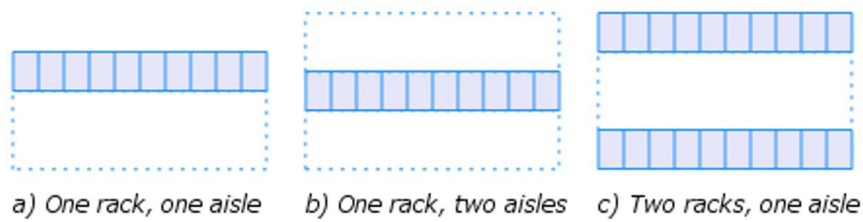
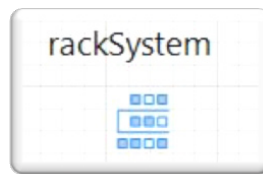


Figura 128. Distintas configuraciones para definir un PalletRack.

El ancho de las celdas del *PalletRack* (es decir, su dimensión a lo largo de la fila) se calcula automáticamente como la longitud del pasillo dividida por el número de posiciones por fila. Por otro lado, la profundidad de la celda es especificada por el usuario en el parámetro correspondiente. Cabe añadir que la profundidad de la celda, la topología de los pasillos y los elementos de la red dibujados por el usuario tienen que ser consistentes, por ejemplo, no debe haber intersecciones de *nodes*.

1.6.2.1. RackSystem



Para crear sistemas de celdas de almacenamiento complejos basados en una composición de varios *PalletRack* agregados, se utiliza el bloque *RackSystem* (ver Figura 129). Este bloque se utiliza para representar un número de *PalletRack* como un solo bloque con múltiples filas y pasillos y sirve como punto central de acceso y gestión para ellos. Esto tiene sentido si los almacenamientos se utilizan de forma similar.

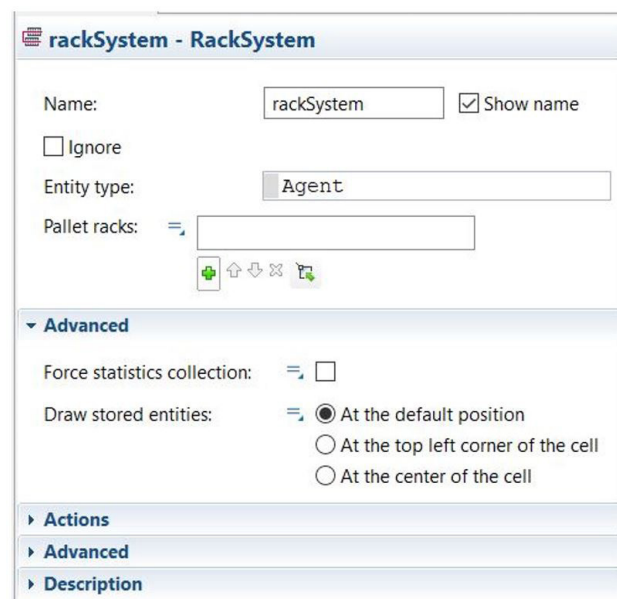


Figura 129. Un RackSystem definido en Anylogic.

Para crear un *RackSystem*, primero debe definir gráficamente cada almacén individual con un *PalletRack* y especificar el número de posiciones y niveles del almacén. No es necesario que los *PalletRack* sean idénticos: pueden tener diferentes capacidades y tamaños de celdas; tampoco es necesario que estén alineados gráficamente. A continuación, debe proporcionar la lista de *PalletRack* que se hayan definido en bloque de modelado de procesos del *RackSystem*.

1.7. SOURCE.



El bloque *Source* genera *agentes* y es generalmente un punto de partida de un modelado de procesos. Los *agentes* pueden ser estándar o de cualquier tipo definido por el usuario. Se pueden personalizar los *agentes* generados especificando el tipo de agente en el campo **New entity** y, a continuación, especificar la acción que se debe realizar antes de que el *agente* salga del bloque *Source* en el campo de acción **On exit**.

Hay varias maneras de definir cuándo y cuántos *agentes* deben generarse:

- A partir de la tasa de un ratio de entrada de agentes por unidad de tiempo (Que se puede cambiar dinámicamente llamando a `set_rate()`).
- Mediante una tasa de llegada de agentes definida a partir de una función estadística.
- Utilizando un Schedule que defina, horario de tiempos de llegada exactos y cantidades.
- Y también puede llamar programáticamente a la función `inject()` de este bloque.

