



Universidad de Valladolid



ESCUELA DE INGENIERÍAS  
INDUSTRIALES

Máster en Ingeniería Industrial

# **MASTER EN INGENIERÍA INDUSTRIAL**

## **ESCUELA DE INGENIERÍAS INDUSTRIALES**

### **UNIVERSIDAD DE VALLADOLID**

## **TRABAJO FIN DE MÁSTER**

**Desarrollo de modelos de simulación Witness orientados a favorecer el aprendizaje del Lean Manufacturing en asignaturas de grado y máster en la EII**

Autor: D. Ester Bausela Martínez  
Tutor: D. Juan José De Benito Martín  
Co-Tutor: D. Pedro Sanz Angulo

Valladolid, Abril, 2017





Universidad de Valladolid



ESCUELA DE INGENIERÍAS  
INDUSTRIALES

Máster en Ingeniería Industrial

**MASTER EN INGENIERÍA INDUSTRIAL**  
ESCUELA DE INGENIERÍAS INDUSTRIALES  
UNIVERSIDAD DE VALLADOLID

**TRABAJO FIN DE MÁSTER**

**Desarrollo de modelos de simulación Witness orientados a favorecer el aprendizaje del Lean Manufacturing en asignaturas de grado y máster en la EII**

Autor: D. Ester Bausela Martínez  
Tutor: D. Juan José De Benito Martín  
Co-Tutor: D. Pedro Sanz Angulo

Valladolid, Abril, 2017



## *RESUMEN*

La finalidad de este Trabajo Fin de Máster es, fundamentalmente, educacional. En concreto, el objetivo consiste en generar modelos de simulación 3D que permitan visualizar fácilmente las ventajas de introducir algunas de las técnicas y herramientas que promueve la filosofía Lean Manufacturing; de este modo, contribuimos a lograr una formación más visual y práctica de los futuros egresados. Se han generado varios modelos de procesos productivos, similares a los que existen en cualquier industria, con el fin de que los alumnos sean capaces de identificar los problemas presentes en cada uno de ellos y seleccionar las técnicas Lean que mejor se adapten al problema. También se han generado los modelos que resultan de implantar las mejoras, lo que permite comparar los procesos antes y después, y apreciar más fácilmente el resultado de las herramientas introducidas.

**Palabras clave:** Lean Manufacturing, Witness, simulación, representación en 3D, educativo

## *ABSTRACT*

The purpose of this Master's Final Project is fundamentally educational. In particular, the aim is to generate 3D simulation models, which easily permit visualizing the advantages of introducing some of the Lean Manufacturing techniques and tools. In this way, a contribution towards a more visual and practical training of future graduates can be achieved. Several models of manufacturing processes, similar to those that exist in any industry, have been generated with the aim that students will be able to identify existing problems in each model and to select the lean manufacturing techniques that best fit to the problem. The models resulting from implementing the improvements have also been generated on the original models, which will allow to compare the models before and after, easily appreciating the result of the introduction and use of the Lean tools.

**Key words:** Lean Manufacturing, Witness, simulation, 3D representation, educational



# *AGRADECIMIENTOS*

A mis tutores, Juan José y Pedro, por el apoyo y ayuda recibidos durante la elaboración de este Trabajo Fin de Máster

A mi familia, por haber estado a mi lado en todo momento, por su apoyo incondicional y por sus sabios consejos.

A mis amigos, gracias por todos los momentos que me habéis reglado durante estos años, gracias por convertirlos en un apoyo fundamental.

“Caminante, son tus huellas  
en el camino y nada más;  
Caminante, no hay camino  
se hace camino al andar”

*Antonio Machado*



# ÍNDICE

<b>INTRODUCCIÓN</b> .....	<b>1</b>
1. ANTECEDENTES Y MOTIVACIÓN.....	3
2. OBJETIVOS Y PLANTEAMIENTO DEL TRABAJO .....	3
3. DESARROLLO DEL TRABAJO .....	4
<b>Capítulo 1.: ESTADO DEL ARTE</b> .....	<b>7</b>
1. LEAN MANUFACTURING .....	9
1.1. INTRODUCCIÓN A LEAN MANUFACTURING.....	9
1.2. TÉCNICAS DE LEAN MANUFACTURING.....	10
1.3. LEAN MANUFACTURING EN EL ENTORNO EDUCACIONAL.....	12
2. WITNESS.....	12
2.1. INTRODUCCIÓN A WITNESS .....	12
2.2. DESCRIPCIÓN DEL PROGRAMA .....	12
2.3. CONSIDERACIONES AL REALIZAR EL MODELO EN 3D.....	14
<b>Capítulo 2.: DESCRIPCIÓN DE LOS MODELOS I</b> .....	<b>17</b>
1. TÉCNICA KANBAN.....	19
1.1. DESCRIPCIÓN DEL MODELO SIN TÉCNICA KANBAN .....	19
1.2. IMPLEMENTACIÓN DEL KANBAN .....	23
2. TEORÍA DE LAS LIMITACIONES .....	30
2.1. DESCRIPCIÓN DEL MODELO CON CUELLO DE BOTELLA .....	30
2.2. IMPLEMENTACIÓN DE TÉCNICA LEAN (CAPACITARIO).....	35
<b>Capítulo 3.: DESCRIPCIÓN DE LOS MODELOS II</b> .....	<b>39</b>
1. TÉCNICA POKA YOKE .....	41
1.1. DESCRIPCIÓN DEL MODELO SIN POKA YOKE.....	41
1.2. IMPLEMENTACIÓN DE POKA YOKE DE ADVERTENCIA .....	46
1.3. IMPLEMENTACIÓN DE POKA YOKE DE FIABILIDAD .....	50
2. TÉCNICA CÉLULA EN U (OPERARIO ACOMPAÑANDO A PIEZA) .....	52
2.1. DESCRIPCIÓN DEL MODELO CON DISPOSICIÓN LINEAL .....	52
2.2. IMPLEMENTACIÓN DE LA CÉLULA EN U .....	57
3. TÉCNICA HEIJUNKA .....	61
3.1. DESCRIPCIÓN DEL MODELO .....	61
3.2. IMPLEMENTACIÓN DE LA TÉCNICA HEIJUNKA .....	67
<b>Capítulo 4.: COMPORTAMIENTO DE LOS MODELOS</b> .....	<b>71</b>
1. TÉCNICA KANBAN.....	73
1.1. MODELO SIN KANBAN.....	73
1.2. MODELO CON KANBAN.....	76
1.3. COMPARACIÓN DE LOS MODELOS.....	79
2. TEORÍA DE LAS LIMITACIONES .....	81
2.1. MODELO CON CUELLO DE BOTELLA.....	82
2.2. MODELO CON ELEMENTO CAPACITARIO .....	84
2.3. COMPARACIÓN DE LOS MODELOS.....	85
3. TÉCNICA POKA YOKE .....	86
3.1. MODELO SIN POKA YOKE, CON POKA YOKE DE ADVERTENCIA Y CON POKA YOKE DE FIABILIDAD .....	86
3.2. COMPARACIÓN DE LOS MODELOS.....	90

4.	<i>TÉCNICA CÉLULA EN U</i> .....	95
4.1.	MODELO CON DISPOSICIÓN LINEAL.....	95
4.2.	MODELO CON CÉLULA EN U.....	97
4.3.	COMPARACIÓN DE RESULTADOS.....	99
5.	<i>TÉCNICA HEIJUNKA</i> .....	101
5.1.	MODELO SIN HEIJUNKA NI SMED.....	102
5.2.	MODELO CON HEIJUNKA Y SMED .....	105
<b>CONCLUSIONES Y LÍNEAS FUTURAS.....</b>		<b>109</b>
<b>REFERENCIAS.....</b>		<b>115</b>

# ÍNDICE DE FIGURAS

Figura 1.1: elementos que se pueden simular en Witness .....	13
Figura 1.2: ventana de simulación de Witness .....	14
Figura 1.3: representación en 2D de conveyor.....	15
Figura 1.4: representación en 2D de path .....	15
Figura 1.5: representación en 2D de track .....	15
Figura 1.6: representación en 3D de conveyor.....	15
Figura 1.7: representación en 3D de path .....	15
Figura 1.8: representación en 3D de track .....	15
Figura 2.1: representación del modelo sin kanban .....	19
Figura 2.2: representación de la pieza 2D y 3D modelos técnica kanban .....	20
Figura 2.3: entrada de piezas en el instante inicial en el modelo sin kanban .....	20
Figura 2.4: regla entrada contenedores fabricación en el modelo sin kanban.....	20
Figura 2.5: regla de distribución de piezas a contenedores fabricación en el modelo sin kanban .....	21
Figura 2.6: inicialización contador contenedores fabricación en el modelo sin kanban.....	21
Figura 2.7: regla de entrada para ficticia_ensamblar en el modelo sin kanban .....	22
Figura 2.8: regla de salida vehiculo modelo sin kanban .....	22
Figura 2.9: regla de colocación las piezas en los contenedores de acabado en el modelo sin kanban .....	22
Figura 2.10: regla de salida de piezas de los contenedores de almacenaje zona acabado modelo sin kanban ...	23
Figura 2.11: representación del modelo con kanban.....	24
Figura 2.12: descripción de la zona de acabado modelo con kanban .....	24
Figura 2.13: regla de entrada final_buzón_acción modelo con kanban.....	25
Figura 2.14: regla de entrada ficticia_unir_kanban en el modelo con kanban.....	25
Figura 2.15: acciones a la salida ficticia_unir_kanban en el modelo con kanban.....	25
Figura 2.16: regla de salida salida_ficticia en modelo con kanban.....	26
Figura 2.17: regla de entrada ficticia_separar_piezas en modelo con kanban.....	26
Figura 2.18: zona de fabricación modelo con kanban .....	26
Figura 2.19: definición de necesidad_fabricar (1) en modelo con kanban.....	27
Figura 2.20: inicialización de la variable necesidad_fabricar en modelo con kanban.....	27
Figura 2.21: regla entrada máquina ficticia hacia_fabrica en modelo con kanban .....	27
Figura 2.22: regla salida máquina ficticia hacia_fabrica en modelo con kanban .....	27
Figura 2.23: regla de entrada de fabricacion3 en el modelo con kanban .....	27
Figura 2.24: orden de producción en fabricacion2 en el modelo con kanban.....	28
Figura 2.25: regla de salida máquina ficticia_fabri_ensa modelo con kanban.....	28
Figura 2.26: parada de fabricacion3 en acción de entrada salida_prod modelo con kanban .....	28
Figura 2.27: acción a la entrada trans1_2 modelo con kanban .....	28
Figura 2.28: regla de entrada fabricacion2 modelo con kanban.....	29
Figura 2.29: regla de salida fabricacion2 modelo con kanban.....	29
Figura 2.30: regla de entrada hacia_buzon modelo con kanban.....	29
Figura 2.31: regla de entrada en el momento inicial para trans1_2 en el modelo con kanban.....	29
Figura 2.32: representación del modelo con cuello de botella.....	30
Figura 2.33: representación 3D pieza1 modelos teoría de las limitaciones.....	30
Figura 2.34: representación 3D pieza3 modelos teoría de las limitaciones.....	30
Figura 2.35: representación 3D pieza2 modelos teoría de las limitaciones.....	30
Figura 2.36: regla de entrada de piezas al proceso modelo con cuello de botella .....	31
Figura 2.37: definición de path en averías en los modelos teoría de las limitaciones .....	32
Figura 2.38: recorrido operario entre máquinas consecutivas modelos teoría de las limitaciones .....	32
Figura 2.39: recorrido operario entre máquinas en paralelo modelos teoría de las limitaciones .....	32
Figura 2.40: salida de piezas cinta1_2 modelo con cuello de botella.....	32
Figura 2.41: paths necesarios en la operación de sacar piezas del cuello de botella en el modelo con cuello de botella.....	33
Figura 2.42: descripción del path cint1_2_maq2 en el modelo con cuello de botella.....	33
Figura 2.43: descripción del path cint1_2_cuell en el modelo con cuello de botella.....	33
Figura 2.44: descripción del path cuello_cont en el modelo con cuello de botella.....	33
Figura 2.45: descripción del path sal_cuello_2 en el modelo con cuello de botella.....	33
Figura 2.46: ensamblaje en los modelos teoría de las limitaciones.....	34
Figura 2.47: salida ensamblaje modelos teoría de las limitaciones.....	34
Figura 2.48: salida de piezas del proceso en los modelos teoría de las limitaciones.....	34
Figura 2.49: acción sacar piezas de los modelos teoría de las limitaciones.....	34
Figura 2.50: regla salida de piezas del proceso modelos teoría de las limitaciones.....	34
Figura 2.51: necesidad de sacar piezas del contenedor de salida del proceso modelos teoría de las limitaciones.....	35
Figura 2.52: acción de ensamblar piezas para el transporte en el modelos teoría de las limitaciones.....	35
Figura 2.53: representación del modelo capacitario.....	36
Figura 2.54: regla de salida de cinta1_2 en el modelo capacitario.....	36

Figura 2.55: regla de entrada a la maq1 modelo capacitario.....	36
Figura 2.56: regla de entrada de la maquina entr_proce en modelo capacitario.....	37
Figura 3.1: representación del modelo sin poka yoke .....	41
Figura 3.2: definición atributos tipo0 modelo sin poka yoke .....	42
Figura 3.3: representación 3D de la pieza tipo0 en los modelos técnica poka yoke.....	42
Figura 3.4: representación 3D de la pieza tipo1 en los modelos técnica poka yoke.....	42
Figura 3.5: acción a la salida de la maquina1 en el modelo sin poka yoke .....	43
Figura 3.6: representación de las piezas sin la calidad mínima en los modelos técnica poka yoke.....	43
Figura 3.7: regla de entrada en la máquina ficticia_ensamble en los modelos técnica poka yoke.....	44
Figura 3.8: regla de salida de la máquina ficticia_ensamble en los modelos técnica poka yoke.....	44
Figura 3.9: representación del ensamblaje incorrecto en los modelos técnica poka yoke .....	44
Figura 3.10: disposición de las piezas en los procesos independientes modelo sin poka yoke .....	45
Figura 3.11: representación de la zona específica en el modelo sin poka yoke.....	45
Figura 3.12: salida de control_tipo0 en los técnica modelos poka yoke .....	45
Figura 3.13: caso de sacar piezas de malas_tipo0 modelos técnica poka yoke .....	46
Figura 3.14: necesidad de sacar piezas de almacen_malas_0 modelos técnica poka yoke.....	46
Figura 3.15: necesidad de ensamblar piezas a la salida de la piezas_tipo0_calidad en los modelos técnica poka yoke.....	46
Figura 3.16: regla de entrada de sal0, salida de piezas de la zona0 en los modelos técnica poka yoke .....	46
Figura 3.17: representación del modelo poka yoke de advertencia .....	47
Figura 3.18: definición del atributo tipo_acople en modelo con poka yoke de advertencia .....	48
Figura 3.19: representación en 3D de cont_ens0 modelo poka yoke advertencia .....	48
Figura 3.20: representación en 3D de cont_ens1 modelo poka yoke advertencia .....	48
Figura 3.21: poka yoke en la salida de maquina4 modelo poka yoke advertencia.....	48
Figura 3.22: salida de piezas de cinta desecho modelo poka yoke advertencia .....	49
Figura 3.23: salida de piezas del contenedor chatarra_comun modelo poka yoke advertencia.....	49
Figura 3.24: disposición de las piezas en la maquina5 en modelo poka yoke advertencia.....	49
Figura 3.25: poka yoke entrada zona específica tipo1 en modelo poka yoke advertencia .....	49
Figura 3.26: separación de piezas incorrectas tipo modelo poka yoke advertencia.....	50
Figura 3.27: variable contador de chatarra_tipo_0 modelo poka yoke advertencia.....	50
Figura 3.28: representación del modelo poka yoke de fiabilidad.....	50
Figura 3.29: definición de atributo tipo_acople en modelo poka yoke de fiabilidad.....	51
Figura 3.30: clasificación de las piezas en modelo con poka yoke de fiabilidad .....	51
Figura 3.31: representación del modelo acompañar pieza lineal.....	52
Figura 3.32: máquinas pertenecientes a la zona1 modelo acompañar pieza lineal.....	52
Figura 3.33: máquinas pertenecientes a la zona2 modelo acompañar pieza lineal.....	52
Figura 3.34: máquinas pertenecientes a la zona3 modelo acompañar pieza lineal.....	53
Figura 3.35: salida del proceso modelo acompañar pieza lineal.....	53
Figura 3.36: representación de la pieza_prin modelos acompañar pieza.....	53
Figura 3.37: representación de la pieza_sec modelos acompañar pieza .....	53
Figura 3.38: representación del conjunto modelos acompañar pieza.....	53
Figura 3.39: llegada de pieza_prin modelos acompañar pieza .....	54
Figura 3.40: llegadas de pieza_sec modelos acompañar pieza .....	54
Figura 3.41: descripción de la actividad entrada_proceso modelo acompañar pieza lineal.....	55
Figura 3.42: descripción del path cont_pieza_entrada en modelo acompañar pieza lineal.....	55
Figura 3.43: definición del tiempo de preparación de la actividad entrada_proceso en modelo acompañar pieza lineal.....	55
Figura 3.44: definición de operario en entrada_proceso modelo acompañar pieza lineal.....	55
Figura 3.45: cambio de valor operacion_op1 modelo acompañar pieza lineal.....	55
Figura 3.46: entrada de piezas en la maquina1 en el modelo acompañar pieza lineal .....	56
Figura 3.47: definición del nuevo ciclo a realiza por el op_ent_1_2_3 en modelo acompañar pieza lineal.....	56
Figura 3.48: descripción de la entrada a la maquina4 en modelo acompañar pieza lineal.....	56
Figura 3.49: definición de maquina7 en modelo acompañar pieza lineal .....	56
Figura 3.50: descripción de la maquina8 en el modelo acompañar pieza lineal.....	56
Figura 3.51: salida de piezas de los modelos acompañar pieza .....	57
Figura 3.52: representación del modelo acompañar pieza en U .....	58
Figura 3.53: entrada de piezas en maquina2 modelo acompañar pieza en U.....	59
Figura 3.54: definición de próximo valor de la operacion_op2 en modelo acompañar pieza U.....	59
Figura 3.55: regla de entrada maquina4 en modelo acompañar pieza U.....	59
Figura 3.56: definición de entrada de piezas maquina6 modelo acompañar piezas U.....	60
Figura 3.57: variación variable operacion_op2 salida maquina6 modelo acompañar pieza U.....	60
Figura 3.58: entrada de piezas en la maquina8 modelo acompañar pieza U.....	60
Figura 3.59: variación de la variable operacion_op3 en la maquina7 modelo acompañar pieza U .....	60
Figura 3.60: fin de ciclo zona3 modelo acompañar pieza U.....	60
Figura 3.61: acción de salida maquina9 modelo acompañar pieza U .....	60
Figura 3.62: acción a la entrada de la maquina10 modelo acompañar pieza U.....	61
Figura 3.63: regla de entrada de la maquina10 modelo acompañar pieza U .....	61
Figura 3.64: acción a la salida maquina10 modelo acompañar pieza U.....	61

Figura 3.65: representación del modelo técnicas Heijunka y Smed .....	62
Figura 3.66: representación de la pieza1 modelos técnica heijunka .....	63
Figura 3.67: representación de la pieza2 modelos técnica heijunka .....	63
Figura 3.68: representación de la pieza3 modelos técnica heijunka .....	63
Figura 3.69: regla de entrada al modelo sin heijunka ni smed .....	63
Figura 3.70: incremento contador entrada modelo sin heijunka ni smed.....	63
Figura 3.71: inicialización de contadores entrada modelo sin heijunka ni smed .....	63
Figura 3.72: operación cambio de pieza maq1 en modelo sin heijunka ni smed.....	63
Figura 3.73: salida de piezas en los contenedores en el modelo heijunka.....	64
Figura 3.74: funcionamiento del clasificar modelos técnica heijunka .....	64
Figura 3.75: representación lote_cliente1 modelo heijunka.....	65
Figura 3.76: representación lote_cliente2 modelo heijunka.....	65
Figura 3.77: representación lote_cliente3 modelo heijunka.....	65
Figura 3.78: primer tipo a realizar por unión modelos técnica heijunka.....	65
Figura 3.79: regla de entrada en el modelo sin heijunka ni smed.....	66
Figura 3.80: acciones de entrada en el modelo sin heijunka ni smed .....	66
Figura 3.81: acción previa cambio de tipo de pack en unión en el modelo sin heijunka ni smed .....	66
Figura 3.82: acciones al finalizar la acción unir en modelo sin heijunka ni smed .....	66
Figura 3.83: disposición de los lotes en su contenedor correspondiente en modelo heijunka .....	67
Figura 3.84: disposición de los lotes en su contenedor correspondiente en modelo heijunka .....	67
Figura 3.85: regla de entrada al modelo con heijunka y smed .....	68
Figura 3.86: inicialización de contadores entrada modelo con heijunka y smed .....	68
Figura 3.87: operación cambio de pieza maq2 en modelo con heijunka y smed .....	68
Figura 3.88: acción previa cambio de tipo de conjunto en unión en el modelo con heijunka y smed.....	68
Figura 3.89: acción a la entrada modelo con heijunka y smed.....	68
Figura 3.90: acción a la salida de la máquina unión modelo con heijunka y smed.....	69
Figura 4.1: representación del modelo 3D sin kanban .....	73
Figura 4.2: representación del modelo 3D sin kanban en el momento inicial.....	73
Figura 4.3: representación del modelo 3D sin kanban, llenado de fabr1 .....	74
Figura 4.4: representación del modelo 3D sin kanban, transporte de la pieza desde zona de fabricación a la de acabado.....	74
Figura 4.5: representación del modelo 3D sin kanban, almacenaje de las piezas en alm1.....	74
Figura 4.6: representación del modelo 3D sin kanban, colocación de las piezas en la entrada al proceso acabado .....	74
Figura 4.7: representación del modelo 3D sin kanban, llenado de los contenedores de acabado.....	75
Figura 4.8: representación del modelo 3D sin kanban, llenado de la cinta de entrada a la zona de acabado .....	75
Figura 4.9: representación del modelo 3D sin kanban, transporte de 5 piezas estando los contenedores de acabado llenos .....	75
Figura 4.10: representación del modelo 3D sin kanban, llenado de los contenedores de la zona de acabado.....	76
Figura 4.11: representación del modelo 3D sin kanban, salida de piezas del proceso .....	76
Figura 4.12: representación del modelo 3D con kanban .....	76
Figura 4.13: representación del modelo 3D con kanban, transporte de kanban .....	77
Figura 4.14: representación del modelo 3D con kanban, comienzo de fabricación en el momento inicial .....	77
Figura 4.15: representación del modelo 3D con kanban, contenedor fabr1 con 5 piezas .....	77
Figura 4.16: representación del modelo 3D con kanban, unión de las 5 piezas con k_transporte1 .....	78
Figura 4.17: representación del modelo 3D con kanban, llenado del contenedor alm1 junto con k_transporte1 .....	78
Figura 4.18: representación del modelo 3D con kanban, llenado del cinta entrada y recogida k_transporte1 .....	78
Figura 4.19: representación del modelo 3D con kanban, sin necesidades de transporte pero sí de fabricación ..	78
Figura 4.20: representación del modelo 3D con kanban, k_fabrica hacia buzón_almacen .....	79
Figura 4.21: representación del modelo 3D con kanban, sin necesidades de fabricación ni de transporte.....	79
Figura 4.22: representación del modelo 3D con kanban, tareas mantenimiento con movimiento entre paths .....	79
Figura 4.23: representación del modelo 3D sin kanban en el instante 1000 .....	80
Figura 4.24: representación del modelo 3D con kanban en el instante 1000 .....	80
Figura 4.25: representación del modelo 3D sin kanban en el instante 2000 .....	80
Figura 4.26: representación del modelo 3D con kanban en el instante 2000 .....	81
Figura 4.27: representación del modelo 3D sin kanban en el instante 2500 .....	81
Figura 4.28: representación del modelo 3D con kanban en el instante 2500 .....	81
Figura 4.29: representación del modelo 3D con cuello de botella .....	82
Figura 4.30: entrada piezas al modelo 3D con cuello de botella .....	82
Figura 4.31: operario dirigiéndose a sacar piezas cinta1_2 modelo 3D con cuello de botella.....	83
Figura 4.32: operario sacando piezas de la cinta1_2 cuello de botella, modelo 3D cuello de botella .....	83
Figura 4.33: transporte de las piezas por el path cuello_cont hacia el contenedor sal_cuello, modelo 3D cuello de botella.....	83
Figura 4.34: operario trasladándose a la cinta1_2 desde el cuello de botella, modelo 3D cuello de botella .....	83
Figura 4.35: operario trasladándose otra máquina desde el cuello de botella, modelo 3D cuello de botella.....	83
Figura 4.36: operario sacando piezas de la cinta salida_proceso, modelo 3D cuello de botella .....	83
Figura 4.37: representación del modelo 3D capacitario .....	84
Figura 4.38: entrada de piezas en el modelo 3D capacitario.....	84

Figura 4.39: parada de la maq1 con stock necesario en cinta1_2 en el modelo 3D capacitario .....	84
Figura 4.40: finalización de la actividad de introducir piezas en el modelo 3D capacitario.....	85
Figura 4.41: visualización del modelo 3D cuello de botella en el instante 1000 .....	85
Figura 4.42: visualización del modelo 3D con elemento capacitario en el instante 1000 .....	85
Figura 4.43: visualización del modelo 3D cuello de botella en el instante 2000 .....	86
Figura 4.44: visualización del modelo 3D con elemento capacitario en el instante 2000 .....	86
Figura 4.45: representación del modelo 3D sin poka yoke.....	86
Figura 4.46: representación del modelo 3D sin poka yoke, pieza antes de defecto de calidad.....	87
Figura 4.47: representación del modelo 3D sin poka yoke, pieza con defecto de calidad.....	87
Figura 4.48: representación del modelo 3D sin poka yoke, ensamblaje tipo0.....	87
Figura 4.49: representación del modelo 3D sin poka yoke, conjunto ensamblaje tipo0 .....	87
Figura 4.50: representación del modelo 3D sin poka yoke, pieza defecto calidad antes de ensamblaje .....	87
Figura 4.51: representación del modelo 3D sin poka yoke, pieza defecto calidad después de ensamblaje incorrecto .....	87
Figura 4.52: representación del modelo 3D sin poka yoke, ensamblaje tipo1.....	87
Figura 4.53: representación del modelo 3D sin poka yoke, conjunto ensamblaje tipo1 .....	87
Figura 4.54: representación del modelo 3D sin poka yoke, ensamblaje incorrecto .....	88
Figura 4.55: representación del modelo 3D sin poka yoke, representación del conjunto incorrecto .....	88
Figura 4.56: representación del modelo 3D sin poka yoke, colocación correcta .....	88
Figura 4.57: representación del modelo 3D sin poka yoke, colocación de forma incorrecta .....	88
Figura 4.58: representación del modelo 3D sin poka yoke, pieza con defecto de calidad antes de entrar en control .....	88
Figura 4.59: representación del modelo 3D sin poka yoke, pieza con defecto de calidad después de entrar en control.....	88
Figura 4.60: representación del modelo 3D con poka yoke advertencia .....	89
Figura 4.61: representación del modelo 3D con poka yoke advertencia, control de calidad zona común.....	89
Figura 4.62: representación del modelo 3D con poka yoke advertencia, control de tipo, ensamblaje y calidad en la zona específica.....	89
Figura 4.63: representación del modelo 3D con poka yoke de fiabilidad.....	90
Figura 4.64: representación del modelo 3D sin poka yoke, instante 217 .....	90
Figura 4.65: representación del modelo 3D con poka yoke de advertencia, instante 217.....	90
Figura 4.66: representación del modelo 3D con poka yoke de fiabilidad, instante 217 .....	90
Figura 4.67: representación del modelo 3D sin poka yoke, instante 387 .....	91
Figura 4.68: representación del modelo 3D con poka yoke de advertencia, instante 387.....	91
Figura 4.69: representación del modelo 3D con poka yoke de fiabilidad, instante 387 .....	91
Figura 4.70: representación del modelo 3D sin poka yoke, instante 302 .....	91
Figura 4.71: representación del modelo 3D poka yoke de advertencia, instante 302.....	91
Figura 4.72: representación del modelo 3D con poka yoke de fiabilidad, instante 302.....	91
Figura 4.73: representación del modelo 3D sin poka yoke, instante 556 .....	92
Figura 4.74: representación del modelo 3D poka yoke de advertencia, instante 556.....	92
Figura 4.75: representación del modelo 3D con poka yoke de fiabilidad, instante 556 .....	92
Figura 4.76: representación del modelo 3D sin poka yoke, instante 1500 .....	92
Figura 4.77: representación del modelo 3D con poka yoke de advertencia, instante 1500.....	92
Figura 4.78: representación del modelo 3D con poka yoke de fiabilidad, instante 1500 .....	93
Figura 4.79: representación del modelo 3D sin poka yoke, instante 2500 .....	93
Figura 4.80: representación del modelo 3D con poka yoke de advertencia, instante 2500.....	93
Figura 4.81: representación del modelo 3D con poka yoke de fiabilidad, instante 2500 .....	94
Figura 4.82: representación del modelo 3D sin poka yoke, instante 3500 .....	94
Figura 4.83: representación del modelo 3D con poka yoke de advertencia, instante 3500.....	94
Figura 4.84: representación del modelo 3D con poka yoke de fiabilidad, instante 3500 .....	95
Figura 4.85: visualización del modelo 3D acompañar pieza lineal .....	95
Figura 4.86: visualización del instante inicial modelo 3D acompañar pieza lineal.....	96
Figura 4.87: visualización del traslado pieza hacia maquina1 y hacia maquina9 instante 70 modelo 3D acompañar pieza lineal.....	96
Figura 4.88: visualización del traslado pieza entre maquina4 y maquina5 instante 130.8 modelo 3D acompañar pieza lineal.....	96
Figura 4.89: visualización del instante 280, finalización de un ciclo en dos zonas, modelo 3D acompañar pieza lineal .....	96
Figura 4.90: visualización del instante 344, desplazamiento a la primera máquina de la zona, modelo 3D acompañar pieza lineal.....	97
Figura 4.91: visualización del instante 376, desplazamiento a la primera máquina zona1 y 2, modelo 3D acompañar pieza lineal.....	97
Figura 4.92: visualización del modelo 3D acompañar pieza en U .....	97
Figura 4.93: visualización del instante inicial modelo 3D acompañar pieza en U.....	98
Figura 4.94: visualización del instante 61 modelo 3D acompañar pieza en U.....	98
Figura 4.95: visualización del instante 174, comienzo de operación maquina8 modelo 3D acompañar pieza en U .....	98
Figura 4.96: visualización instante 176, operación maquina8 y desplazamiento hacia maquina3 modelo 3D acompañar pieza en U .....	99
Figura 4.97: visualización del instante 1097.6, comienzo de la zona1 y 3 modelo 3D acompañar pieza en U ....	99

<i>Figura 4.98: visualización del instante 2000, modelo 3D acompañar pieza lineal</i> .....	100
<i>Figura 4.99: visualización del instante 2000, modelo 3D acompañar pieza en U</i> .....	100
<i>Figura 4.100: visualización del instante 4000, modelo 3D acompañar pieza lineal</i> .....	100
<i>Figura 4.101: visualización del instante 4000, modelo 3D acompañar pieza en U</i> .....	100
<i>Figura 4.102: visualización del instante 10000, modelo 3D acompañar pieza lineal</i> .....	100
<i>Figura 4.103: visualización del instante 10000, modelo 3D acompañar pieza en U</i> .....	101
<i>Figura 4.104: visualización del instante 20000, modelo 3D acompañar pieza lineal</i> .....	101
<i>Figura 4.105: visualización del instante 20000, modelo acompañar pieza en U</i> .....	101
<i>Figura 4.106: visualización del modelo 3D técnica heijunka y smed</i> .....	102
<i>Figura 4.107: lotes de piezas en trans1 en el modelo 3D sin heijunka ni smed</i> .....	102
<i>Figura 4.108: acción en maq1 debida al cambio del tipo de pieza en el instante 324, modelo 3D sin heijunka ni smed</i> .....	102
<i>Figura 4.109: primer pieza2 salida de la maq1 instante 392, modelo 3D sin heijunka ni smed</i> .....	102
<i>Figura 4.110: acción en maq2 debida al cambio del tipo de pieza en el instante 400, modelo 3D sin heijunka ni smed</i> .....	102
<i>Figura 4.111: entrada en zona específica a cada pieza en lote de 3 piezas, modelo 3D sin heijunka ni smed</i> ...	103
<i>Figura 4.112: piezas necesarias para el ensamblaje cliente1, modelo 3D sin heijunka ni smed</i> .....	103
<i>Figura 4.113: visualización del primer conjunto cliente1 instante 985 modelo 3D sin heijunka ni smed</i> .....	103
<i>Figura 4.114: visualización del último conjunto para el cliente1 en el instante 1705, modelo 3D sin heijunka ni smed</i> .....	104
<i>Figura 4.115: acción asociada al cambio de client1 a cliente2 en unión, modelo 3D sin heijunka ni smed</i> .....	104
<i>Figura 4.116: visualización del primer conjunto cliente2 en el instante 1776, modelo 3D sin heijunka ni smed</i> ..	104
<i>Figura 4.117: visualización del último conjunto cliente2 y piezas cliente3, modelo 3D sin heijunka ni smed</i> .....	104
<i>Figura 4.118: visualización del último conjunto cliente2 y primer conjunto cliente3, modelo 3D sin heijunka ni smed</i> .....	105
<i>Figura 4.119: visualización del último conjunto cliente3 y comienzo conjunto cliente1, modelo 3D sin heijunka ni smed</i> .....	105
<i>Figura 4.120: visualización de los lotes que entran en la cinta trans1, modelo 3D con heijunka y smed</i> .....	106
<i>Figura 4.121 acción asociada al cambio tipo en maq1 instante 189, modelo 3D con heijunka y smed</i> .....	106
<i>Figura 4.122: primera pieza2 después cambio de tipo instante 203.5, modelo 3D con heijunka y smed</i> .....	106
<i>Figura 4.123: visualización del primer conjunto a realizar del cliente1, modelo 3D con heijunka y smed</i> .....	106
<i>Figura 4.124: visualización de 4 conjuntos del primer lote del cliente1, modelo 3D con heijunka y smed</i> .....	106
<i>Figura 4.125: acción previa en la máquina unión para la realización del segundo lote instante 820.7, modelo 3D con heijunka y smed</i> .....	107
<i>Figura 4.126: primer conjunto del primer lote cliente2 instante 856, modelo 3D con heijunka y smed</i> .....	107
<i>Figura 4.127: último conjunto para el cliente2 en el primer lote, modelo 3D con heijunka y smed</i> .....	107
<i>Figura 4.128: primer conjunto del primer lote cliente3, modelo 3D con heijunka y smed</i> .....	107
<i>Figura 4.129: último conjunto para el cliente3 en el primer lote, modelo 3D con heijunka y smed</i> .....	108

# ÍNDICE DE TABLAS

<i>Tabla 2.1: tiempo de ciclo las máquinas modelos técnica kanban .....</i>	<i>20</i>
<i>Tabla 2.2: tiempos de ciclo así como prioridad en los modelos teoría de las limitaciones .....</i>	<i>31</i>
<i>Tabla 3.1: tiempo de ciclo y calidad mínima zona común modelo sin poka yoke .....</i>	<i>43</i>
<i>Tabla 3.2: tiempo de ciclo y calidad mínima zona específica modelo sin poka yoke .....</i>	<i>43</i>
<i>Tabla 3.3: tiempo de ciclo de las máquinas en el modelo poka yoke de advertencia .....</i>	<i>47</i>
<i>Tabla 3.4: tiempos de ciclo modelo poka yoke de fiabilidad .....</i>	<i>51</i>
<i>Tabla 3.5: valores que toman las variables que condicionan los movimientos de los operarios modelo acompañar pieza lineal .....</i>	<i>54</i>
<i>Tabla 3.6: valores que toman las variables que condicionan los movimientos de los operarios modelo acompañar pieza en U .....</i>	<i>58</i>
<i>Tabla 3.7: necesidades de los clientes modelos técnica Heijunka y Smed .....</i>	<i>62</i>
<i>Tabla 3.8: definición de entrada de piezas al modelo sin heijunka ni smed .....</i>	<i>62</i>
<i>Tabla 3.9: valor de las variables a la salida de los contenedores almacenaje modelo heijunka .....</i>	<i>64</i>
<i>Tabla 3.10: definición de entrada de piezas al modelo con heijunka y smed .....</i>	<i>67</i>
<i>Tabla 3.11: definición tamaños lote en union con heijunka y smed .....</i>	<i>67</i>
<i>Tabla 4.1: comparación del número de piezas en modelo kanban a diferentes tiempos .....</i>	<i>80</i>
<i>Tabla 4.2: comparación del número de piezas en modelo cuello de botella a diferentes tiempos .....</i>	<i>85</i>
<i>Tabla 4.3: comparación del número de piezas en modelo poka yoke a diferentes tiempos .....</i>	<i>95</i>
<i>Tabla 4.4: comparación del número de piezas en modelo poka yoke a diferentes tiempos .....</i>	<i>99</i>
<i>Tabla 4.5: tiempos en los que se producen piezas para cada cliente, modelo sin heijunka ni smed .....</i>	<i>105</i>
<i>Tabla 4.6: tiempos en los que se producen piezas para cada cliente, modelo con heijunka y smed .....</i>	<i>108</i>

# INTRODUCCIÓN



## 1. ANTECEDENTES Y MOTIVACIÓN

En la actualidad, las técnicas Lean Manufacturing están presentes en la mayoría de los procesos productivos; obviamente, dependiendo del proceso del que se trate se usarán más unas técnicas que otras. La aplicación de estas técnicas ha hecho que se disminuyan considerablemente los gastos que se generan, ya sea por defectos de calidad, almacenaje de stock innecesario, elevados tiempos de desplazamiento, ... La disminución de gastos hace que las empresas deseen aplicar técnicas Lean, por lo que surge la necesidad de que los estudiantes conozcan la filosofía y las distintas técnicas Lean para desempeñar mejor su trabajo en el futuro.

Como se ha comentado anteriormente, es necesario que los estudiantes conozcan las técnicas de Lean Manufacturing debido a las necesidades del mercado. El estudio de estas técnicas debe tener su base teórica, para conocer cuáles son sus puntos más importantes, pero también es necesario que tenga una base práctica. Sin embargo, esta base práctica es muy poco usada, haciendo que se conozcan los fundamentos teóricos de estas técnicas, pero no tanto los prácticos, que serán los que se les exijan a los estudiantes en su entorno de trabajo, debido al auge de estas técnicas en la mayoría de los procesos productivos.

Además, otro aspecto que es de vital importancia en las empresas es el conocimiento del comportamiento de sus procesos, así como conocer cómo se van a comportar incluso antes de implantarlos. Predecir el comportamiento de un futuro proceso productivo mediante la simulación proporciona una idea muy exacta de cómo se comportará, pudiendo tomar decisiones sin realizar grandes inversiones. Por esta razón, también resulta muy adecuado poseer conocimientos de simulación, para conocer el proceso productivo óptimo, realizando las mejoras oportunas, sin correr grandes riesgos por implantaciones erróneas.

Ambas razones han originado este Trabajo Fin de Máster, en el que se pretende dar a conocer las técnicas Lean Manufacturing mediante un programa de simulación, Witness. De esta forma, se van a poder visualizar los contenidos teóricos de Lean Manufacturing desarrollados en las asignaturas de gestión de la producción de una forma más práctica que una clase magistral. Además, se podrán adquirir conocimientos de simulación que, como se ha mencionado anteriormente, son también muy necesarios a la hora de implementar un proceso productivo o conocer cómo se va a comportar.

Mediante una serie de simulaciones, se van a poder visualizar en 3D estas técnicas, para ser utilizadas en asignaturas impartidas en la Escuela de Ingenierías Industriales de Valladolid, en diferentes titulaciones de Grado (Organización, Mecánica, ...) como de Máster (Ingeniería Industrial, Logística, ...).

## 2. OBJETIVOS Y PLANTEAMIENTO DEL TRABAJO

Las técnicas Lean Manufacturing están muy presentes en muchos procesos productivos. El objetivo principal de este TFM es **crear diferentes modelos de simulación que sirvan de apoyo en el proceso de enseñanza-aprendizaje de las herramientas Lean Manufacturing**. Se busca dar a conocer a los alumnos las técnicas Lean Manufacturing de una forma visual, de forma que descubran sus características más importantes de una forma amena y entretenida.

Para lograr este objetivo, vamos a utilizar una herramienta de simulación, Witness, que permite representar los modelos tanto en 2D como en 3D. En consecuencia, para alcanzar el objetivo de este TFM será preciso conocer los fundamentos de la simulación y adquirir destreza en el uso de Witness. En consecuencia, uno de los objetivos que se plantea este Trabajo Fin de Master es mostrar un primer acercamiento a este programa, conociendo qué se puede implementar en él, así como mostrar la programación, mediante ejemplos, que se puede usar para realizar la simulación de los procesos productivos.