Por ejemplo, como se muestra en la Figura 131 (ver **Arrivals defined by**) se puede modelar un proceso de Poisson mediante una tasa de llegada de *agentes* especificando el tiempo entre llegadas distribuido exponencialmente; y también puede establecer el número de *agentes* en cada llegada y limitar el número total de llegadas.

source1 - Source

Name: Show name Ignore

Entity type:

Arrivals defined by:

Interarrival time:

Multiple entities per arrival:

Limited number of arrivals:

New entity:
(to create a custom type, drag it from the palette)

Location of arrival:

Advanced

Actions

On before arrival:

On at exit:

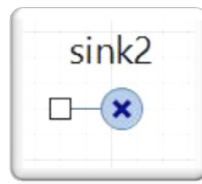
On exit:

Advanced

Description

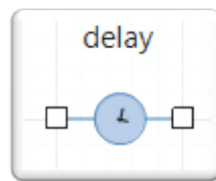
Figura 130. Un bloque de modelado de procesos *Source* definido en Anylogyc mediante una tasa de llegadas.

1.8. SINK.



El bloque de modelado de procesos *Sink*, es normalmente un punto final en un modelo de procesos y su función es eliminar a los *agentes*. A menos que use un bloque *Sink*, los *agentes* no serán removidos del modelo y eliminados; por lo tanto, no debe dejar un puerto desconectado al final de un proceso.

1.9. DELAY.



El bloque *Delay* retrasa a los *agentes* durante un tiempo determinado (ver Figura 131). Este tiempo de retardo se evalúa dinámicamente, puede ser estocástico y puede depender del *agente*, así como de cualquier otra condición. Opcionalmente, puede calcularse como la duración de la ruta de animación del *Delay* dividida por la "velocidad" del *agente*.

Por otra parte, el bloque *Delay* puede ejercer un retardo en varios *agentes* de manera simultánea e independientemente, todo ello sin que se exceda la capacidad de la cola del *Delay*. Si el número de *agentes* que se encuentran en la cola del *Delay* excede su capacidad, el objeto *Delay* les permitirá completar sus tiempos de retardo y no dejará entrar nuevos *agentes* hasta que el número caiga por debajo de su capacidad.

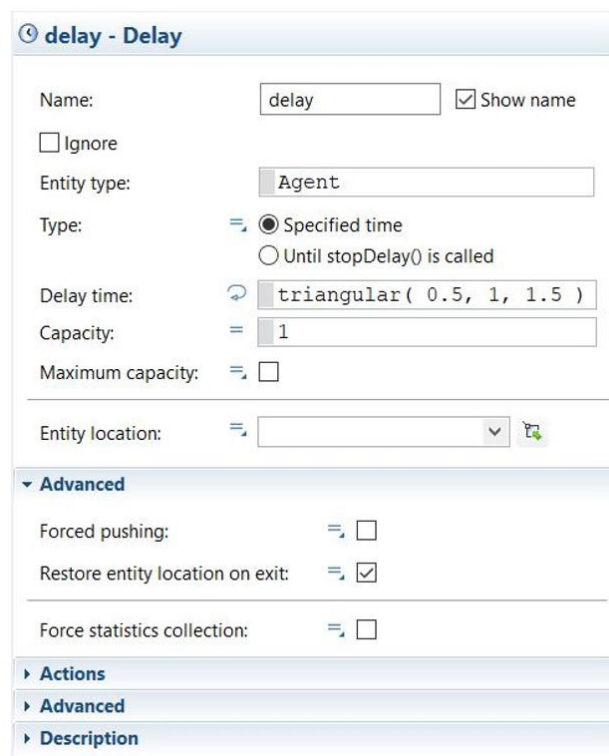
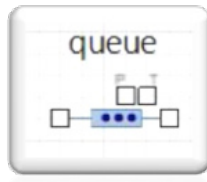


Figura 131. Un bloque modelado de procesos *Delay* definido en Anylogic.

1.10. QUEUE.



Una *Queue*, es una cola de *agentes* esperando ser aceptados por el siguiente objeto en el flujo de proceso, o un almacenamiento de propósito general para los *agentes*. Opcionalmente, se puede asociar un tiempo máximo de espera con un *agente*. También puede eliminar *agentes* programáticamente desde cualquier posición de la cola o cambiar la capacidad de la cola dinámicamente.