De esta forma, se van a poder visualizar las técnicas al implementarse en distintos procesos productivos. Cabe destacar que los procesos no son reales; dado que la finalidad de este proyecto es educacional, no empresarial, no se quiere conocer cómo afectarían estas técnicas a un proceso real. Por esta razón, los datos que se han usado para la modelización de las técnicas Lean, así como de los procesos productivos, no son reales. De esta forma, se va a poder apreciar las mejoras que nos proporcionan estas técnicas, al hacer la comparación de un proceso con técnicas Lean con procesos en los que no se han aplicado.

Otra consideración importante para entender las técnicas Lean Manufacturing, es poder conocer cómo afectan estas técnicas al modelo. Por esta razón, se va a poder visualizar un modelo que no posee técnica Lean para posteriormente visualizarlo con la técnica Lean. De esta forma, se va a poder apreciar claramente cómo afecta la técnica Lean al proceso, debido a que se van a poder realizar comparaciones entre ambas visualizaciones.

Las conclusiones que se van a obtener no serán solo en forma de visualización, sino que se van a obtener resultados cuantitativos en forma de número de piezas obtenidas, tiempos que tardan las piezas en producirse, ...

En cuanto al planteamiento del trabajo, primero será necesario realizar un estudio sobre las técnicas Lean Manufacturing, para conocer todos los conceptos para realizar una simulación lo más exacta posible. A continuación, será necesario conocer el funcionamiento del programa de simulación, Witness, para saber que limitaciones se podrían tener al realizar la simulación en 3D.

Una vez que se conocen las técnicas de Lean Manufacturing y las posibilidades que nos aporta una simulación en 3D, será necesario discriminar aquellas técnicas Lean Manufacturing que no se puedan visualizar en un modelo en 3D como es el caso de la técnica de las 5S.

Finalmente, se construirán los modelos que permitan visualizar las herramientas seleccionadas, haciendo la comprobación de que en la visualización en 3D el modelo se aprecia cómo se desea, debido a que el modelo en 2D no posee las dificultades que se generan en el modelo en 3D.

### **3. DESARROLLO DEL TRABAJO**

El presente TFM se va a organizar en tres apartados fundamentalmente, en los que se tratarán todos los aspectos que se han creído necesarios para la comprensión del TFM, así como para poder entender los resultados obtenidos.

Para tener un primer contacto con la filosofía Lean y el entorno Witness, en el Capítulo 1, Estado del Arte, se hará un pequeño repaso a la filosofía Lean y a las técnicas que la componen,

prestando mayor atención a las técnicas que posteriormente se implementarán, así como una descripción del programa Witness, indicando sus distintos usos, así como una breve descripción de los elementos que se pueden simular y de su funcionamiento.

En los Capítulos 2 y 3, Descripción de los modelos I y II, se procederá a explicar las características de una serie de procesos productivos en los que no se han aplicado las técnicas Lean seleccionadas. Posteriormente, se aplicarán las técnicas Lean a dichos procesos productivos, explicándose las modificaciones que se han efectuado indicándose las nuevas configuraciones obtenidas al aplicar las técnicas Lean.

En concreto, en el Capítulo 2 se representarán los modelos asociados a las técnicas kanban y teoría de las limitaciones, herramientas que permiten reducir la cantidad de stock generado. En el Capítulo 3, por su parte, se mostrarán los modelos asociados las técnicas poka yoke, Célula en U y Heijunka, tratando en ellos mejoras asociadas a calidad y disminución de tiempos (lead time, tiempo de espera de los clientes, ...).

Una vez que se conoce cómo será el proceso productivo, sin implementar la técnica Lean e implementándola, en el Capítulo 4, se podrán visualizar en 3D los aspectos más característicos del funcionamiento de los modelos previamente explicados. Además, se podrán visualizar los modelos en unos instantes determinados, obteniendo resultados cuantitativos en ambos procesos en relación a números indicadores como, por ejemplo, el número de piezas defectuosas, nivel de stock, etc.

Para finalizar, en el apartado conclusiones y líneas futuras, se presentarán las principales conclusiones obtenidas durante la realización del presente trabajo, y se presentarán algunas de las posibles líneas de actuación futura.



## Capítulo 1.: ESTADO DEL ARTE



## 1. LEAN MANUFACTURING

### 1.1. INTRODUCCIÓN A LEAN MANUFACTURING

El pensamiento Lean se originó en Japón en la fábrica de coches Toyota, debido a las graves consecuencias que tuvo la segunda guerra mundial para Japón. La empresa Toyota se caracterizaba por producir pequeñas cantidades de gran variedad de productos a un precio bajo. Esto implicaba que no se podía beneficiar de economías de escala, como ocurría en el mercado automovilístico norteamericano, que producía grandes cantidades de productos de poca variedad. Por tanto, los japoneses tenían que mejorar su productividad sin poder aprovecharse de las economías de escala.

La filosofía seguida por Toyota consistía en aunar los principios de JIT (realizar el producto deseado, en el tiempo deseado y con la calidad deseada, usando la teoría de los 5 ceros: cero defectos, cero plazos, cero stocks, cero burocracia y cero averías) y los principios de Jidoka (realización correcta de las actividades, automatización con toque humano, sin realizar actividades innecesarias debidas, por ejemplo, a defectos de calidad), es decir, eliminar los despilfarros (Cuatrecasas, 2010).

Las medidas que se realizaron en el sistema productivo de Toyota para poder tener una posición competitiva en el mercado fueron las siguientes:

- Eliminar el despilfarro que se generaban en el proceso productivo y producir sólo cuando fuese necesario, es decir, Just in time.
- Mantener una relación de confianza y transparencia con los proveedores, para generar una colaboración a largo plazo.
- Una mayor participación de los empleados en la producción, ya sea teniendo en cuenta sus sugerencias, realizando mantenimiento preventivo, parar la producción cuando el producto no es el adecuado, ...
- Garantizar que la calidad es total, eliminando los productos del proceso productivo cuando no se cumpla esta característica e implementando técnicas para que no se generen desperfectos.

Para tener una mejor idea de qué se desea con la filosofía Lean, será necesario definir qué se denomina *despilfarro*: son las actividades de un proceso productivo en las cuales se consume un recurso, pero éste no genera ningún valor en el producto final. Estos despilfarros pueden ser (Wahab, 2013): por exceso de producción, por procesamiento inadecuado, por stocks, por transportes innecesarios, movimiento innecesario de personas, por tiempos de espera y por defectos de calidad.

La crisis de los años 70 generó, entre otros problemas, el aumento de los precios de las materias primas, haciendo que la firma japonesa se adaptase con rapidez a la situación económica, lo que la permitió salir de las crisis sin grandes dificultades. El éxito de la metodología Lean en la empresa Toyota durante la crisis, hizo que las empresas del ámbito automovilístico la implantasen, aunque no resultó tan fácil su implantación en otros sistemas productivos. Esta metodología se basa en satisfacer lo máximo posible las necesidades del consumidor con el menor despilfarro (muda) posible (Demeter, 2011). Una definición más concreta es: fabricar lo máximo posible con lo menor posible, ya sea esfuerzo humano, tiempo o espacio requerido para el almacenaje (Womack, 2003).

## 1.2. TÉCNICAS DE LEAN MANUFACTURING

### **SMED**

Como se ha comentado anteriormente, uno de los objetivos del Lean es evitar despilfarros. La técnica de SMED se centra en el despilfarro de tiempo por la preparación de las máquinas. La técnica distingue entre preparación interna (realizada con la máquina parada) y externa (estando la máquina en funcionamiento).

La implantación del SMED se lleva a cabo en una serie de etapas. Primero, se debe observar el proceso para conocer las distintas operaciones y poder diferenciar las preparaciones internas y externas. Una vez que se tienen identificadas qué tipo de preparación son, las preparaciones internas que se puedan realizar cuando la máquina se encuentra parada se convertirán en externas. A continuación, se procederá a mejorar los aspectos que sean posibles en las preparaciones, para disminuir tiempos. Por último, se deberá hacer una estandarización del proceso (Shingo, 2003).

Los beneficios que aporta la técnica del SMED al proceso productivo son: reducción del tamaño del lote, el tiempo de producción y el nivel de inventario, posibilidad de adaptarse a posibles fluctuaciones del mercado, y un mejor control de la calidad al trabajar con lotes pequeños.

### **5S**

Se trata de una técnica de fácil implantación, que no necesita de grandes desembolsos de dinero y que permite obtener mejoras en los puestos de trabajo. La técnica de las 5S es un sistema visual basado en la limpieza y la organización de la zona de trabajo cuya finalidad es mejorar la calidad, la seguridad y la productividad. Para conseguir que un entorno de trabajo cumpla las 5S se deben realizar las siguientes tareas: eliminar los elementos innecesarios (clasificación, Seiri), ordenar los elementos que son necesarios para llevar a cabo la producción (orden, Seiton), limpiar la zona de trabajo (Limpieza, Seisō) y conseguir que las etapas anteriores permanezcan en el tiempo, es decir, mantener el orden y limpieza (Mantener la disciplina, Shitsuke).

### **Mantenimiento Productivo Total (TPM)**

La técnica de mantenimiento productivo total tiene como objetivo eliminar las averías a través de los propios trabajadores, estando éstos concienciados de que tienen que realizar las operaciones de mantenimiento, para conseguir que se eliminen las posibles anomalías de las máquinas antes de que éstas lleguen a ser averías. Para ello, debe formarse a los operarios para que puedan reconocer las anomalías y poder subsanarlas. En esta técnica es de vital importancia el uso de las 5S para poder ver las anomalías.

### **Kanban**

Se trata de un sistema de programación en el que se genera la cantidad justa en el momento adecuado. Cada operación del proceso productivo retira los elementos de la operación anterior (sistema pull), comenzándose a producir en éstos la cantidad que se ha retirado. Este sistema está basado en tarjetas, en las que se especifica la cantidad de elementos que debería tener el contenedor al que va acompañando. Además, en las tarjetas se indica qué tipo de elemento hay que fabricar, centro de trabajo donde se utilizan, dónde se tienen que fabricar, o cuántas fabricar, etc. Los kanban pueden ser de producción, que indican cuánto hay que

fabricar para el proceso posterior y sólo se mueven dentro del puesto de trabajo, o de transporte, que se mueven entre distintos puestos de trabajo e indican cuantos elementos hay que realizar en el puesto anterior.(Prieto, 2016)

### **Flexibilidad**

La demanda de un determinado producto puede variar, por lo que será necesario que los trabajadores se adapten a la nueva situación. Se pueden adaptar de dos formas, aumentando o disminuyendo las operaciones que tiene que realizar cada operario (por lo que será deseado que éstos estén en células en forma de U, para que puedan realizar todas las actividades en el menor tiempo posible o que los trabajadores sean polivalentes y puedan trabajar en otro centro de trabajo) o aumentando o disminuyendo el número de trabajadores para realizar una tarea (por lo que habrá que reconfigurar las operaciones a realizar por un operario).

### **Heijunka**

Uno de los objetivos de la metodología Lean es reducir inventarios, pero la demanda de los clientes es muy variable, por lo que la producción hay que adaptarla a esta condición, ya sea por flexibilidad o compensando la demanda, es decir, realizar distintas secuencias de fabricación de productos. Con la técnica Heijunka no se consigue ajustar la producción a la demanda del cliente, sino que se basa en ella para realizar la secuencia de productos a realizar; de esta forma, no se producen variaciones de producción. Así se producirá en pequeños lotes varios productos, para no estar condicionados a una única demanda de un cliente.

### **Estandarización de las operaciones**

Esta técnica Lean consiste en determinar las operaciones que debe realizar un operario polivalente para obtener una alta productividad (mínimo número de trabajadores ya que realizan las operaciones sin movimientos inútiles) con la menor cantidad de trabajo en curso. Para ello, será necesario definir el Takt time (ritmo de trabajo para cumplir con las necesidades del cliente), inventario estándar (el requerido para mantener el proceso productivo sin problemas) y secuencia de trabajo (el operario deberá hacer las operaciones dentro del tiempo que tiene establecido).

Para obtener una estandarización de operaciones será necesario una hoja estándar de operaciones que deberá poseer: ficha de capacidad (indica la capacidad de cada máquina, así como el tiempo que tarda en hacer las operaciones), hoja de ruta estándar (operaciones que debe realizar cada trabajador en un tiempo determinado) y hoja de trabajo estándar (donde se indica el movimiento que lleva cada operario y la ubicación del material que debe de usar y de las máquinas).

### **Jidoka**

Esta tarea busca comprobar que la calidad de los productos sea la adecuada. Se define como automatización con toque humano, ya que los propios trabajadores se encargan de que las piezas no posean defectos; es decir, son operarios de línea además de inspectores de calidad. Los empleados se encargan de que los defectos no se propaguen a lo largo del proceso productivo; para ello, tendrán que detectar el problema cuando ocurra, detener la producción para que no se produzcan más errores en otros elementos (andón), posteriormente corregir el problema y determinar la causa que ha generado el problema y eliminarla (Poka Yoke).

### 1.3. LEAN MANUFACTURING EN EL ENTORNO EDUCACIONAL

Como se ha comentado tanto en el apartado de Antecedentes como en el de Objetivos, conocer las técnicas de Lean Manufacturing es necesario en el ámbito empresarial, no solo desde el punto de vista de conceptual sino también desde el punto de vista práctico. Por esta razón, la forma de enseñar las técnicas Lean debe adaptarse al nivel de exigencia de las empresas.

En la mayoría de los casos, los estudiantes no pueden conocer *in situ* cómo se aplican técnicas en los casos reales, es decir, en los procesos productivos, dado que es muy complicado para éstos acceder a las industrias. Por este motivo, se han creado escuelas Lean, para que los alumnos puedan conocer de forma más práctica las técnicas Lean, sin necesidad de acudir a procesos productivos reales.

La Universidad de Valladolid, con el apoyo de Renault Consulting, inauguró en 2014 la primera Escuela Lean de España, donde los alumnos pueden enfrentarse a problemas que se pueden encontrar en los procesos productivos, aplicando técnicas Lean para mejorarlos.

En las Universidades que no poseen métodos de aprendizaje como los que hay disponibles en la Escuela Lean, la forma de enseñar estas técnicas es generalmente muy teórica, faltando la experiencia práctica que desean las empresas. Una forma suplir la carencia de estos métodos de aprendizaje, sería mediante simulaciones, como las que se van a presentar en los próximos capítulos. En los distintos modelos se van a poder identificar el problema que poseen, ya sea por despilfarro de piezas, tiempos, ..., para poder aplicar la técnica Lean asociada, pudiéndose apreciar las mejoras que se generan al aplicar las técnicas Lean Manufacturing.

## 2. WITNESS

### 2.1. INTRODUCCIÓN A WITNESS

Witness es una potente herramienta de simulación, usada en una gran variedad de ámbitos, desde métodos educacionales (como se trata de este caso), para poder entender el funcionamiento de algunas técnicas, hasta el ámbito empresarial, para poder modelar distintos entornos de trabajo y poder realizar una toma de decisiones antes de llevar a cabo el proyecto.

Actualmente, el uso empresarial está muy extendido, debido a que se pueden diseñar modelos de producción antes de tomar una decisión, como se ha comentado anteriormente, pero también es útil para conocer cómo se va a comportar el modelo en distintos instantes de tiempo, conocer cómo va a ser el funcionamiento del proceso, poder realizar distintas alternativas para conocer con cuál el funcionamiento se aproxima mejor al deseado, ... (Ortega 2010).

Otra gran ventaja que posee Witness es la capacidad visual, permitiendo observar, tanto en 2D como en 3D, cómo es el funcionamiento del proceso productivo en distintos instantes de tiempos, conociendo en todo momento lo que está ocurriendo en el modelo.

### 2.2. DESCRIPCIÓN DEL PROGRAMA

Como se ha comentado anteriormente, Witness es un programa de simulación y como tal, debe poseer todos los elementos que se puedan necesitar para implementar modelo que se

desea simular. Los elementos con los que se puede trabajar en Witness son los que aparecen en la Figura 1.1. Estos elementos pueden dividirse en cuatro grupos principalmente, indicándose a continuación los elementos que forman cada uno de los grupos (De Benito, 2014):

- Elementos procesos discretos: Part, Machine, Conveyor, Buffer, Labor, Track, Vehicle, Module, PF Network, PF Section, PF Carrier y Path.
- Elementos procesos continuos: Fluids, Tanks, Processors y Pipes.
- Elementos lógicos: Attributes, Variables, Distributios, Files, Function, Shift y Part file.
- Elementos gráficos: Timeseries, Histograms y Piechart.

Los elementos que se van a usar a lo largo de este Trabajo Fin de Máster serán elementos de procesamiento discreto, aunque también se necesitarán elementos lógicos para definir las características necesarias en el proceso. A continuación, se van a describir los elementos que se han usado en la construcción de los modelos:

- Part: elementos principales del modelo, que se irán desplazando de una máquina a otra. Estos elementos procederán del **world**, pudiéndose describir tanto el momento como la cantidad en la que entran en el modelo. Cuando finalizan las operaciones que tenían previstas en el modelo se podrán disponer en **ship** (salida del modelo) o hacia **scrap** (desecho).
- Machine: son los elementos que realizan las operaciones a los parts. Éstas pueden ser: single, si sólo trabajan con un part cada vez; batch, si trabajan con varias piezas; assembly, ensamblando varias piezas a la vez; production, entrando una pieza y generándose una cantidad fija; general, cuando entra una cantidad fija de piezas a la máquina y se produce otra cantidad fija; multi-station, máquinas en las que a las piezas se la realizan varias operaciones. A las máquinas se las puede asociar acciones previas a realizar las operaciones, acciones a la entrada de las piezas, a la salida y al finalizar la operación.
- Buffers: elementos en los que se almacenan piezas. Generalmente, son elementos pasivos, dado que no pueden disponer piezas a otros elementos.
- Conveyor: son elementos de transporte que, a diferencia de los buffers, son elementos activos.
- Labour: son los recursos para realizar una determinada actividad. En algunos casos se puede asociar un labour a una máquina o a las tareas de mantenimiento de éstas.
- Vehicle: son los elementos que transportan piezas de un punto de carga a uno de descarga.
- Track: son los elementos por los que circulan los vehicles. Se deben definir los puntos de carga y descarga.
- Attributes: valores asociados a los parts, para diferenciar unos parts de otros.
- Variables: son valores, pero asociados a un determinado instante de la simulación, no asociados a parts.

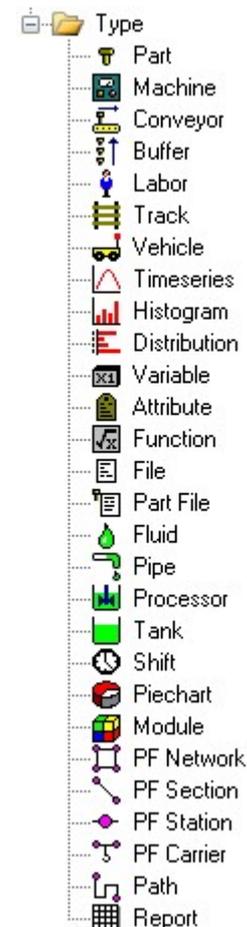


Figura 1.1: elementos que se pueden simular en Witness

- Path: son los trayectos que deben realizar los labour entre dos elementos.

Una vez que se conocen los elementos que se han usado en la simulación, en la Figura 1.2 se puede apreciar la interfaz de Witness.