La disciplina de colas puede ser FIFO (predeterminada) (ver **Queuing** Figura 132), LIFO o basada en prioridades. La prioridad puede almacenarse explícitamente en el *agente* o calcularse en función de las propiedades del agente y de las condiciones externas.

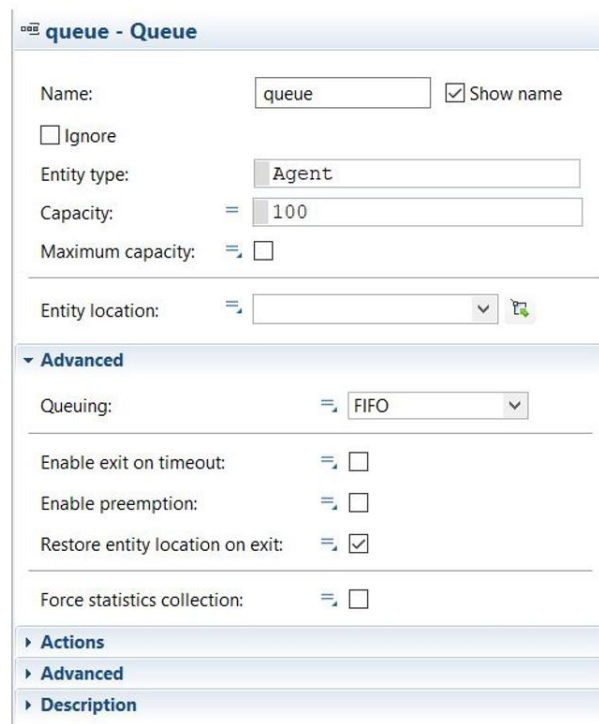
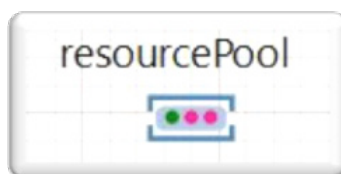


Figura 132. Un bloque de modelado de procesos Queue definido en Anylogic.

1.11. RESOURCEPOOL.



El bloque *ResourcePool* define un conjunto de unidades de recursos que pueden ser retenidas y liberadas por los *agentes* utilizando bloques de modelado de procesos. Las unidades de recursos en de un *ResourcePool* pueden tener propiedades individuales, pueden ser animadas, recoger estadísticas basadas en unidades, etc.

En Anylogic se puede definir recursos personalizados que representen por ejemplo personal, equipo, etc. Por otra parte, la capacidad de este objeto (el número de unidades de recursos) puede definirse directamente (ver **Capacity** Figura 133).

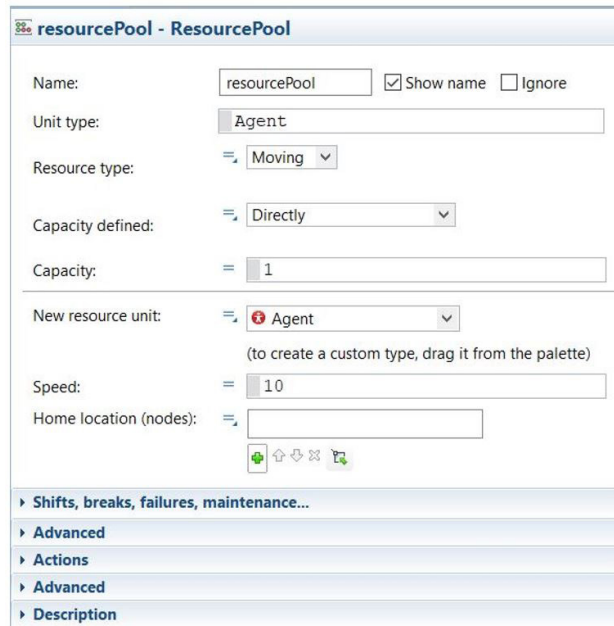


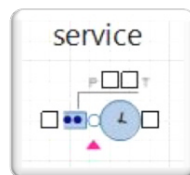
Figura 133. Un bloque de *ResourcePool* definido en Anylogic.

Los recursos pueden ser de tres tipos: *static*, *moving* y *portable* (ver **Unit Type** Figura 133):

- Los recursos tipo *static* son recursos que están ligados a una ubicación particular (es decir, un *node*) dentro de una Network y no pueden moverse o ser movidos por otro *agente*.
- Los recursos tipo *moving* tienen la capacidad de moverse por sí mismos; y por ejemplo, pueden representar personal, vehículos, etc.
- Los recursos tipo *portable* son un tipo de recursos que solo pueden ser movidos por *agentes* o por recursos móviles; pueden representar palés, containers, etc.