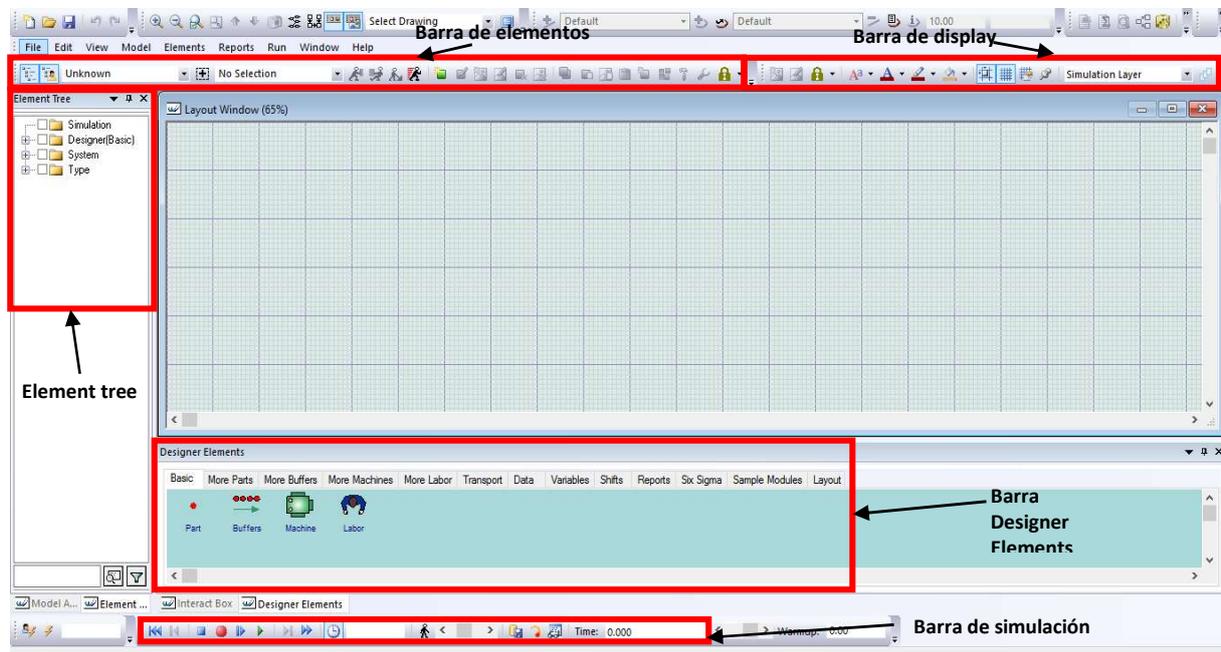


Figura 1.2: ventana de simulación de Witness

- Layout: donde se crearán los diferentes elementos, conectándoles mediante reglas de entrada y salida.
- Barra de simulación: se encuentran los botones que nos van a permitir llevar a cabo la simulación, dados por run, stop, batch (simular hasta un determinado instante), retroceder, ... Además, se puede apreciar el instante en el que se encuentra la simulación, dado por time.
- Element tree. En ellas se pueden observar varias pestañas: pestaña de Simulación, donde aparecerán todos los elementos que se han definido en el modelo; en la asociada a Designer (Basic) aparecerán los elementos básicos de la simulación, part, machine, buffer y labour; en System aparecerán los elementos de dónde pueden proceder los parts o a dónde se dirigen después de finalizar el proceso; Type son todos los elementos que se pueden generar en los modelos, Figura 1.1.
- Designer elements: donde se pueden seleccionar los distintos elementos creándose con todas las representaciones posibles. Por ejemplo, una máquina poseerá representación de la pieza que está procesando, el estado en el que se encuentra, el icono del operario, ...
- Barra de display: usada para cambiar la apariencia de los elementos, desde los iconos que se les ha asociado hasta el formato de su letra.
- Barra de elementos: permite modificar los elementos que se han generado, desde su representación gráfica hasta las reglas de entrada y salida de ese elemento.

### 2.3. CONSIDERACIONES AL REALIZAR EL MODELO EN 3D

Una vez que se conocen los elementos que van a formar parte de los modelos, se procederá a realizar una serie de consideración para entender mejor los modelos.

A lo largo del TFM se hará referencia a máquinas ficticias; éstas son máquinas que se han creado para poder simular una actividad sin ser realizada por una máquina física, es decir, cuando lo que se quiere representar es una actividad sin dar importancia a una máquina. En estos casos, sólo se ha representado el nombre de la máquina y en algunas ocasiones el operario y el part que lleva asociado. Estas máquinas se han usado principalmente en las ocasiones donde se quiere simular una operación realizada por un operario, el ensamblaje de piezas en el final de un proceso, para transporte de éstas por un vehículo o la división de éstas después de haber realizado el transporte. En algunos casos, se han usado estas máquinas en la salida de piezas del proceso para mejorar la visualización del modelo en 3D: cuando se almacenan piezas a la salida del proceso en contenedores, mediante estas máquinas ficticias del tipo ensamblaje se evita la formación grandes acumulaciones.

Para poder distinguir ciertos elementos (path, track, conveyor, ...), tanto en el 3D como en el modelo 2D, se ha decidido que posean una misma representación en los distintos modelos. De esta forma, se poseen las siguientes representaciones:

- Conveyor: el método de llevar piezas de una máquina a otra en la mayoría de los casos se realizará mediante conveyor, debido a que se tratan de elementos activos, pudiéndose definir en ellos reglas de entrada y salida. Su representación en 2D es la mostrada en Figura 1.3, y la representación en 3D en la Figura 1.6.
- Path: usado para definir los trayectos a realizar por los operarios entre las máquinas. Su representación en 2D se puede apreciar en la Figura 1.4, y en 3D viene dada por la Figura 1.7.
- Track: usados para el transporte de piezas desde una zona del proceso a otra mediante un vehículo. Su representación en 2D se puede apreciar en la Figura 1.5, siendo la representación en 3D la mostrada en la Figura 1.8.



Figura 1.3: representación en 2D de conveyor



Figura 1.4: representación en 2D de path



Figura 1.5: representación en 2D de track

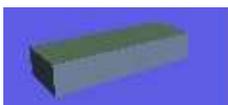


Figura 1.6: representación en 3D de conveyor



Figura 1.7: representación en 3D de path

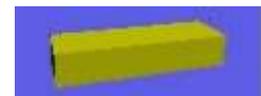


Figura 1.8: representación en 3D de track



## Capítulo 2.: DESCRIPCIÓN DE LOS MODELOS I



En el presente capítulo, se va a proceder a explicar los modelos en los que se pueden implementar las técnicas kanban y elemento capacitivo, así como los modelos en los que no se han implementado. De esta forma, se disminuirán los niveles de stock intermedio en ambos modelos y, con ello, el capital inmovilizado que se genera.

En primer lugar, se van a explicar los modelos en los que se va a aplicar la técnica kanban. Para ello, se explicará un modelo que no posee kanban, generándose por tanto una gran cantidad de stock, para posteriormente aplicar técnica kanban, produciendo sólo cuando es necesario.

Posteriormente, se explicará un modelo que posee una máquina que es cuello de botella, no pudiendo procesar todas las piezas que se generan. Para solucionar este problema, se implanta un elemento capacitivo, para evitar el stock intermedio que se crea.

## 1. TÉCNICA KANBAN

### 1.1. DESCRIPCIÓN DEL MODELO SIN TÉCNICA KANBAN

Se posee un proceso productivo dividido en dos zonas: zona de fabricación y zona de acabado. Este proceso se puede visualizar en la Figura 2.1, donde se puede apreciar, en la zona superior, la zona de fabricación y, en la zona inferior, la de acabado. El transporte de las piezas de una zona a otra la realiza un **vehículo** que circula a través de unas vías denominadas **hacia\_almacen** y **hacia\_final**. La entrada en la zona de acabado (**entrada\_final**) se realizará en lotes de 5 piezas. En la zona de fabricación hay tres contenedores de almacenaje, igual que en la zona de acabado. Cabe destacar que, aunque en la zona final se almacenen en cada uno de los contenedores una cantidad superior a la necesaria en esta zona (5 piezas), la cantidad en éstos no será superior a 10 piezas. Además, se disponen de dos operarios, **operario1** y **operario2** que se ocupan de las tareas de mantenimiento de las máquinas (**operario1** para la zona de acabado y **operario2** en la zona de fabricación), éstos deberán trasladarse por los paths que se han implementado (**fab2\_3**, **fab2\_1**, **maquina3\_2** y **maquina2\_1**).

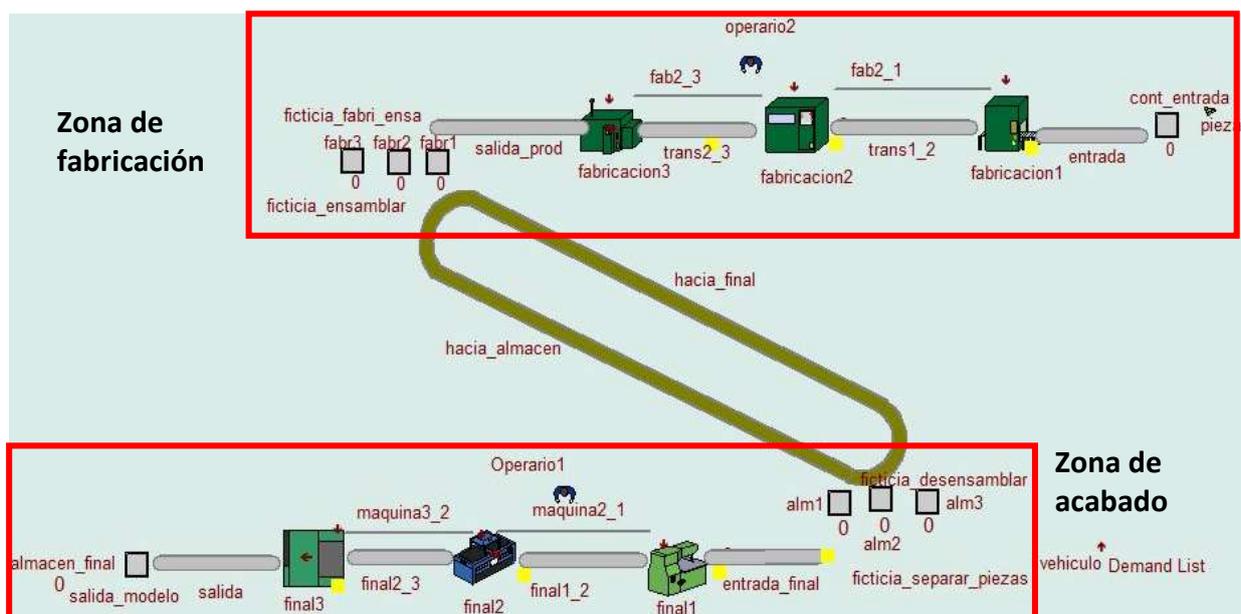


Figura 2.1: representación del modelo sin kanban

La entrada de piezas al proceso se realizará a través del contenedor **cont\_entrada**, donde se depositarán las piezas hasta que sean necesarias en el proceso productivo. En este contenedor se ha implementado una variable denominada **piezas\_entrada**, para conocer el número de

piezas que entran en el modelo. La representación las piezas tanto en 2D como en 3D se pueden apreciar en la Figura 2.2



Figura 2.2: representación de la pieza 2D y 3D modelos técnica kanban

A las piezas se las realizan una serie de operaciones, tanto en la zona de fabricación como en la de acabado, siendo el tiempo de cada una de las operaciones diferente. El tiempo de ciclo de las distintas máquinas, así como de los posibles fallos, se muestran en la Tabla 2.1. Por tanto, se puede apreciar que el tiempo de ciclo de la zona de fabricación, sin tener en cuenta las operaciones de mantenimiento, será de 21 s. En la zona de acabado, formado por tres operaciones, el takt time es de 170 s., como en el caso anterior, sin tener en cuenta las operaciones de mantenimiento.

Máquina	Tiempo de ciclo (s)	Mantenimiento (time between failures)
Fabricación1	8	650 s, time repair=45
Fabricación2	7	400 s, time repair=45
Fabricación3	6	600 s, time repair=30
Final1	60	400 s, time repair=30
Final2	50	800 s, time repair=20
Final3	60	500 s, time repair=60

Tabla 2.1: tiempo de ciclo las máquinas modelos técnica kanban

Comparando los tiempos de ciclo de las dos zonas, se puede comprobar que el takt time de la zona de acabado es muy superior a la zona de fabricación, haciendo que no se pueda procesar todas las piezas que se generan en la zona de fabricación.

Cabe destacar que, en el momento inicial, para simular que ya se encuentra en un estado estacionario, a la entrada de cada una de las máquinas de la zona de fabricación ya se disponen 5 piezas. Como se puede apreciar en la Figura 2.3, se cogerán piezas de *world* cuando las máquinas no hayan realizado ninguna operación, éstas no hayan cogido ninguna pieza y haya menos de 5 piezas en las cintas correspondientes.

```

Edit INPUT RULE FOR CONVEYOR trans1_2
Select Search Editor Print
IF NParts (trans1_2) <= 4 AND NQps (fabricacion2) = 0 AND NParts (fabricacion2) = 0
PULL from pieza out of 'WORLD'
ELSE
Wait
FNDIF
    
```

Figura 2.3: entrada de piezas en el instante inicial en el modelo sin kanban

Para poder almacenar el stock generado por la diferencia de takt time, se dispone de tres contenedores en la zona de fabricación, denominados *fabr1*, *fabr2*, *fabr3* y otros tres en la zona de acabado, denominados *alm1*, *alm2* y *alm3*. Los contenedores de la zona de fabricación poseen una cantidad máxima de almacenaje de 100 piezas. Se ha implementado una variable de tipo vector denominada *cant\_contenedor\_fabr()* de tamaño 3, para

```

Edit Actions On Input For Buffer fabr1
Select Search Editor Print
cant_contenedor_fabr(1) = cant_contenedor_fabr(1) + 1
    
```

Figura 2.4: regla entrada contenedores fabricación en el modelo sin kanban

poder conocer la cantidad que se ha depositado en cada contenedor de la zona de fabricación. De esta forma, `cant_contenedor_fabr(1)` corresponderá al contenedor `fabr1` y se procederá de la misma forma para `fabr2` y `fabr3`. En la Figura 2.4, se puede apreciar la regla de entrada de estos contenedores, viéndose cómo se incrementa esta variable cada vez que entra una pieza en el contenedor.

La acción de llenar los contenedores a la salida de fabricación la lleva a cabo una máquina ficticia denominada `ficticia_fabri_ensa`, que saca las piezas de `salida_prod` y las dispone en los contenedores `fabr1`, `fabr2` y `fabr3`. Las piezas se dispondrán en un contenedor u otro dependiendo de dónde se haya colocado la anterior (es decir, dependiendo del valor de la variable `cant_contenedor_fabr()`), teniendo en cuenta que el número de piezas consecutivas depositadas en un contenedor no puede ser superior a 10, cuando se llega a este límite, se comienza a llenar otro, como se puede apreciar en las reglas de salida de la máquina, Figura 2.5. De esta manera, cuando en `fabr3` (que se trata del contenedor que tiene la menor prioridad de recibir piezas) la variable `cant_contenedor_fabr(3)` tome valor 10, se inicializarán a cero los valores del vector `cant_contenedor_fabr()`, como se puede apreciar en la Figura 2.6. Así, cuando llegue la próxima pieza desde `salida_prod`, ésta se depositará en el contenedor `fabr1`, dado que el valor de `cant_contenedor_fabr(1)` tiene valor 0.

Select	Search	Editor	Print
<pre> IF cant_contenedor_fabr(1) &lt;= 9   PUSH to fabr1 ELSEIF cant_contenedor_fabr(2) &lt;= 9   PUSH to fabr2 ELSEIF cant_contenedor_fabr(3) &lt;= 9   PUSH to fabr3 ELSE   Wait ENDIF </pre>			

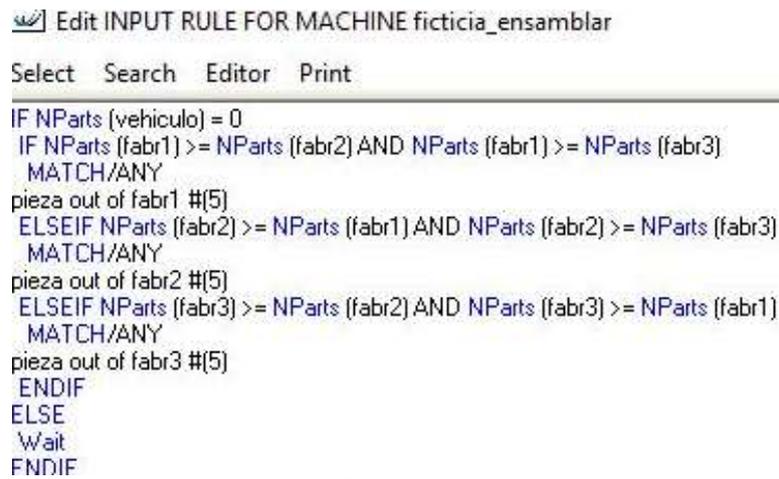
Select	Search	Editor	Print
<pre> cant_contenedor_fabr(3) = cant_contenedor_fabr(3) + 1 IF cant_contenedor_fabr(3) = 10   cant_contenedor_fabr(1) = 0   cant_contenedor_fabr(2) = 0   cant_contenedor_fabr(3) = 0 ENDIF </pre>			

Figura 2.5: regla de distribución de piezas a contenedores fabricación en el modelo sin kanban

Figura 2.6: inicialización contador contenedores fabricación en el modelo sin kanban

La llegada de piezas a la zona de almacenaje de acabado se realiza en lotes de 5 piezas, situándolas en los contenedores de entrada a la zona de acabado (`alm1`, `alm2` y `alm3`). Antes de realizar el transporte de las piezas, se ha implementado una máquina ficticia denominada `ficticia_ensamblar`, que juntará las cinco piezas que se deben transportar desde la zona de fabricación a la zona de acabado, en una única pieza. Cabe destacar que, a la hora de definir esta máquina, se ha decidido que se mantengan los parts iniciales. La necesidad de ensamblar las piezas se debe a que el `vehículo` sólo es capaz de transportar una pieza cada vez, y el modelo exige el transporte de cinco piezas.

Como se ha comentado anteriormente, la capacidad máxima de cada uno de los contenedores de la zona final es de 10 piezas. Cabe señalar que, aunque éstos se encuentren con su número máximo de piezas, se realizará el transporte de un lote más de piezas, esperando éste el momento en el que sea necesario rellenar uno de los contenedores. Por tanto, se dispone de un stock de 10 piezas en cada uno de los contenedores y 5 piezas que se encuentran transportadas por el `vehículo` al final de la vía `hacia_final`. Por esta razón, `ficticia_ensamblar` cogerá piezas de los contenedores de fabricación para llevarlos a la zona de acabado, mediante el `vehículo`, cuando haya en éste no haya ninguna pieza (Figura 2.7), es decir, no se posea el stock de 5 piezas.



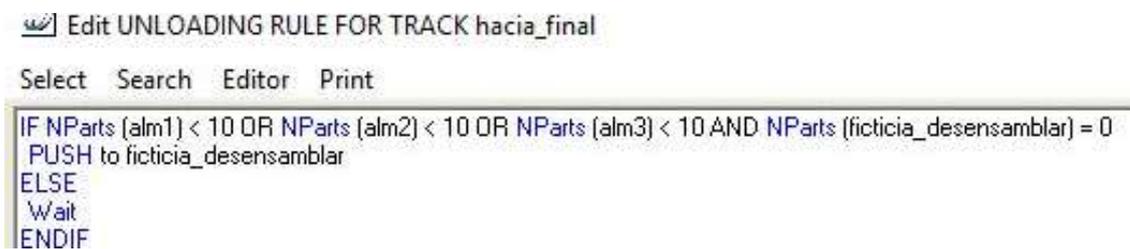
```

Edit INPUT RULE FOR MACHINE ficticia_ensamblar
Select Search Editor Print
IF NParts (vehiculo) = 0
IF NParts (fabr1) >= NParts (fabr2) AND NParts (fabr1) >= NParts (fabr3)
MATCH/ANY
pieza out of fabr1 #(5)
ELSEIF NParts (fabr2) >= NParts (fabr1) AND NParts (fabr2) >= NParts (fabr3)
MATCH/ANY
pieza out of fabr2 #(5)
ELSEIF NParts (fabr3) >= NParts (fabr2) AND NParts (fabr3) >= NParts (fabr1)
MATCH/ANY
pieza out of fabr3 #(5)
ENDIF
ELSE
Wait
ENDIF

```

Figura 2.7: regla de entrada para ficticia\_ensamblar en el modelo sin kanban

Al llegar al final del trayecto, el **vehículo** no realizará automáticamente la descarga de la pieza, dado que puede que los contenedores pueden que se encuentren llenos y no sea necesaria esta acción, haciendo que la pieza tenga que esperar a ser descargada cuando en alguno de los contenedores se saquen 5 piezas. Además, cabe destacar que puede que la máquina **ficticia\_desensamblar** (que se explicará a continuación) esté procesando alguna pieza, haciendo que los contenedores reciban próximamente piezas. Por tanto, la regla de salida de piezas del **vehículo** deberá tener la condición de que alguno de los contenedores tenga menos de 10 piezas y que la máquina desensamblar no esté procesando ninguna pieza, como se muestra en la Figura 2.8.



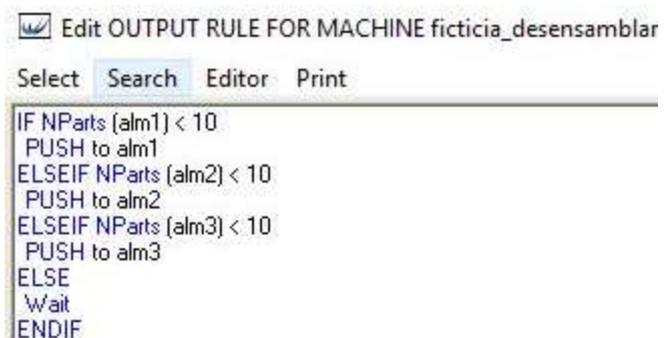
```

Edit UNLOADING RULE FOR TRACK hacia_final
Select Search Editor Print
IF NParts (alm1) < 10 OR NParts (alm2) < 10 OR NParts (alm3) < 10 AND NParts (ficticia_desensamblar) = 0
PUSH to ficticia_desensamblar
ELSE
Wait
ENDIF

```

Figura 2.8: regla de salida vehiculo modelo sin kanban

Después de realizarse el transporte de la pieza (conjunto de las cinco piezas), con un tiempo de carga y descarga de la pieza de 5s., será necesario volver a tener esas cinco piezas. Para ello, se dispone de una máquina ficticia de producción, **ficticia\_desensamblar**. Con esta máquina se generan las cinco piezas que había antes de realizar el transporte. Esta máquina ficticia colocará las piezas en los contenedores de acabado, siendo prioritario el contenedor **alm1**, como se puede apreciar en la regla de salida de la máquina, en la Figura 2.9.



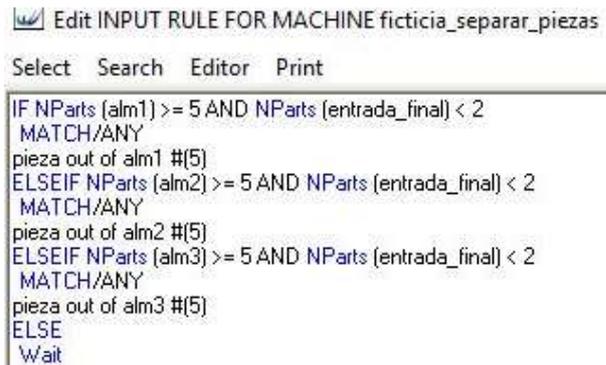
```

Edit OUTPUT RULE FOR MACHINE ficticia_desensamblar
Select Search Editor Print
IF NParts (alm1) < 10
PUSH to alm1
ELSEIF NParts (alm2) < 10
PUSH to alm2
ELSEIF NParts (alm3) < 10
PUSH to alm3
ELSE
Wait
ENDIF

```

Figura 2.9: regla de colocación las piezas en los contenedores de acabado en el modelo sin kanban

La entrada de piezas a la zona de acabado se realizará en lotes de 5 piezas, pero no se situarán en la cinta `entrada_final` hasta que no haya en ésta un mínimo de dos piezas. Para sacar piezas de los contenedores de acabado y situarlas en la cinta `entrada_final`, se dispone de una máquina ficticia denominada `ficticia_separar_piezas`, que solo actuará cuando haya menos de dos piezas en `entrada_final`, como se aprecia en la Figura 2.10.



```

Edit INPUT RULE FOR MACHINE ficticia_separar_piezas
Select Search Editor Print
IF NParts (alm1) >= 5 AND NParts (entrada_final) < 2
MATCH/ANY
pieza out of alm1 #(5)
ELSEIF NParts (alm2) >= 5 AND NParts (entrada_final) < 2
MATCH/ANY
pieza out of alm2 #(5)
ELSEIF NParts (alm3) >= 5 AND NParts (entrada_final) < 2
MATCH/ANY
pieza out of alm3 #(5)
ELSE
Wait

```

Figura 2.10: regla de salida de piezas de los contenedores de almacenaje zona acabado modelo sin kanban

Cuando se realicen todas las operaciones de la zona de acabado, se disponen las piezas en una cinta denominada `salida`, para colocarlas en un contenedor denominado `almacen_final`. A la entrada de este contenedor se ha implementado una variable denominada `piezas_salida`, para conocer las piezas que salen del proceso.

Para evitar la acumulación de piezas en este contenedor, y mejorar la apariencia visual del modelo 3D, se dispone de una máquina ficticia, denominada `salida_modelo`, que coge piezas de este contenedor cuando en él haya como máximo 10 piezas, depositándolas en `ship`.

## 1.2. IMPLEMENTACIÓN DEL KANBAN

En el modelo descrito anteriormente se produce un elevado stock, la mayor parte almacenado en la zona de salida de fabricación (`fabr1`, `fabr2` y `fabr3`), dado que en la zona de almacenaje de acabado (`alm1`, `alm2` y `alm3`) está limitada la cantidad a 10 piezas en cada contenedor (aunque se está almacenando más cantidad que la necesaria). Para eliminar la cantidad de stock que se genera, tanto en la zona de acabado como en la zona de fabricación, se ha generado un modelo gobernado por la técnica kanban. Es decir, se ha determinado un stock mínimo en cada una de las zonas de almacenaje, de 10 unidades en cada una, garantizando que el proceso no se paralice por falta de piezas. Los dos tipos de kanban que se han implementado son: de transporte y de fabricación. El kanban de transporte será el encargado de dar la orden de transporte de piezas desde los contenedores de fabricación hasta los de acabado y el kanban de fabricación el de dar la orden de fabricación cuando en los contenedores de almacenaje de la zona de fabricación no se posea el stock mínimo.

Cabe destacar que, en este modelo, en comparación con el modelo descrito anteriormente, sólo se han implementado dos contenedores de almacenaje en cada una de las zonas, acabado (`alm1` y `alm2`) y fabricación (`fabr1` y `fabr2`). Esta disminución del número de contenedores es debido a que, como se ha comentado anteriormente, con un sistema kanban no se necesita gran nivel de almacenaje, solo el mínimo, cinco piezas en cada uno de los contenedores; es decir, un stock de 10 piezas en cada una de las zonas. El modelo con la implementación del sistema kanban se muestra en la Figura 2.11, pudiéndose apreciar las diferencias con el modelo comentado anteriormente.

Una vez que se conoce cómo es el modelo, y sabiendo que el número de máquinas en la zona de fabricación y acabado, así como el tiempo de ciclo de éstas es igual que en el modelo sin kanban, se va a proceder a explicar el funcionamiento del modelo con kanban.

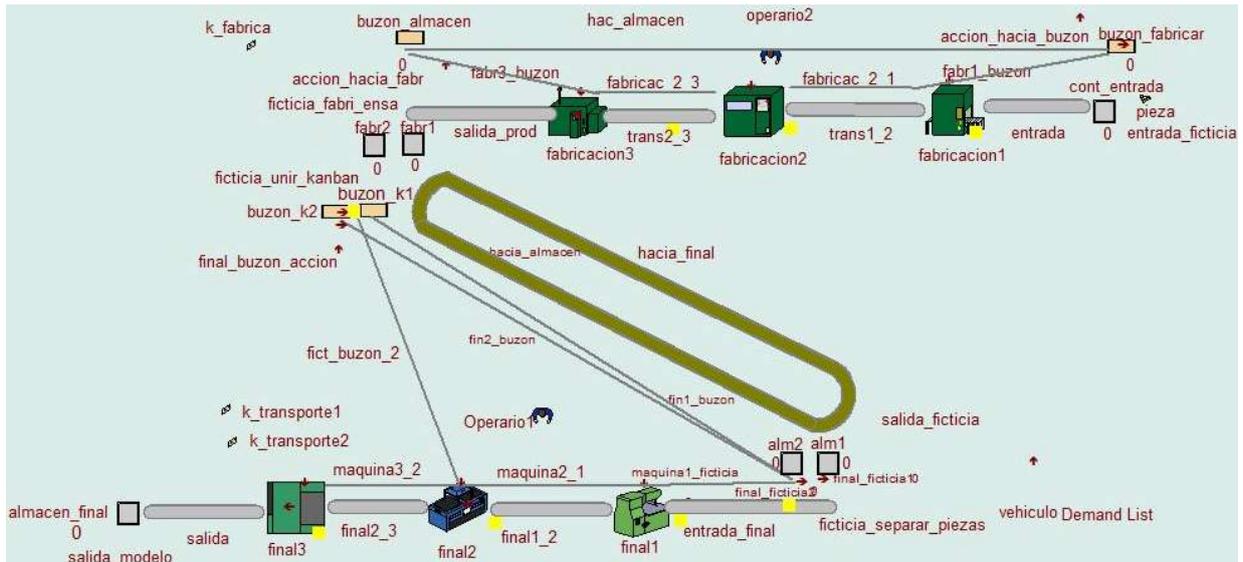


Figura 2.11: representación del modelo con kanban

Para poder visualizar mejor cómo es la zona de acabado, se ha dispuesto la Figura 2.12, donde se pueden apreciar mejor los elementos que forman parte de ésta. En la zona de acabado, el modelo sigue trabajando de la misma forma que sin kanban, con lotes de 5 piezas. Por tanto, para evitar stock innecesario, estos contenedores almacenarán como máximo 5 unidades cada uno, para asegurar que la entrada de la zona de acabado siempre tenga stock. Estos contenedores poseen, cada uno de ellos, un kanban de transporte, denominados **k\_transporte1** y **k\_transporte2** (que indicarán la necesidad de transporte entre la zona de fabricación y acabado), siendo el correspondiente al contenedor **alm1** el **k\_transporte1** y el **k\_transporte2** el que se asigne al contenedor **alm2**. Cuando alguno de los dos contenedores se quede vacío, se transportarán piezas desde la zona de fabricación a la de acabado, hasta que los dos contenedores estén llenos, es decir, el stock en la zona de acabado sea de 10 piezas.

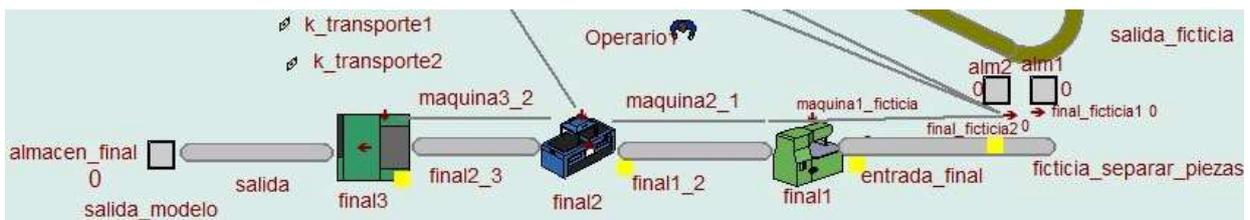


Figura 2.12: descripción de la zona de acabado modelo con kanban

Tanto el **k\_transporte1** como el **k\_transporte2** llegarán a sendos buffers ficticios denominados **final\_ficticia1** y **final\_ficticia2**. Ambos buffers son necesarios para poder simular que los kanban de transporte se encuentran en los contenedores **alm1** y **alm2**. En cuanto al uso del kanban de transporte, se realizará cuando la cantidad en uno de los contenedores sea cero (dado que la cantidad máxima en éstos es de 5 piezas y los lotes son de 5 unidades, es decir, cuando se tiene necesidad de sacar de un contenedor, éste queda vacío). Por tanto, el kanban de transporte saldrá del buffer correspondiente y se depositará en su buzón correspondiente, es decir, el **k\_transporte1** en **buzon\_k1** y el **k\_transporte2** en **buzon\_k2**.

El método visual de llevar los kanban de transporte desde la zona de almacenaje de acabado hacia su buzón correspondiente, se realizará mediante una máquina ficticia denominada **final\_buzon\_accion**. Esta acción necesitará de un operario que la realice; por tanto, el **operario1** que se encarga de las tareas de mantenimiento de la zona de acabado será el que

lleve el kanban de una zona a otra. La máquina ficticia `final_buzon_accion` llevará el kanban de transporte desde `final_ficticia` hasta `buzon_k` cuando alguno de los dos contenedores no posea piezas, como se puede apreciar en la Figura 2.13. Además, el `operario1` tendrá que llevar el kanban por un camino estipulado, pudiendo ser `final1_buzon` o `final2_buzon`, dependiendo de si el contenedor que se ha quedado vacío sea el contenedor `alm1` o `alm2`. También se ha definido un camino para que el operario se mueva entre los buzones y la `maquina2`, denominado `fict_buzon_2`, dado que conectando esta máquina con los buzones ya puede hacer el recorrido a las otras máquinas. De la misma forma, se ha creado un camino entre los contenedores y la `maquina1`, denominada `maquina1_ficticia`.

```

Edit INPUT RULE FOR MACHINE final_buzon_accion
Select Search Editor Print
IF NParts (alm1) = 0 AND NParts (final_ficticia1) = 1
MATCH/ANY
k_transporte1 out of final_ficticia1 With Operario1 Using Path #(1)
ELSEIF NParts (alm2) = 0 AND NParts (final_ficticia2) = 1
MATCH/ANY
k_transporte2 out of final_ficticia2 With Operario1 Using Path #(1)
ELSE
Wait
ENDIF

```

Figura 2.13: regla de entrada `final_buzón_acción` modelo con kanban

Como se ha comentado anteriormente, que llegue un kanban de transporte a uno de los buzones implica que se debe de retirar un lote de 5 piezas de uno de los contenedores de fabricación (`fabr1` o `fabr2`). Las 5 piezas del contenedor se retirarán mediante una máquina ficticia denominada `ficticia_unir_kanban`, en esta operación se retirarán las piezas de uno de los contenedores y el kanban de transporte correspondiente. `Ficticia_unir_kanban` es una máquina de ensamblaje que, a la hora de su definición, se ha determinado que mantengan los parts de inicio (el kanban de transporte que se encontraba en el buzón y las cinco piezas que se encontraban en el contenedor de fabricación), haciendo que se unan en una única pieza, como se aprecia en la Figura 2.14. Una vez que se conoce cómo es el funcionamiento de `ficticia_unir_kanban`, se puede comprender la necesidad de haber creado dos `buzones_k`; si se dispusiese de un único buzón, la máquina no sería capaz de coger el `k_transporte` que se encontrase en segunda posición si hubiese las suficientes piezas en su contenedor de almacenaje de fabricación asociado. A la salida de la `ficticia_unir_kanban` se identificará con una variable qué tipo de kanban de transporte se ha cogido del buzón. Para ello, a cada uno de los dos kanban de transporte se le ha asignado un tipo; de este modo, `k_transporte1` tendrá el `tipo1` y `k_transporte2` tendrá el `tipo2`. De esta forma, la variable que identifica la máquina `ficticia_unir_kanban` será `camino`, que toma valor 1 si el `tipo` de `k_transporte` es 1 y toma valor 2 si el `tipo` de éste es 2, como se puede apreciar en la Figura 2.15.

```

Edit INPUT RULE FOR MACHINE ficticia_unir_kanban
Select Search Editor Print
IF NParts (fabr1) = 5 AND NParts (buzon_k1) = 1
MATCH/ANY
pieza out of fabr1 #(5)k_transporte1 out of buzon_k1 #(1)
ELSEIF NParts (fabr2) = 5 AND NParts (buzon_k2) = 1
MATCH/ANY
pieza out of fabr2 #(5)k_transporte2 out of buzon_k2 #(1)
ELSE
Wait
ENDIF

Edit Actions On Input to Cycle For Machine ficticia_uni
Select Search Editor Print
IF k_transporte1:tipo = 1
camino = 1
ELSEIF k_transporte2:tipo = 2
camino = 2
ENDIF

```

Figura 2.15: acciones a la salida `ficticia_unir_kanban` en el modelo con kanban

Figura 2.14: regla de entrada `ficticia_unir_kanban` en el modelo con kanban

Una vez que se poseen las piezas y el kanban unidos, se llevarán a la zona de acabado mediante el `vehículo`, por la vía `hacia_final`. Cuando llega a la zona de acabado, kanban y el

lote de cinco piezas son una única pieza. Por tanto, será necesario tener una máquina ficticia que desensamble las piezas del kanban; es decir, por un lado, se tengan las cinco piezas que irán a los contenedores *alm1* o *alm2* y el kanban que irá al buffer *final\_ficticio1* o *final\_ficticio2*. Esta máquina, *ficticia\_salida*, produce las piezas del part anterior (cabe destacar que éste es el motivo por el que en la máquina *ficticia\_unir\_kanban* se indicó mantener el part de origen). Depositar las piezas y el kanban en su contenedor correspondiente vendrá dado por el valor de la variable *camino*, como se muestra en la Figura 2.16.

```

Edit OUTPUT RULE FOR MACHINE salida_ficticia
Select Search Editor Print
IF camino = 2
  PUSH k_transporte2 to final_ficticia2.pieza to alm2
ELSEIF camino = 1
  PUSH k_transporte1 to final_ficticia1.pieza to alm1
ELSE
  Wait
ENDIF
    
```

Figura 2.16: regla de salida *salida\_ficticia* en modelo con kanban

Como ya ocurriese en el modelo sin kanban, cuando en la *entrada\_final* hay menos de dos piezas, será necesario sacar un lote de cinco piezas de uno de los contenedores de la entrada a acabado; esta acción la llevará a cabo la máquina ficticia *ficticia\_separar\_piezas*, Figura 2.17. El hecho de sacar piezas de uno de los contenedores (las cinco piezas es la cantidad máxima de éstos) conllevará que tenga que actuar la máquina *hacia\_buzon\_accion* (para que se vuelvan a depositar piezas en el contenedor que ha quedado vacío). Después de haber colocado las piezas en la cinta *entrada\_final*, el proceso transcurre de la misma forma que en el modelo sin kanban.

```

Edit INPUT RULE FOR MACHINE ficticia_separar_piezas
Select Search Editor Print
IF NParts (alm1) = 5 AND NParts (entrada_final) < 2
  MATCH/ANY
  pieza out of alm1 #(5)
ELSEIF NParts (alm1) = 5 AND NParts (entrada_final) < 2
  MATCH/ANY
  pieza out of alm2 #(5)
ELSE
  Wait
ENDIF
    
```

Figura 2.17: regla de entrada *ficticia\_separar\_piezas* en modelo con kanban

Una vez descrito el funcionamiento de la zona de acabado, se procederá a explicar cómo trabaja la zona de fabricación, teniendo en cuenta que las necesidades de esta zona vendrán dadas por la zona de acabado. Como se ha comentado anteriormente, se disponen de dos contenedores en la zona de fabricación, *fabr1* y *fabr2*, con un stock máximo de cada uno de ellos de 5 piezas. Cuando en alguno de estos contenedores no haya piezas, será necesario fabricar. Para dar la orden de fabricación se dispone de un kanban de fabricación denominado *k\_fabrica*. La cantidad que hay que fabricar será la necesaria para que los contenedores de la zona de fabricación se llenen, y en la entrada de cada una de las máquinas haya cinco piezas (para que cuando se saque un lote de cinco piezas de los contenedores de fabricación, pase el menor tiempo posible hasta que éstos se llenen). El *k\_fabrica* puede encontrarse en dos buzones, *buzon\_almacen* y *buzon\_fabricar*, como se aprecia en la Figura 2.18. Dependiendo de si hay necesidad de fabricar, estará en un buzón u otro, como se explicará a continuación.