Los recursos tipo *moving* y *portable* tienen sus lugares de origen dentro de una Network donde pueden regresar o ser devueltos opcionalmente.

1.12. SERVICE.



El bloque de modelado de procesos *Service* conceptualmente se entiende como una operación sobre un *agente* que dura un tiempo determinado y que, a su vez, necesita una cantidad de recursos para poderla llevar a cabo. Para ello, se apodera de un número determinado de unidades de recursos, retrasa al *agente* y libera las unidades aprehendidas una vez ha finalizado el tiempo de retardo. Este bloque debe usarse si el *agente* no necesita hacer nada más que ejecutar un retardo entre que se toman los recursos hasta que se liberen más tarde.

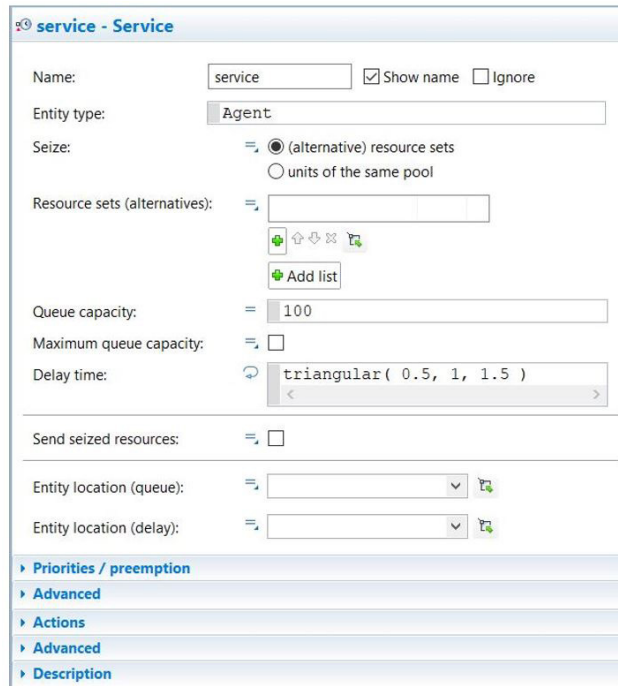


Figura 134. Un bloque de modelado de procesos *Service* definido en Anylogic.

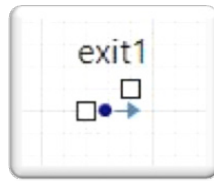
Para que el bloque *Service* pueda apoderarse de recursos, se debe seleccionar el bloque de *ResourcePool* de donde se quiera proveer de recursos al bloque (ver **Resource sets** Figura 134). Una vez se ha seleccionado, se debe especificar cuántos recursos son necesarios para completar la operación en el bloque *Service*.

Por otra parte, el bloque *Service* cuenta con un retardo que mantiene a los *agentes* durante un tiempo determinado. Este tiempo de retardo, al igual que en el bloque de modelado de procesos *Delay*, se evalúa dinámicamente; puede ser estocástico y puede depender del *agente*, así como de cualquier otra condición (ver **Delay time** Figura 134). Además, también incorpora una cola de *agentes* esperando a poder completar el tiempo de retardo programado en este bloque (ver **Queue** Figura 134). Si el número de *agentes* excede la capacidad de la cola, el bloque *Service* les permitirá completar sus tiempos de retardo y no dejará entrar nuevos *agentes* hasta que el número caiga por debajo de su capacidad.

1.13. ENTER.

El bloque de modelado de procesos *Enter* inserta los *agentes* en un punto concreto del modelo de procesos. A diferencia del bloque *Source*, estos *agentes* ya deben existir en el modelo. Este bloque se utiliza normalmente para transferir los *agentes* ya creados al flujo de procesos. Para insertar el *agente* a través de este bloque se llama a su función `take(agent)`.

1.14. EXIT.



El bloque de modelado de procesos *Exit* sirve para sacar a los *agentes* entrantes del flujo de procesos y permite al usuario especificar qué hacer con ellos. Si el bloque *Exit* se combina con el bloque *Enter*, se puede implementar un enrutamiento personalizado en el modelado de procesos para los *agentes* que han sido insertados a través de él.