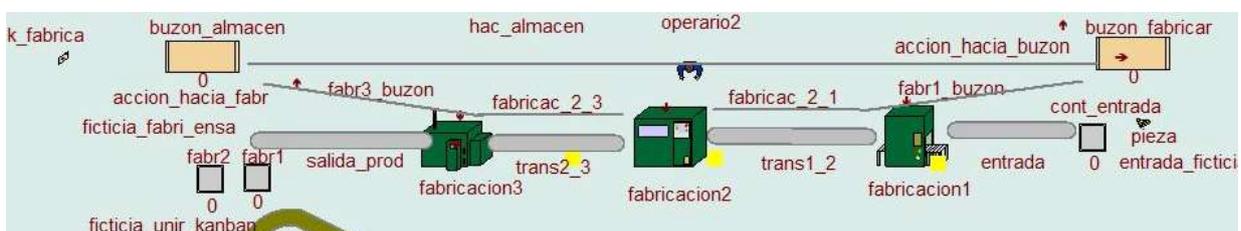
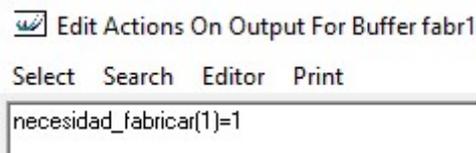


Figura 2.18: zona de fabricación modelo con kanban

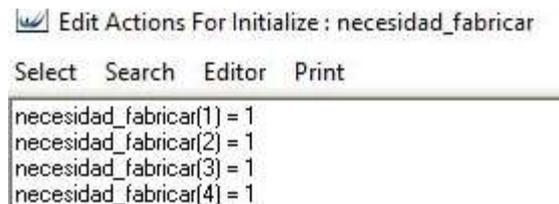
Existe una variable tipo vector, denominada `necesidad_fabricar()`, de dimensión cuatro, que almacena un dato asociado a cada máquina y a la entrada de piezas del proceso, es decir: `necesidad_fabricar(1)` asociado a `fabricacion3`, `necesidad_fabricar(2)` asociado a `fabricacion2`, `necesidad_fabricar(3)` asociado a `fabricacion1` y `necesidad_fabricar(4)` a la entrada de piezas al proceso, `entrada`. Cuando en los contenedores donde se almacena el stock de fabricación no haya las piezas suficientes, será necesario fabricar, es decir, la variable `necesidad_fabricar(1)` tomará valor 1, como se puede apreciar en la Figura 2.19. Además, en la inicialización del proceso esta variable toma valor 1 en todos sus componentes, para que sea necesario fabricar en todas las máquinas, Figura 2.20.



```

Edit Actions On Output For Buffer fabri1
Select Search Editor Print
necesidad_fabricar(1)=1
  
```

Figura 2.19: definición de `necesidad_fabricar(1)` en modelo con kanban

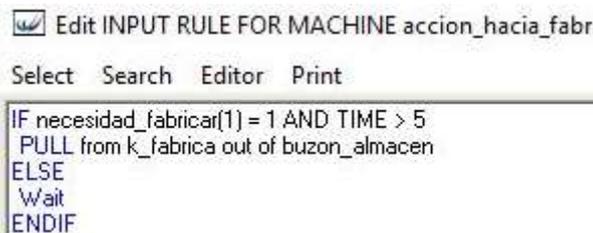


```

Edit Actions For Initialize : necesidad_fabricar
Select Search Editor Print
necesidad_fabricar(1) = 1
necesidad_fabricar(2) = 1
necesidad_fabricar(3) = 1
necesidad_fabricar(4) = 1
  
```

Figura 2.20: inicialización de la variable `necesidad_fabricar` en modelo con kanban

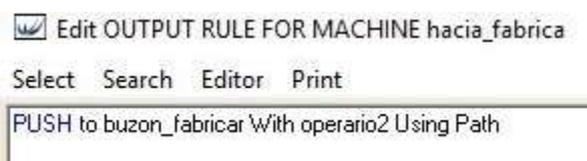
En el momento en el que la variable `necesidad_fabricar(1)` tome valor 1, será necesario llevar el `k_transporte` del `buzon_almacen` a `buzon_fabricar`. Esto se realizará mediante una máquina ficticia denominada `hacia_fabri`, como se aprecia en la Figura 2.21. Como en el caso del kanban de transporte, el operario que se encarga de las operaciones de mantenimiento de la zona de fabricación, el `operario2`, traslada el kanban por un camino definido como `hac_almacen` (cabe destacar que la operación `hacia_fabrica` se encuentra en el `buzon_almacen`, por lo que hay que indicar que el `operario2` use el camino en la regla de salida, Figura 2.22)



```

Edit INPUT RULE FOR MACHINE accion_hacia_fabri
Select Search Editor Print
IF necesidad_fabricar(1) = 1 AND TIME > 5
PULL from k_fabrica out of buzon_almacen
ELSE
Wait
ENDIF
  
```

Figura 2.21: regla entrada máquina ficticia `hacia_fabrica` en modelo con kanban

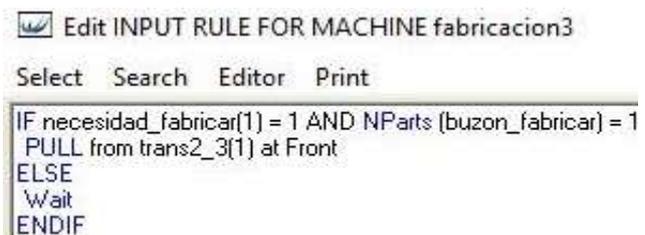


```

Edit OUTPUT RULE FOR MACHINE hacia_fabrica
Select Search Editor Print
PUSH to buzon_fabricar With operario2 Using Path
  
```

Figura 2.22: regla salida máquina ficticia `hacia_fabrica` en modelo con kanban

Como se ha comentado anteriormente, el funcionamiento de `fabricacion3` vendrá dado cuando la variable `necesidad_fabricar(1)` tome valor 1. Además, la necesidad de fabricar no sólo dependerá de esta variable, sino de que el kanban de fabricación esté en el `buzon_fabricar`, en ese momento, `fabricacion3` cogerá piezas de la cinta que la abastece, como se puede apreciar en la Figura 2.23. El hecho de que se ponga la condición de que el `k_fabricacion` deba estar en el buzón de fabricar es para que visualmente se observe que la orden de fabricar viene dada por el kanban.



```

Edit INPUT RULE FOR MACHINE fabricacion3
Select Search Editor Print
IF necesidad_fabricar(1) = 1 AND NParts (buzon_fabricar) = 1
PULL from trans2_3(1) at Front
ELSE
Wait
ENDIF
  
```

Figura 2.23: regla de entrada de `fabricacion3` en el modelo con kanban

Cuando **fabricacion3** está produciendo, será necesario que reciba piezas de **fabricacion2**. Por tanto, en **fabricacion3** hay que definir que **necesidad\_fabricar(2)** tome valor uno cuando no haya piezas en la cinta de entrada a **fabricación3**, como se indica en la Figura 2.24.

```

Edit Actions On Input to Cycle For Machine fabricacion3
Select Search Editor Print
IF NParts (trans2_3) = 0
necesidad_fabricar(2) = 1
ENDIF
    
```

Figura 2.24: orden de producción en fabricacion2 en el modelo con kanban

Las máquinas **fabricacion2**, **fabricacion1** y la cinta **entrada** poseen la misma regla de entrada y salida que las mostradas para **fabricacion3**, obviamente, cambiando las cintas de las que cogen las piezas y el índice de la variable **necesidad\_fabricar()**. De esta forma, **fabricación2** comenzará a coger piezas cuando la variable **necesidad\_fabricar(2)** tome valor 1.

La forma de disponer las piezas en los contenedores **fabr1** y **fabr2** vendrá por una máquina ficticia denominada **ficticia\_fabri\_ensa**, que depositará las piezas en uno u otro contenedor en función de si en los buzones hay kanban, dando prioridad al **k\_transporte1**, es decir, si hay necesidad de llevar piezas al contenedor **alm1** se depositarán las piezas en **fabr1**. Si no hay ningún kanban en los buzones, tendrá prioridad el contenedor **fabr1**, como se puede apreciar en la Figura 2.25

```

Edit OUTPUT RULE FOR MACHINE ficticia_fabri_ensa
Select Search Editor Print
IF NParts (buzon_k1) = 1
PUSH to fabr1
ELSEIF NParts (buzon_k2) = 1
PUSH to fabr2
ELSEIF NParts (fabr1) <5
PUSH to fabr1
ELSEIF NParts (fabr2) <5
PUSH to fabr2
ELSE
Wait
ENDIF
    
```

Figura 2.25: regla de salida máquina ficticia\_fabri\_ensa modelo con kanban

Ya se ha comentado cuándo hay que fabricar, pero también hay que indicar cuándo se deja de fabricar, es decir, en el momento en el que los contenedores **fabr1** y **fabr2** poseen el stock necesario y a la entrada de cada una de las máquinas haya 5 piezas. En el caso de **fabricacion3** dejará de producir cuando ya estén llenos los contenedores de fabricación o la cantidad que haya entre los contenedores y la cinta **salida\_fabrica** sea de 10 piezas (implica que los dos contenedores estén o estarán llenos sin necesidad de que **fabricacion3** produzca más piezas), como se aprecia en la Figura 2.26.

```

Edit Actions On Join For Conveyor salida_prod
Select Search Editor Print
IF NParts (salida_prod) + NParts (ficticia_fabri_ensa)+ NParts (fabr1)+ NParts (fabr2) = 10
necesidad_fabricar(1) = 0
ENDIF
    
```

Figura 2.26: parada de fabricacion3 en acción de entrada salida\_prod modelo con kanban

En el caso de las máquinas intermedias, éstas dejarán de fabricar piezas cuando en las cintas que las conectan haya 5 piezas (para garantizar el stock previo que debe de cada una de las máquinas). Por tanto, cuando en la cinta que une **fabricacion2** y **fabricacion1** haya cinco piezas, no será necesario que **fabricacion1** siga produciendo, por tanto **necesidad\_fabricar(3)** tomará valor 0, acción que se implementará en cada una de las cintas en la acción de entrada, como se puede apreciar en la Figura 2.27.

```

Edit Actions On Join For Conveyor trans1_2
Select Search Editor Print
IF NParts (trans1_2) = 5
necesidad_fabricar(3) = 0
ENDIF
    
```

Figura 2.27: acción a la entrada trans1\_2 modelo con kanban

Además, en la definición de acciones de entrada y salida de cada una de las máquinas se implementarán una serie de condiciones. En la regla de entrada, comprobar las piezas que quedan en la cinta que la abastece, si en ella quedan menos de 4 piezas, la máquina anterior tendrá que fabricar. Por ejemplo, si cuando *fabricacion2* coge una pieza de *trans1\_2*, en ésta quedan menos de 5 piezas, *necesidad\_fabricar(3)* tomará valor 1, Figura 2.28; en caso contrario, tomará valor 0. Además, en la regla de salida se definirá cuando deja de fabricar esa máquina que si en la cinta donde deja las piezas hay al menos cuatro piezas, esta máquina dejará de fabricar. Es decir, si en la cinta *trans2\_3* hay 4 piezas, cuando *fabricacion3* deje la pieza, habrá 5 en la cinta, haciendo que la variable *necesidad\_fabricar(2)* tome valor 0, Figura 2.29.

```

Edit Actions On Input to Cycle
Select Search Editor Print
IF NParts(trans1_2) < 5
neces_fabricar(2) = 1
ELSE
neces_fabricar(2) = 0
ENDIF

```

Figura 2.28: regla de entrada *fabricacion2* modelo con *kanban*

```

Edit Actions On Output from Cycle
Select Search Editor Print
IF NParts(trans2_3) < 4
neces_fabricar(2) = 1
ELSE
neces_fabricar(2) = 0
END IF

```

Figura 2.29: regla de salida *fabricacion2* modelo con *kanban*

Cuando los contenedores de la zona de fabricación se encuentran llenos, es decir, la variable *necesidad\_fabricar(1)* tome valor 0 y las cintas de la zona de fabricación posean 5 piezas cada una, es decir, todas las componentes del vector *necesidad\_fabricar()* tomen valor 0, no será necesario fabricar en ninguna de las máquinas. En ese momento, el *k\_fabrica* deberá situarse en el *buzon\_almacen*, para indicar que no se necesita fabricar más. La acción de llevar el *k\_fabrica* desde el *buzon\_fabricar* hasta *buzon\_almacen* la realiza una máquina ficticia denominada *hacia\_buzon*, que simulará el movimiento del *operario2* con el *k\_fabrica* de la zona de comienzo de fabricación al almacén de fabricación. La regla de entrada de esta acción se puede apreciar en la Figura 2.30.

```

Edit INPUT RULE FOR MACHINE hacia_buzon
Select Search Editor Print
IF necesidad_fabricar(1) = 0 AND necesidad_fabricar(2) = 0 AND necesidad_fabricar(3) = 0 AND necesidad_fabricar(4) = 0
PULL from k_fabrica out of buzon_fabricar
ELSE
Wait
ENDIF

```

Figura 2.30: regla de entrada *hacia\_buzon* modelo con *kanban*

En el momento en el que comience la simulación, como ocurriese en el modelo sin *kanban*, se desea que ya se encuentre en el estado periódico; por esa razón, en el inicio de la simulación en las cintas de entrada a las distintas máquinas ya se disponen de 5 piezas. De este modo, en el momento inicial (número de operaciones realizadas por cada una de las máquinas es cero) no es necesario esperar que *fabricacion1* empiece a producir para que *fabricacion2* posea piezas, lo que nos permite trabajar como deseamos desde el primer momento. Para ello, se ha implementado en las cintas la regla que se muestra en Figura 2.31.

```

Edit INPUT RULE FOR CONVEYOR trans1_2
Select Search Editor Print
IF NOps (fabricacion2) = 0 AND NParts (trans1_2) < 4 AND NParts (buzon_fabricar) = 1
PULL from pieza out of WORLD
ELSE
Wait
ENDIF

```

Figura 2.31: regla de entrada en el momento inicial para *trans1\_2* en el modelo con *kanban*

## 2. TEORÍA DE LAS LIMITACIONES

### 2.1. DESCRIPCIÓN DEL MODELO CON CUELLO DE BOTELLA

El modelo simula un proceso productivo en el que una pieza principal pasa por una serie de operaciones para, posteriormente, ensamblarla con otra. Una vez obtenido el conjunto, éste continúa el proceso. En la Figura 2.32 se puede apreciar cómo es el proceso productivo con las piezas que interfieren, las distintas máquinas por las que deben pasar éstas, las cintas que llevan las piezas de una a otra máquina, los operarios necesarios para realizar las operaciones de mantenimiento de las máquinas (cada uno se encargará de cuatro máquinas que se encuentran delimitadas por las líneas grises), así como los caminos que deben usar para moverse entre las máquinas. La salida de piezas del proceso se realizará cuando en la cinta *salida\_proceso* haya al menos ocho piezas.

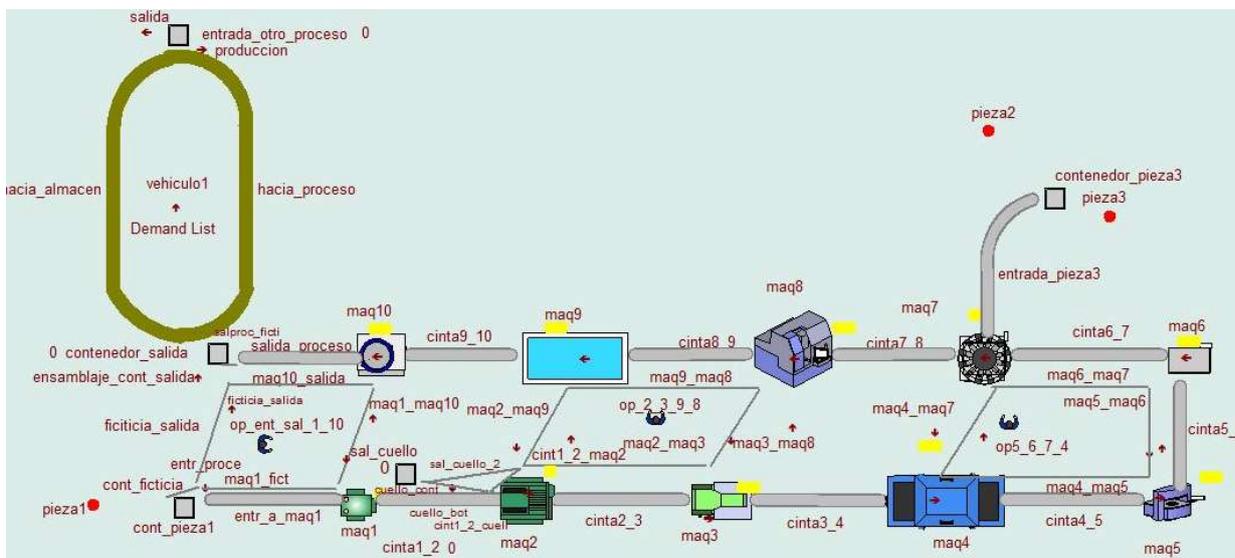


Figura 2.32: representación del modelo con cuello de botella

Al modelo llegan dos piezas: *pieza1*, la pieza principal del proceso, a la cual se le realizan una serie de operaciones y que se ensamblará con la *pieza3*. Al conjunto se le denominará *pieza2*. La razón de generar una nueva pieza es por la visualización en 3D. Debido a que si a la salida de la máquina de ensamblaje (*maq7*) se cambia el icono para simular un ensamblaje, el color de éste es el asignado por defecto, no el color que poseen *pieza1* y *pieza3*. La visualización de la *pieza1* se puede apreciar en la Figura 2.33, la *pieza3* según la Figura 2.34 y el conjunto, *pieza2*, según la Figura 2.35.

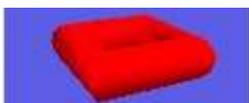


Figura 2.33: representación 3D pieza1 modelos teoría de las limitaciones



Figura 2.34: representación 3D pieza3 modelos teoría de las limitaciones

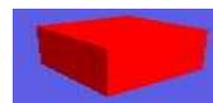


Figura 2.35: representación 3D pieza2 modelos teoría de las limitaciones

Los tiempos de ciclo, así como la prioridad de las máquinas para el operario a la hora de realizar las operaciones de mantenimiento, se pueden apreciar en la Tabla 2.2 (incluyendo el tiempo de ciclo de las máquinas ficticias, entrada y salida de piezas del proceso). Como se puede ver, existe dos máquinas que poseen un elevado tiempo de ciclo, *maq2* y *ficticia\_salida*, aunque esta última realiza la operación cada 8 piezas, lo que implica que no es el cuello de botella. Por tanto, se llega a la conclusión de que el cuello de botella del proceso es la *maq2*.

Máquina	Tiempo de ciclo(s)	Prioridad	Mantenimiento (time between failures)
Entr_proce	6	2	
Maq1	10	4	700
Maq2	50	3	960
Maq3	6	4	800
Maq4	2	3	1000
Maq5	13	5	650
Maq6	6.5	2	800
Maq7	4.5	4	600
Maq8	2.5	2	1200
Maq9	3	3	900
Maq10	6	3	800
Ficticia_salida	32	1	

Tabla 2.2: tiempos de ciclo así como prioridad en los modelos teoría de las limitaciones

La entrada de la pieza principal al proceso, **pieza1**, la realiza el **op\_ent\_sal\_1\_10** cogiéndola del **cont\_pieza1** y situándola en **entrada\_a\_maq**, en donde se ha implementado una variable denominada **piezas\_entrada**, para conocer cuántas piezas entran en el proceso. Para poder realizar esta actividad, ha sido necesario crear una máquina ficticia, denominada **entr\_proce**, realizándose esta operación en 6 s. Aunque el tiempo de ciclo de **entr\_proce** sea de 1s, el operario debe realizar el movimiento desde el **cont\_pieza1** hasta la máquina ficticia para dejar la pieza (dado que **entr\_proce** se encuentra en el comienzo de **entrada\_a\_maq**), realizándose estos movimientos por el path **cont\_ficticia**. Como se puede apreciar en la Figura 2.32, **op\_ent\_sal\_1\_10** realiza tanto esta operación como la salida de piezas del proceso. Por esta razón, se ha implementado una variable **neces\_salida\_proceso** (que se desarrollará al describir la salida de piezas del proceso), para que cuando **op\_ent\_sal\_1\_10** esté realizando la operación de salida del proceso (la variable tomará valor 1), no realice la operación de introducir piezas al proceso. En consecuencia, la regla de entrada de la máquina ficticia es la que se muestra en la Figura 2.36.

```

Edit INPUT RULE FOR MACHINE entr_proce
Select Search Editor Print
IF neces_salida_proceso = 0
  PULL from cont_pieza1 With op_ent_sal_1_10#(1) Using Path
ELSE
  Wait
ENDIF

```

Figura 2.36: regla de entrada de piezas al proceso modelo con cuello de botella

En las **maq2, 3, 4, 5 y 6**, se realizan operaciones a la **pieza1** según los tiempos de ciclo indicados en la Tabla 2.2. Además, estas máquinas necesitan unas tareas de mantenimiento originadas por las averías que se han programado (indicadas el tiempo en la misma tabla), éstas deben de ser llevadas a cabo por los operarios. El operario realizará las tareas de mantenimiento que le corresponda según en la célula en la que se encuentre (Figura 2.32, zona delimitada por las líneas grises que conectan las máquinas), además los trayectos de los operarios se realizan por path (líneas grises que delimitan las zonas). Así, cuando se tenga que realizar una operación de mantenimiento, el operario se desplazará de una máquina a otra por los distintos paths que forman su célula, indicándose en la regla de breakdowns la necesidad de usar los paths, como se puede apreciar en la Figura 2.37. En el caso de que el operario se encuentre en una máquina, y exista la necesidad de realizar operaciones de mantenimiento en otra que no se encuentra consecutiva, el operario se desplazará por los distintos paths

hasta llegar a la máquina que debe ir (como se ha comentado anteriormente, éstos forman una célula y el operario siempre irá por el trayecto más corto). Se ha establecido un tiempo de desplazamiento diferente entre las máquinas, dependiendo si el movimiento se realiza de forma horizontal (máquinas consecutivas) o en vertical (máquinas que se encuentran paralelas), como se muestra en las Figura 2.38 y Figura 2.39.

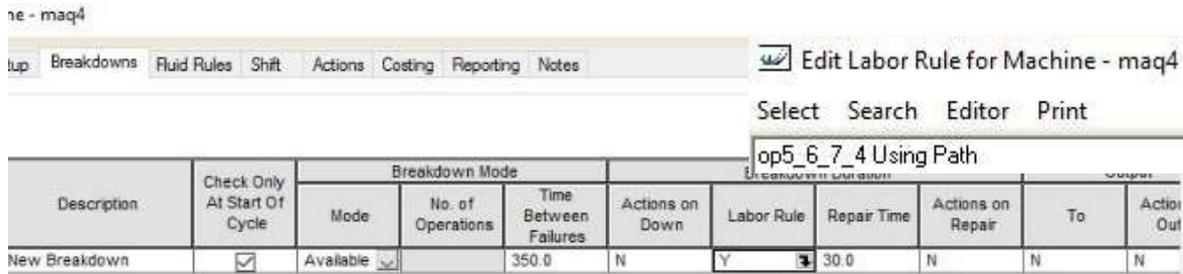


Figura 2.37: definición de path en averías en los modelos teoría de las limitaciones

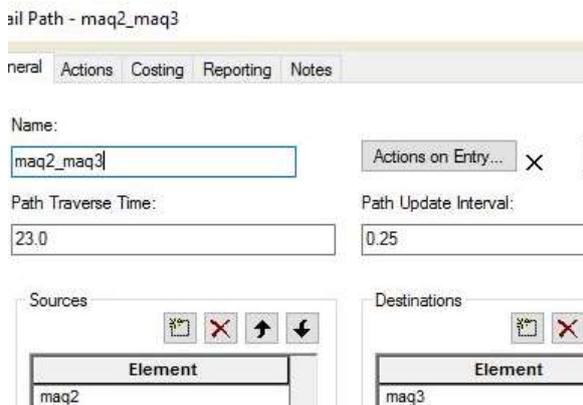


Figura 2.38: recorrido operario entre máquinas consecutivas modelos teoría de las limitaciones

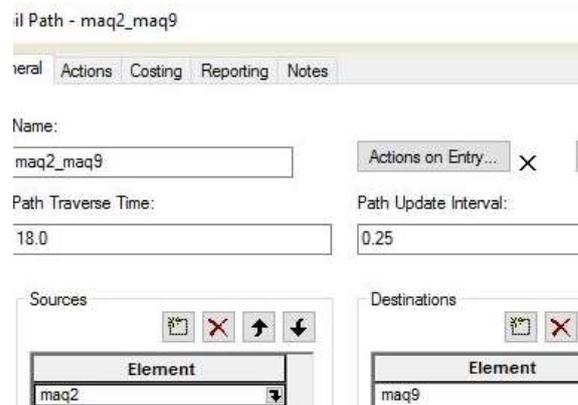


Figura 2.39: recorrido operario entre máquinas en paralelo modelos teoría de las limitaciones

Para evitar que la **maq1** se colapse al seguir produciendo piezas sin que la **maq2** pueda procesarlas debido a su elevado tiempo de ciclo, se posee una máquina ficticia denominada **cuello\_bot**. Esta actividad la realiza el **op\_2\_3\_9\_8**, haciendo que saque las piezas del proceso, cuando en la **cinta1\_2** hay una cantidad superior a 5 piezas, como indica la Figura 2.40, situándolas en el contenedor **sal\_cuello**. Además, a esta máquina ficticia se la ha establecido prioridad 1, para que el operario siempre vaya a ella cuando hay necesidad de sacar piezas y no se acumulen, haciendo que la **maq1** no pueda saturarse.

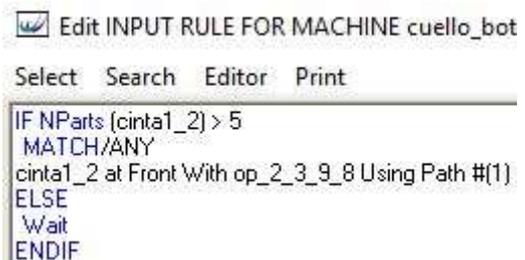


Figura 2.40: salida de piezas cinta1\_2 modelo con cuello de botella

Como se ha comentado anteriormente, los operarios deben realizar los recorridos entre las máquinas por paths. En el caso de sacar piezas del cuello de botella se han definido varios paths, como se puede apreciar en la Figura 2.41. Por tanto, cuando el operario tenga que sacar piezas de la cinta, irá desde la **maq2** hasta la **cinta1\_2**, usando el path **cint1\_2\_maq2** (Figura 2.42). Posteriormente, usará el path **cint1\_2\_cuell** para situarse en **cuello\_bot** (Figura 2.43), llevará la pieza que ha sacado de la cinta por el path **cuello\_cont** (Figura 2.44) para depositarlo en el contenedor **sal\_cuello**. En este contenedor se ha implementado una variable **piezas\_cuello** para conocer el número de piezas que son necesarias sacar del proceso. Una vez

que ha depositado la pieza en el contenedor, puede ocurrir que sea necesario sacar más piezas, haciendo que use los paths descritos anteriormente. En el caso de que no sea necesario sacar más piezas de la *cinta1\_2*, y haya alguna máquina que necesite operación de mantenimiento, el *operario\_2\_3\_8\_9* tendrá que ir por el path *sal\_cuello\_2* (Figura 2.45), en caso contrario, irá a su posición de reposo.

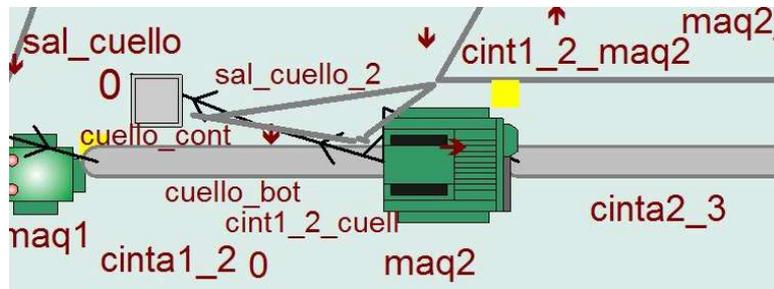


Figura 2.41: paths necesarios en la operación de sacar piezas del cuello de botella en el modelo con cuello de botella

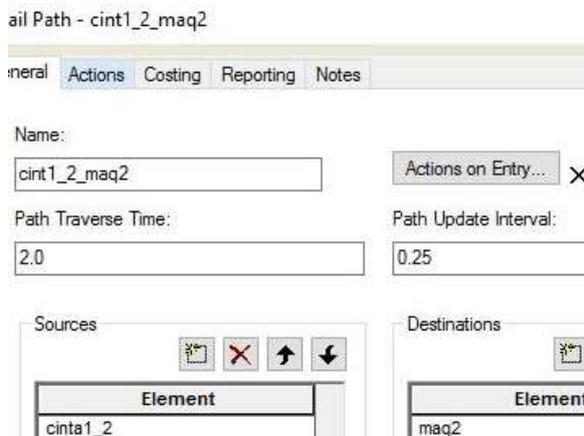


Figura 2.42: descripción del path *cint1\_2\_maq2* en el modelo con cuello de botella

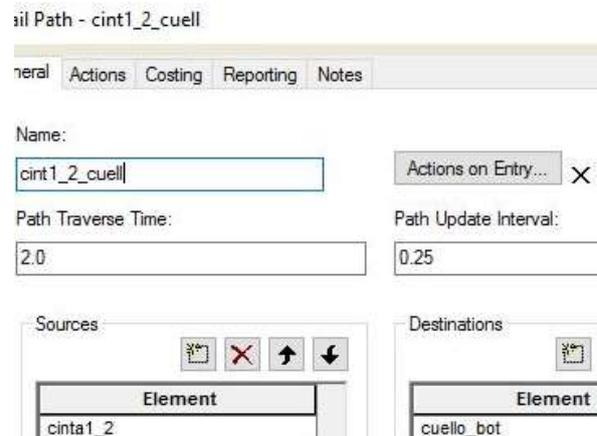


Figura 2.43: descripción del path *cint1\_2\_cuell* en el modelo con cuello de botella

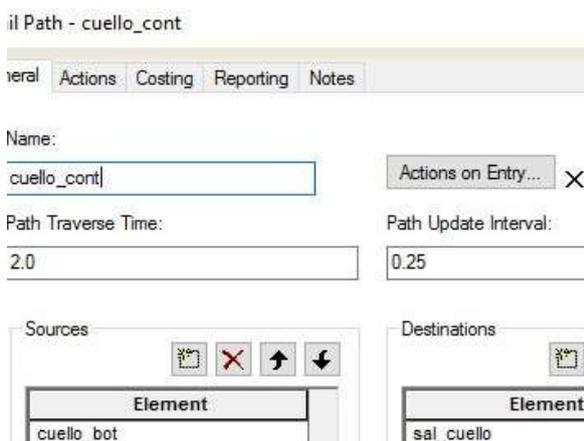


Figura 2.44: descripción del path *cuello\_cont* en el modelo con cuello de botella

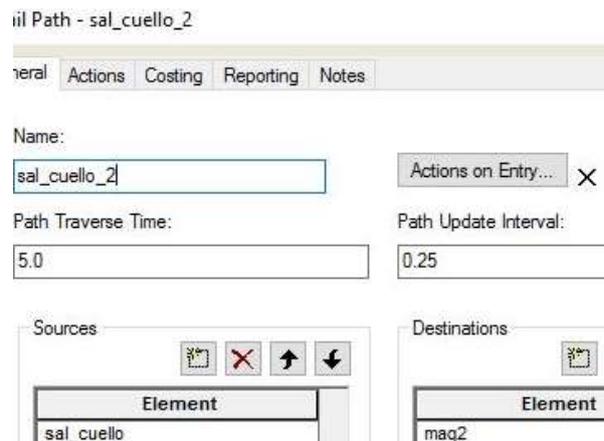


Figura 2.45: descripción del path *sal\_cuello\_2* en el modelo con cuello de botella

En la *maq7* se realiza el ensamblaje de la *pieza1* con la *pieza3*. La *pieza3* llega a través de una cinta denominada *entrada\_pieza3*, la cual recoge las piezas de un *contenedor\_pieza3*. La regla para unir estas piezas es la que se muestra en la Figura 2.46, llegando la *pieza1* por la *cinta6\_7* y la *pieza3* por la cinta *entrada\_pieza3*. Como se ha comentado al describir las piezas que van a intervenir en el modelo, la salida de la máquina ensamblaje produce un conjunto, *pieza2*,

como se puede ver en la Figura 2.47. Al indicar que a la salida de esta máquina todo cambie a **pieza2**, se hace que en el modelo 3D el producto de esta máquina sea la representación en 3D de la **pieza2**.

 Edit INPUT RULE FOR MACHINE maq7

Select Search Editor Print

```
MATCH/ANY
cinta6_7 at Front #(1)entrada_pieza3 at Front #(1)
```

Figura 2.46: ensamblaje en los modelos teoría de las limitaciones

 Edit Actions On Finish Cycle For Machine maquina7

Select Search Editor Print

```
CHANGE ALL to pieza2
```

Figura 2.47: salida ensamblaje modelos teoría de las limitaciones

La salida de las piezas de la **maq10** y, por tanto, la salida del proceso, se hace mediante la **salida\_proceso**, cuando en ésta hay al menos 8 piezas. En esta cinta se ha implementado una variable denominada **piezas\_salida**, para conocer el número de piezas que llegan al final del proceso. Como en el caso de la entrada al proceso, la actividad de sacar piezas del proceso la realizará el **op\_ent\_sal\_1\_10** mediante una máquina ficticia, denominada **ficticia\_salida**, tratándose de una máquina general (no es necesario que en **salida\_proceso** haya 8 piezas, dado que coge las piezas de **salida\_proceso** y las pone en espera de entrada de **ficticia\_salida** hasta que no haya 8 piezas y realice la operación).

Al tratarse de una máquina general, se ha decidido implementar en la cinta **salida\_proceso** una variable para conocer las piezas que hay en la cinta, aunque éstas ya estén en la entrada de **ficticia\_salida**. Esta variable es **entrada\_salida**, que aumenta su valor cada vez que entra una pieza en la cinta, Figura 2.48. Cuando esta variable posea un valor mínimo de 8, será necesario que el operario realice la acción de sacar piezas del proceso, haciendo que la variable **neces\_salida\_proceso** tome valor 1, como indica la Figura 2.48. En el caso de que no haya el número mínimo de piezas, no será necesario sacar piezas y, por tanto, la variable **neces\_salida\_proceso** tomará valor 0 y el operario podrá introducir piezas al proceso como se indica la Figura 2.36. Esta variable se ha definido para evitar que el **op\_ent\_sal\_1\_10** esté continuamente realizando el trayecto entre la entrada y la salida de las piezas, dado que si a la salida se la da prioridad 1, siempre que llegue una pieza a la cinta **salida\_proceso**, el operario se movería hasta allí (a través de los distintos paths), gastando un elevado tiempo; por eso, se ha decidido que solo coja piezas de la **cinta\_proceso**, cuando la variable **neces\_salida\_proceso** tome valor 1, Figura 2.49. Cuando acabe la actividad, la variable tomará valor cero si no hay el mínimo de piezas en **salida\_proceso**, haciendo que el **op\_ent\_sal\_1\_10** realice otra actividad. Además, se disminuirá de la variable **entrada\_salida** tantas veces como piezas salen de la máquina, es decir 8, como se puede apreciar en la Figura 2.50

 Edit Actions On Join For Conveyor salida\_proceso

Select Search Editor Print

```
entrada_salida = entrada_salida + 1
IF entrada_salida >= 8
neces_salida_proceso = 1
ELSE
neces_salida_proceso = 0
ENDIF
```

Figura 2.48: salida de piezas del proceso en los modelos teoría de las limitaciones

 Edit INPUT RULE FOR MACHINE ficticia\_salida

Select Search Editor Print

```
IF neces_salida_proceso = 1
MATCH/ANY
salida_proceso at Front With op_ent_sal_1_10 Using Path #(1)
ELSE
Wait
ENDIF
```

Figura 2.49: acción sacar piezas de los modelos teoría de las limitaciones

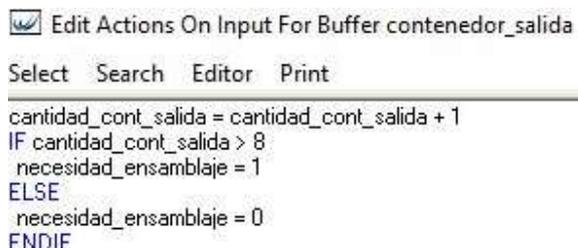
 Edit Actions On Finish Cycle For Machine ficticia\_salida

Select Search Editor Print

```
entrada_salida = entrada_salida - 1
neces_salida_proceso = 0
```

Figura 2.50: regla salida de piezas del proceso modelos teoría de las limitaciones

Después de haber sacado piezas de la cinta `salida_proceso`, será necesario sacar piezas del `contenedor_salida`, dado que, sino éste se llenaría empeorando la visualización y además no representando un proceso real, puesto que estas piezas formarán parte de otro proceso o saldrían del proceso como pieza acabada. En este caso se ha supuesto que las piezas se van a llevar a otro proceso. Para ello se dispone de unas vías por las que circula el `vehiculo1` denominadas `hacia_proceso` y `hacia_almacen`. Como ocurriese en el modelo con kanban, a la hora de trasladar las piezas, éstas deben de ser una única pieza, dado que el `vehiculo1` solo puede llevar una pieza, por tanto, se ha creado una máquina ficticia denominada `ensamblaje_cont_salida`. Como ocurre en el caso anterior, se ha definido en el contenedor una variable denominada `cantidad_cont_salida` que nos determinará cuantas piezas han entrada al `contenedor_salida`, como se aprecia en la Figura 2.51, para saber cuándo es necesario ensamblar las piezas (dado que la máquina ficticia coge las piezas aunque en el contenedor no haya las 8 piezas, las coloca en espera dentro de la máquina). Por tanto, cuando la variable `cantidad_cont_salida` tome un valor mayor a 8, `necesidad_ensamble` tomará valor 1 y se realizará el ensamble, como indica la Figura 2.52. Cuando se acabe la acción del ensamble, se disminuirá en 8 unidades el valor de `necesidad_ensamblaje` y la variable `necesidad_ensamblaje` tomará valor 0.

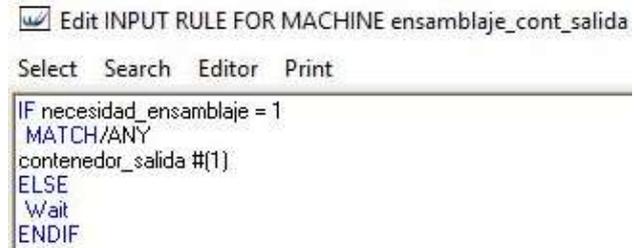


```

Edit Actions On Input For Buffer contenedor_salida
Select Search Editor Print
cantidad_cont_salida = cantidad_cont_salida + 1
IF cantidad_cont_salida > 8
necesidad_ensamblaje = 1
ELSE
necesidad_ensamblaje = 0
ENDIF

```

Figura 2.51: necesidad de sacar piezas del contenedor de salida del proceso modelos teoría de las limitaciones



```

Edit INPUT RULE FOR MACHINE ensamblaje_cont_salida
Select Search Editor Print
IF necesidad_ensamblaje = 1
MATCH/ANY
contenedor_salida #(1)
ELSE
Wait
ENDIF

```

Figura 2.52: acción de ensamblar piezas para el transporte en el modelos teoría de las limitaciones

Cuando llegue la pieza al final del recorrido del `vehículo1`, será necesario volver a tener las 8 piezas (que se han ensamblado para realiza el trayecto), por tanto, se ha definido una máquina de producción para volver a tener las 8 piezas, denominándose a esta máquina `producción`, depositándolas en el contenedor `entrada_otro_proceso`.

En el contenedor `entrada_otro_proceso` se van a almacenar las piezas que transporta el `vehículo1`, habiendo pasado por la máquina ficticia `producción`. Como se ha comentado anteriormente, estas piezas pueden dedicarse a otro proceso que se encontrará a continuación o a un proceso exterior. Se ha supuesto que cuando en `entrada_otro_proceso` haya 10 piezas implica que el otro proceso que hay a continuación no tiene demanda de esas piezas y se van a sacar al exterior 10 piezas del modelo hacia `ship`, mediante una máquina ficticia denominada `salida`.

## 2.2. IMPLEMENTACIÓN DE TÉCNICA LEAN (CAPACITARIO)

En el apartado anterior se genera un stock intermedio a la entrada de la `maq2`, almacenado en `cuell_cont` que sólo tendría salida en los casos en los que la `maq1` estuviese parada, pudiendo ser por mantenimiento o por parada programada. Por tanto, se posee una gran cantidad de piezas a medio procesar con el valor de capital inmovilizado que eso conlleva, el espacio que ocupa, así como la posibilidad de que estas piezas se deterioren.

Para poder procesar la cantidad de piezas que se disponen a la entrada de la `maq2` sería necesario realizar un aumento capacitivo de la `maq2`, aumentando en una unidad esta

máquina. De esta forma, ya no es necesario sacar piezas de la *cinta1\_2* para que la *maq1* se colapse al no poder procesar las piezas la *maq2*, como se puede apreciar en la Figura 2.53. Cabe destacar que el resto del proceso es igual que el que se ha comentado anteriormente. Por tanto, a continuación sólo se procederá a comentar cómo se ha realizado el capacitario de la *maq2*, los elementos que se han añadido para que el proceso funcione correctamente.