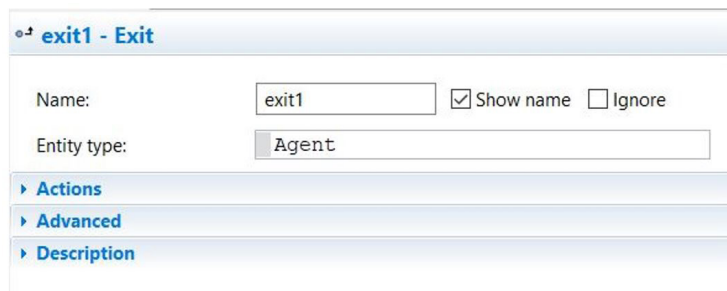
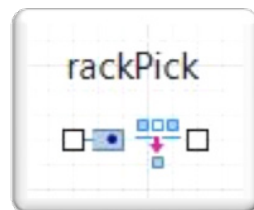


Figura 136. Un bloque de modelado de procesos Exit definido en Anylogic.

1.15. RACKPICK.



El bloque de modelado de procesos *RackPick* toma un *agente* de una celda especificada perteneciente a un *PalletRack* o a un *RackSystem*, y lo traslada a la ubicación de destino especificada dentro de la *Network*. Esta acción se hace opcionalmente con la ayuda de un *ResourcePool* desde donde se toma un determinado número de recursos para realizar el traslado (ver la opción **Use resources to move** Figura 137). En la Figura 138 se muestran los parámetros de configuración adicionales que aparecen al indicar que el bloque *RackPick* va a necesitar recursos para realizar la operación de traslado.

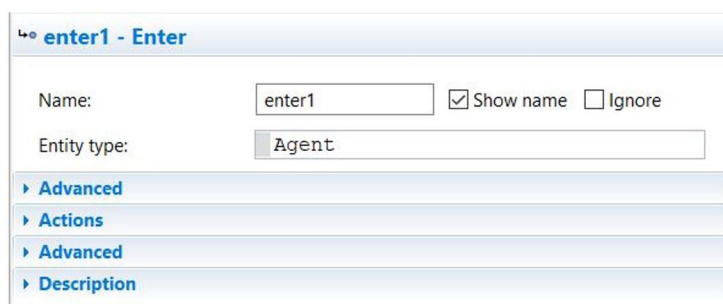


Figura 135. Un bloque de modelado de procesos Enter definido en Anylogic.

Figura 137. Un bloque de modelado de procesos *RackPick* definido en Anylogic.

Figura 138. Parámetros de configuración adicionales para gestionar la utilización de recursos dentro de un bloque de modelado de procesos *RackPick*.

Si se utilizan recursos para mover al *agente*, el bloque *RackPick* lo toma y lo lleva a la ubicación de la celda donde se encuentra el agente que se debe trasladar. Una vez allí, el *recurso* embebe al *agente* y lo traslada al destino especificado dentro de la *Network* para depositarlo. Una vez la operación se ha realizado, el bloque libera el *recurso*. Además, se puede elegir si el *agente* (Con los recursos agregados) se mueve suavemente desde la celda hasta el pasillo y luego a lo largo del pasillo; o si salta desde la celda hasta el pasillo y luego se mueve justo a lo largo del pasillo. Este último caso puede ser útil cuando se animan almacenamientos automatizados.

The screenshot shows the configuration interface for a 'RackStore' block in Anylogic. The title bar reads 'rackStore - RackStore'. The main area contains several settings:

- Name:** A text box containing 'rackStore' and a checked 'Show name' checkbox.
- Ignore:** An unchecked checkbox.
- Entity type:** A dropdown menu set to 'Agent'.
- Pallet rack / Rack system:** A dropdown menu that is currently empty.
- The cell is:** A dropdown menu set to 'Chosen automatically'.
- Choose cell closest to:** A dropdown menu set to 'Front of storage/zone'.
- Move entity to:** A dropdown menu set to 'cell (with elevation)'.
- Elevation time per level:** A text box containing the number '10'.
- Use delay:** An unchecked checkbox.
- Entity location (queue):** A dropdown menu that is currently empty.

Below these settings are four expandable sections:

- Resources:** A section with a collapsed arrow and the option 'Use resources to move' which is unchecked.
- Actions:** A section with a collapsed arrow.
- Advanced:** A section with a collapsed arrow.
- Description:** A section with a collapsed arrow.

Figura 139. Un bloque de modelado de procesos *RackStore* definido en Anylogic.

Al igual que en el bloque de modelado de procesos *RackPick*, esta acción se hace opcionalmente con la ayuda de un *ResourcePool* desde donde se toma un determinado número de *recursos* para realizar el traslado. En la Figura 140 se muestran los parámetros de configuración adicionales que aparecen al indicar que el bloque *RackStore* va a necesitar recursos para realizar la operación de traslado.

▼ Resources

Use resources to move:

Resource sets (alternatives):

Move at the speed of resource:

Task priority:

Task may preempt:

Task preemption policy:

After release resources: Return to home location
 Stay where they are

Return home: each time
 if no other tasks
 custom

Customize resource choice:

► Actions

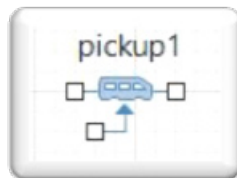
► Advanced

► Description

Figura 140. Parámetros de configuración adicionales para gestionar la utilización de recursos dentro de un bloque de modelado de procesos *RackStore*.

Si se utilizan recursos para mover al *agente*, el bloque *RackStore* los toma, los lleva a la ubicación dentro de la *Network* donde se encuentra el *agente* que se desea almacenar. Una vez allí, el recurso embebe al *agente* y lo traslada al destino especificado dentro de la *Network* para depositarlo. Una vez la operación se ha realizado, el bloque libera el *recurso*. Además, al igual que en el bloque *RackPick*, se puede elegir si el *agente* (con los *recursos* agregados) se mueve suavemente a lo largo del pasillo y luego hasta la celda de destino; o si se mueve a lo largo del pasillo y salta justo en la celda de destino.

1.16. PICKUP.



El bloque de modelado de procesos *Pickup* elimina *agentes* de la cola de un bloque *Queue* conectado en su puerto de carga y los embebe dentro de un *agente* entrante (*agente* contenedor) (ver **Container, Element types** Figura 141). Cuando un *agente* entrante llega la entrada del bloque *Pickup*, esta comprueba el nivel de la cola *Queue* que tiene conectada en el puerto de carga. Si el contenido de la cola es igual a la cantidad especificada en el parámetro **Quantity** (ver Figura 141), esa cantidad de *agentes* se embeben en el *agente* contenedor y este avanza hasta el siguiente bloque de modelado de procesos; cabe destacar que toda la operación se realiza en un tiempo nulo.

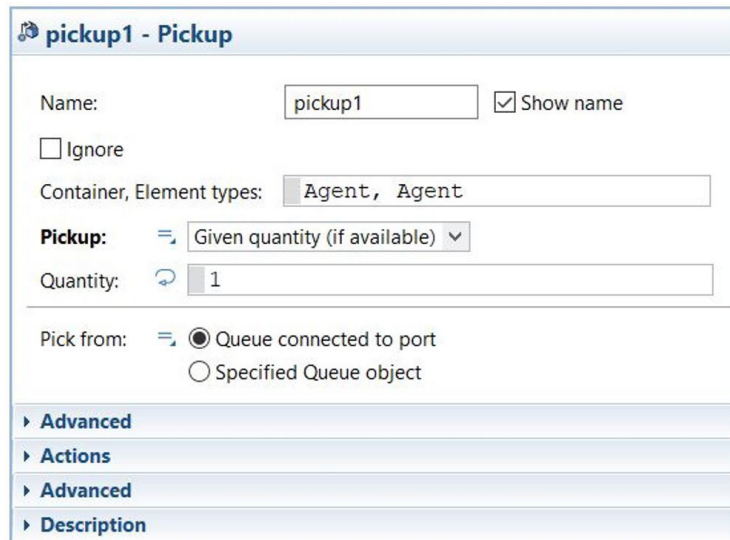
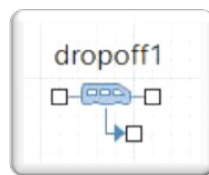


Figura 141. Un bloque de modelado de procesos *Pickup* definido en Anylogic.

1.17. DROPOFF.



El bloque de modelado de procesos *Dropoff* es un objeto que se complementa con el bloque *Pickup*. Este bloque elimina los *agentes* embebidos en un *agente* contenedor entrante y los envía a través del puerto de descarga a la cola del bloque que tenga conectado.

A diferencia del bloque *Pickup*, aquí los *agentes* se desagregan del *agente* contenedor cuando se satisface una la condición dada (ver **Condition** Figura 142); mientras ello no sea así, el *agente* contenedor permanecerá a la espera dentro del bloque *Dropoff*. Cabe añadir que la operación de descarga de *agentes* embebidos es de tiempo 0.

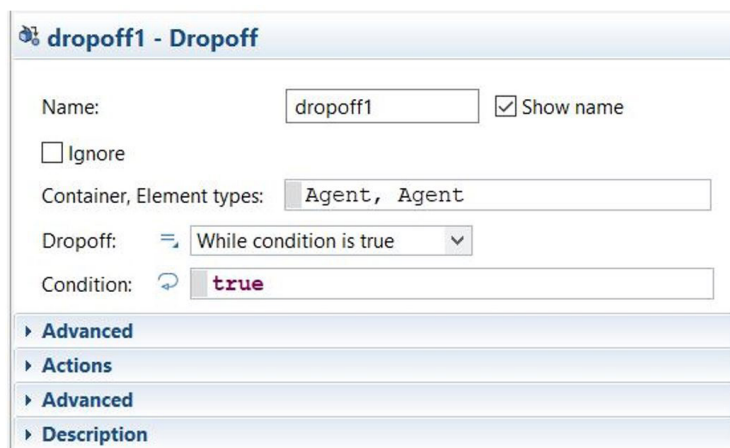
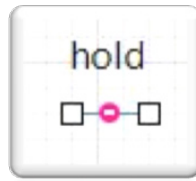


Figura 142. Un bloque de procesos *Dropoff* definido en Anylogic.

1.18. HOLD.



El bloque de modelado de procesos *Hold* sirve para bloquear el flujo de *agentes* a lo largo de un modelado de procesos a partir de una conexión particular. Se utiliza, por ejemplo, cuando el bloque función siguiente puede aceptar *agentes*, pero por el motivo que sea, aun no deben ser procesados. El bloque *Hold* no contiene *agentes* en su interior, y puede ser tratado como una extensión de un puerto (o puertos) de entrada o de salida del bloque de modelado de procesos al que está conectado.

Por otra parte, el estado del bloque se puede controlar programáticamente llamando a la función `block()` para bloquear el flujo de *agentes*, o llamando a la función `unblock()` para desbloquear el flujo de *agentes*. Esta característica hace a este bloque de modelado de procesos muy interesante, ya que ofrece la posibilidad de poder interactuar con el modelado de procesos de forma programática.

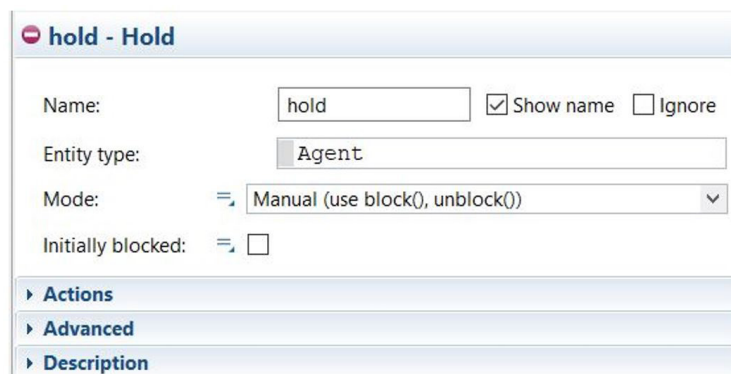


Figura 143. Un bloque de modelado de procesos *hold* definido en Anylogic.

1.19. MOVE TO

El bloque de modelado de procesos *MoveTo* mueve al *agente* entrante a una nueva ubicación dentro de una *Network* (ver **Node** Figura 144). En el caso en que haya recursos asignados al *agente*, los *agentes* se moverán con ellos; en este supuesto, la velocidad de movimiento será la velocidad del *agente* independientemente de la velocidad de los recursos adjuntos.

El tiempo que pasa el *agente* en este bloque es igual a la longitud de la ruta más corta desde la ubicación actual del *agente* hasta la ubicación de destino, dividida por la velocidad del *agente*. Por otra parte, cabe añadir que el *agente* se anima moviéndose a lo largo de la ruta, lo que hace que esta herramienta sea muy interesante para estudiar cómo se desplazan los *agentes* a lo largo de un espacio 2D una vez se ha diseñado un Lay-Out.

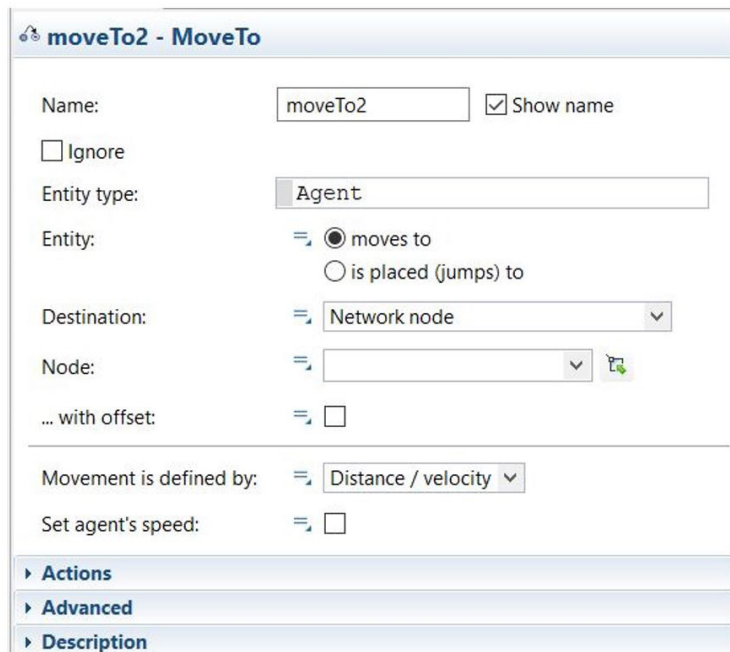
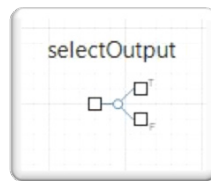


Figura 144. Un bloque de modelado de procesos MoveTo definido en Anylogic.

1.20. SELECTOUTPUT



El bloque de modelado de procesos *SelectOutput* Dirige los *agentes* entrantes a uno de los dos puertos de salida dependiendo de la condición (probabilística o determinista) que se haya configurado en el bloque (ver **Probability** Figura 145).

Este bloque puede utilizarse para clasificar los agentes según ciertos criterios, para dividir aleatoriamente el flujo de *agentes*, etc. Por ejemplo, se puede utilizar para modelar desviaciones en la producción, haciendo que un porcentaje de los *agentes* que evalúa este bloque no continúen en el modelado de procesos.

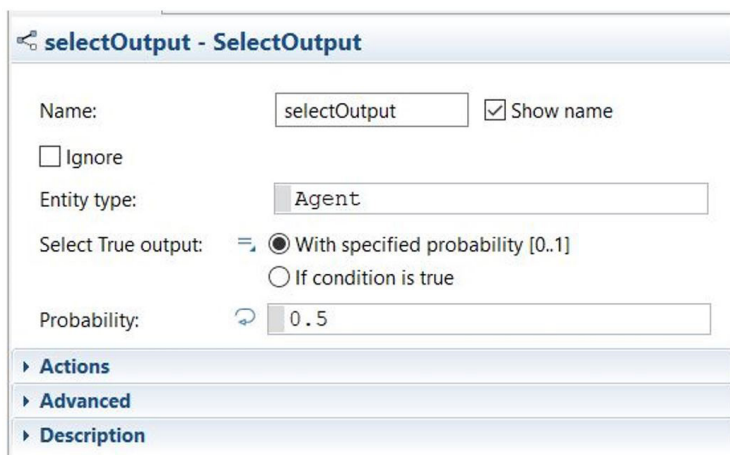
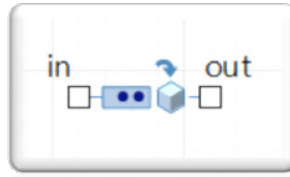


Figura 145. Un bloque de modelado de procesos *SelectOutput* definido en Anylogic.

1.21. BATCH



El bloque de modelado de procesos *Batch* convierte un número de *agentes* en una sola entidad (lote), ya sea desechando los *agentes* originales y creando uno nuevo (cuyas propiedades posiblemente dependen de los originales); o añadiendo los *agentes* originales al contenido de la nueva entidad con la posibilidad de poder desagregarse más tarde en un bloque de modelado de procesos del tipo *Unbatch*.

Este bloque acumula *agentes* dentro de su *queue* interna; y una vez el número de los *agentes* almacenados alcanza el valor especificado (ver **Batch Size** Figura 146); éste se crea el nuevo *agente* (ver **New batch** Figura 146) y lo lanza al siguiente bloque de modelado de procesos.

Figura 146. Un bloque de modelado de procesos Batch definido en Anylogic.