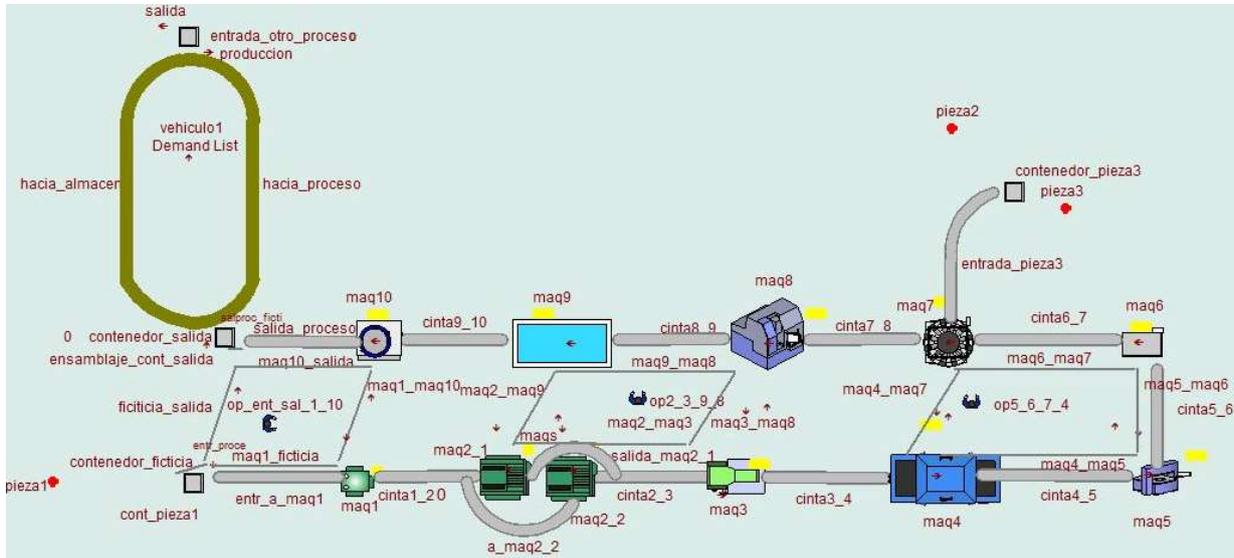


Figura 2.53: representación del modelo capacitario

Como se puede apreciar en la Figura 2.54, la *cinta1\_2* puede disponer las piezas a dos máquinas, en *maq2\_1* o en *maq2\_2*. Por tanto, en la regla de salida de dicha cinta, se ha definido que se disponga la *pieza1* en el transportador *a\_maquina2\_2* o directamente a *maq2\_1*, dependiendo de si la *maq2\_1* o la *maq2\_2* están procesando piezas (es decir, el número de piezas en esas máquinas es cero o no), como se puede apreciar en la Figura 2.54

```

Edit OUTPUT RULE FOR CONVEYOR cinta1_2
Select Search Editor Print
IF NParts (maquina2_1) = 0
  PUSH to maquina2_1
ELSEIF NParts (maquina2_2) = 0
  PUSH to a_maquina2_2 at Rear
ELSE
  Wait
ENDIF
    
```

Figura 2.54: regla de salida de cinta1\_2 en el modelo capacitario

Será necesario disponer de una cinta que lleve las piezas que se procesan en la *maq2\_1* a la *cinta2\_3*, se podía haber hecho que las piezas se dispusieran en la *cinta2\_3* sin la necesidad de esta cinta, pero el modelo en 3D no quedaría real. Por esta razón, se ha creado la cinta *salida\_maq2*. Además, se ha dispuesto de un nuevo path para conectar *maq2\_1* y *maq2\_2*, denominado *maqs*, y también será necesario modificar el path *maq2\_3*, que ahora conecta la *maq2\_2* con la *maq3*, y el path *maq2\_9*, que une la *maq9* con la *maq2\_1*.

Además, otra condición que se ha implementado en el modelo capacitivo, es la posibilidad de dejar de producir en la *maq1* cuando con las *maq2\_1* y *maq2\_2* no se puedan procesar las piezas que se generan en la *maq1*, previamente habiendo comprobado que con estas dos máquinas se satisface la demanda. Esta condición hace que cuando en la *cinta1\_2* haya la suficiente cantidad de piezas para satisfacer la demanda de las *maq2\_1* y *maq2\_2*, la *maq1* no coja piezas de la cinta *entrada\_a\_maq1*. Este caso, se aplicaría cuando se ha comprobado que con el elemento capacitivo se tiene satisfecha la demanda y se puede dejar de fabricar. Esta regla se puede apreciar en la Figura 2.55

```

Edit INPUT RULE FOR MACHINE Maquina1
Select Search Editor Print
IF NParts (cinta1_2) < 8
  PULL from entrada_a_maq1(1) at Front
ELSE
  Wait
    
```

Figura 2.55: regla de entrada a la maq1 modelo capacitario

Además, el `op_sal_ent_1_10` dejará de introducir piezas en el modelo cuando entre la `cinta1_2` y `entrada_a_maq1` haya al menos 13 piezas, como se puede apreciar en la regla de entrada de `entr_proce`, Figura 2.56. De esta forma, se garantiza que hay piezas a la entrada de la `maq1`, `maq2_1` y `maq2_2`, y que se satisface la demanda de `pieza2`.

```
✎ Edit INPUT RULE FOR MACHINE entr_proce
Select Search Editor Print
IF neces_salida1 = 0 AND NParts (cinta1_2) + NParts (maq1) + NParts (entr_a_maq1) + NParts (entr_proce) < 13
  PULL from cont_pieza1 With op_ent_sal_1_10#(1) Using Path
ELSE
  Wait
ENDIF
```

Figura 2.56: regla de entrada de la maquina `entr_proce` en modelo capacitario



## Capítulo 3.: DESCRIPCIÓN DE LOS MODELOS II



A continuación, se van a explicar las técnicas poka yoke, célula en U y Heijunka, asociadas a técnicas Lean que afectan en menor medida al nivel de stock, pero que tienen mayor impacto en la calidad y en los tiempos de los procesos.

Primero, se explicará la técnica poka yoke. Para ello, se construirá un modelo que no posee poka yoke, generándose gran cantidad de piezas defectuosas. Posteriormente, se aplicará un poka yoke de advertencia, lo que permitirá descubrir fácilmente los defectos y reducir la cantidad de piezas defectuosas. Finalmente, se implementará un poka yoke de fiabilidad-control, garantizando que todas las piezas que se generan posean la calidad deseada.

A continuación, se explicará un proceso en el que los operarios acompañan a las piezas, estando las piezas dispuestas de forma lineal. Aplicando técnica Lean, el proceso será en forma de célula en U, disminuyéndose los tiempos de desplazamiento de los operarios.

Para finalizar, se describirá un proceso que trabaja con varias piezas, usando grandes lotes de piezas para formar conjuntos en función de las necesidades de unos clientes, tardando grandes tiempos en ser atendidos. Aplicando las técnicas Heijunka y Smed, se trabajarán con lotes más pequeños, tardando menos tiempo los clientes en ser atendidos.

## 1. TÉCNICA POKA YOKE

### 1.1. DESCRIPCIÓN DEL MODELO SIN POKA YOKE

Se dispone un proceso productivo en el que se generan dos tipos de piezas (**tipo0** y **tipo1**), teniendo en común una serie de operaciones para, posteriormente, dividir el proceso en dos tramos dependiendo del tipo de pieza que se realiza, como se puede apreciar en Figura 3.1.

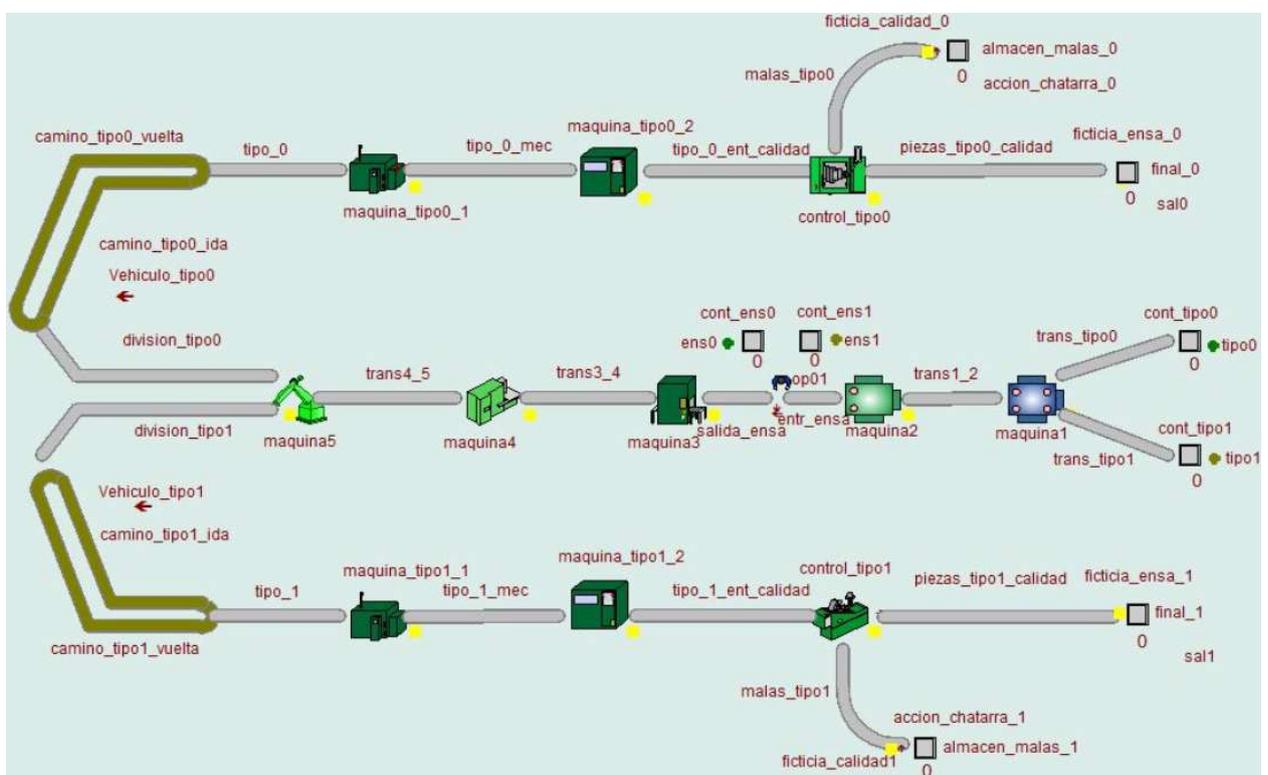


Figura 3.1: representación del modelo sin poka yoke

A la entrada de la zona común llegan dos tipos de piezas, a las cuales se realizan una serie de operaciones en las que se van aportando cierta calidad (en cada máquina una calidad diferente); además, a la salida de la *maquina2* se realizará el ensamblaje con otra pieza. Este ensamblaje dependerá del tipo de pieza que llegue a la máquina de ensamblaje (*ficticia\_emsable*, situada entre *entr\_ensa* y *salida\_ensa*) teniendo que coger su correspondiente de los *cont\_ens0* y *cont\_ens1*, es decir, *tipo0* con *ens0* y *tipo1* con *ens1*.

En la *maquina5* se produce una clasificación de los conjuntos creados en función de su tipo, *tipo0* y *tipo1*, para continuar realizando a cada tipo de pieza sus operaciones características, disponiéndose las piezas de *tipo0* en la parte superior y en la parte inferior las piezas de *tipo1*. En las zonas correspondientes a cada tipo de pieza, se continúa generando una determinada calidad y al final del proceso se posee un control final de las piezas para comprobar que poseen la calidad mínima requerida, que el tipo en esa zona es el correcto y que el ensamblaje se ha realizado de forma correcta. Las piezas que poseen las calidades deseadas irán a los contenedores final de cada una de las zonas, y las que no poseen la calidad se dispondrán en almacén malas para posteriormente desecharlas.

Como se ha comentado anteriormente, a la zona común llegan dos tipos de pieza, *tipo0* y *tipo1*, que se depositarán en su correspondiente contenedor, *cont\_tipo0* y *cont\_tipo1*. Se han definido tres atributos a las piezas: dos asociados al tipo y un tercero asociado a la calidad que va a poseer la pieza. En cuanto a los atributos asociados al tipo de pieza, uno se usa para determinar qué *tipo* de pieza es, si *tipo0* o *tipo1*, y otro asociado al ensamblaje para formar el conjunto, el cual se denomina *tipo\_acople*. Este atributo determinará si la pieza se ensambla correctamente, es decir, *tipo0* con *ens0* o si, por el contrario, se ensambla *tipo0* con *ens1*. Por tanto, a la pieza *tipo0* se la ha asociado el valor 0 para el atributo *tipo*. Además, si un número aleatorio entre el 0 y el 1 toma un valor menor de 0.85, el valor que tome el atributo *tipo\_acople* será el mismo el atributo *tipo*; es decir, 0 cuando el ensamblaje es correcto. En caso contrario, el valor aleatorio sea mayor de 0.85 y el valor del atributo *tipo\_acople* tomará valor 1, siendo el ensamblaje el erróneo, Figura 3.2. En el caso de piezas *tipo1* el procedimiento es el mismo, *tipo\_acople* tomará valor 1 y el ensamblaje será correcto. El tercer atributo que poseen las piezas será *calidad*, cuyo valor irá modificándose en el transcurso del proceso productivo haciendo que una pieza sea defectuosa o no.

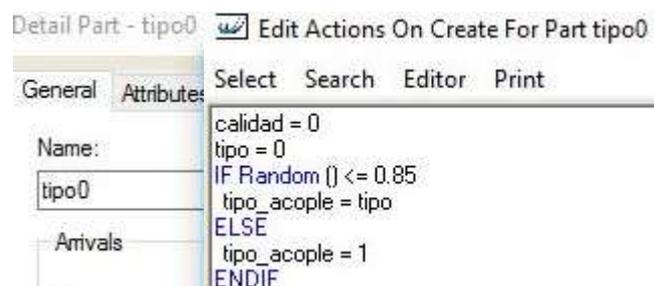


Figura 3.2: definición atributos *tipo0* modelo sin poka yoke

Para poder distinguir en todo momento, tanto en el modelo en 3D como en el 2D, de qué tipo de pieza se trata, la representación de cada pieza es diferente en función del *tipo*, apreciándose el *tipo0* en la Figura 3.3 y el *tipo1* Figura 3.4.

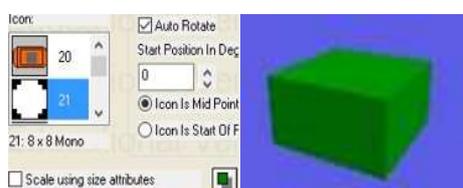


Figura 3.3: representación 3D de la pieza *tipo0* en los modelos técnica poka yoke

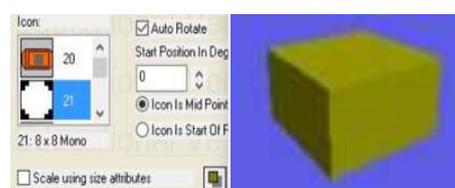


Figura 3.4: representación 3D de la pieza *tipo1* en los modelos técnica poka yoke

Estas piezas se dispondrán en su contenedor correspondiente. Posteriormente, irán a las cintas que las situarán a la entrada de la **maquina1**, como se puede apreciar en la Figura 3.1. A la entrada de estas cintas se disponen de sendas variables denominadas **entrada0** y **entrada1**, incrementándose las variables cada vez que entra una pieza a la cinta, para conocer el número de piezas que entran en el proceso de cada tipo, para hacer una comparación entre los distintos modelos. Como se ha comentado anteriormente, a las piezas se les aporta una determinada calidad en cada máquina. Cabe destacar que, si en una operación no se le proporciona un valor mínimo de calidad, esta pieza será defectuosa y no podrá llegar a la salida del proceso. En la Tabla 3.1 se puede apreciar los valores mínimos de calidad de cada una de las máquinas, así como sus tiempos de ciclo.

Máquina	Tiempo de ciclo (s)	Calidad mínima admisible	Calidad adquirida
Maquina1	12	3	Tnormal(5,1.2,3.2,6.8)
Maquina2	13	7.7	Tnormal(5.5,1.2,3.7,7.3)
Maquina3	10	13	Tnormal(5,3,4.1,7.2)
Maquina4	7	16.5	Tnormal(5,2.5,3.5,7.5)
Maquina5	8		
Ficticia_ensamble	4		

Tabla 3.1: tiempo de ciclo y calidad mínima zona común modelo sin poka yoke

Los tiempos de ciclo, así como la calidad mínima, de cada una de las máquinas en la zona asociada a cada tipo, se puede apreciar en la Tabla 3.2, siendo las mismas en la línea de **tipo0** y **tipo1**.

Máquina	Tiempo de ciclo (s)	Calidad mínima admisible	Calidad adquirida
Maquina_tipo0_1	10	22	Tnormal(4.5,0.8,3,6)
Maquina_tipo0_2	15	26.5	Tnormal(5.3,0.9,3.3,7.3)
Control_tipo0	10		

Tabla 3.2: tiempo de ciclo y calidad mínima zona específica modelo sin poka yoke

En las acciones de salida de cada una de las máquinas, se ha definido la calidad que debe poseer las piezas a la salida de éstas, así como una condición para garantizar la identificación de las piezas que no poseen la calidad suficiente, como se puede apreciar en Figura 3.5. Esta condición se ha realizado para que en la pieza, en los casos en los que no posee la calidad mínima a la salida de esa máquina, cambie el icono que poseía por el icono 44, el cual se ha definido según las características que se pueden apreciar en la Figura 3.6. Además de cambiar el icono, se cambia la forma que tomaría en el modelo en 3D ese icono, de forma que se puede apreciar rápidamente qué pieza no posee la calidad necesaria.

```

Edit Actions On Finish Cycle For Machine maquina1
Select Search Editor Print
calidad = calidad + TNormal (5,1.2,3.2,6.8)
calidad_1 = calidad
IF calidad_1 < 3.5
ICON = 44
ENDIF
    
```

Figura 3.5: acción a la salida de la maquina1 en el modelo sin poka yoke

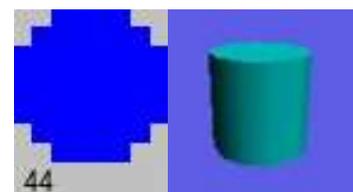


Figura 3.6: representación de las piezas sin la calidad mínima en los modelos técnica poka yoke

Una vez que se dispone la pieza en la salida de la *maquina2* (*entrada\_ensa*), el operario *op01* se encarga de unir la pieza que ya ha sido procesada en las *maquinas1* y *2* con la pieza *ens0* o con *ens1*, que se encuentran respectivamente en *cont\_ens0* y *cont\_ens1*. Esta acción está gobernada por una máquina ficticia denominada *ficticia\_ensamble*, que es una máquina de tipo ensamblaje que necesita de *op01* y que utiliza la regla de entrada de piezas que se muestra en la Figura 3.7. De esta forma, cuando haya una pieza en *entrada\_ensa* y sea de *tipo0* y el atributo *tipo\_acople* sea 0, la unión será la correcta (*tipo0* con *ens0*), es decir, se cogerá una pieza de *cont\_ens0*; en caso contrario, si el *tipo\_acople* toma valor 1, la unión no será la correcta y se cogerá una pieza del contenedor *cont\_ens1*.

Cuando el ensamblaje de piezas no se realice correctamente (*tipo\_acople* no coincida con *tipo*), en la salida de esta máquina de ensamblaje se modificará el icono que posee el conjunto. Cabe destacar que el defecto de calidad predomina sobre el defecto de ensamblaje; por tanto, si una pieza es defectuosa de calidad, el icono que representa a la pieza después del ensamblaje será el de defecto de calidad, aunque el ensamblaje no fuese el correcto. En los casos en los que la pieza posee la calidad adecuada, y el ensamblaje no se ha realiza correctamente, el icono que poseerá será el icono 82, como se muestra en Figura 3.8. La representación, tanto en 2D como en 3D, se puede apreciar en la Figura 3.9. En el caso de que la calidad de la pieza sea la correcta y que la unión se haya realizado correctamente, la descripción de la pieza tanto en 2D como en 3D será idéntica a la que tenía antes de la unión, es decir, la representación de *tipo0* o de *tipo1*, dependiendo del tipo de pieza que sea.

El conjunto pasa por la *maq3* y *maq4*, en las que se le va aportando una determinada calidad como ocurriese en las *maq1*, *maq2* y *ficticia\_ensamblaje*. Como en el caso del ensamblaje de las dos piezas en la máquina *ficticia\_ensamble*, el defecto de calidad predomina sobre el fallo en la unión; si una pieza no está ensamblada correctamente y en las maquinas posteriores no se aporta la calidad mínima, la representación de la pieza sería la indicada en la Figura 3.6.

```

Edit INPUT RULE FOR MACHINE ficticia_ensamble
Select Search Editor Print
IF NOCC (entr_ensa,0,1) = 1
IF entr_ensa:tipo = 0 AND entr_ensa:tipo_acople = 0
MATCH/ANY
tipo0 out of entr_ensa at Front #(1) AND ens0 out of cont_ens0 #(1)
ELSE
IF entr_ensa:tipo = 0 AND entr_ensa:tipo_acople = 1
MATCH/ANY
tipo0 out of entr_ensa at Front #(1) AND ens1 out of cont_ens1 #(1)
ELSE
IF entr_ensa:tipo = 1 AND entr_ensa:tipo_acople = 1
MATCH/ANY
tipo1 out of entr_ensa at Front #(1) AND ens1 out of cont_ens1 #(1)
ELSE
IF entr_ensa:tipo = 1 AND entr_ensa:tipo_acople = 0
MATCH/ANY
tipo1 out of entr_ensa at Front #(1) AND ens0 out of cont_ens0 #(1)
ENDIF
ENDIF
ENDIF
ENDIF
ELSE
Wait
ENDIF
    
```

Figura 3.7: regla de entrada en la máquina *ficticia\_ensamble* en los modelos técnica *poka yoke*

```

calidad = calidad
IF calidad < 4.5
ICON = 44
ELSEIF tipo <> tipo_acople
ICON = 82
ENDIF
    
```

Figura 3.8: regla de salida de la máquina *ficticia\_ensamble* en los modelos técnica *poka yoke*

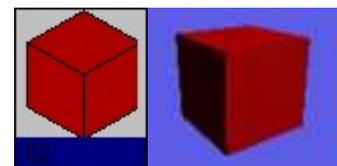


Figura 3.9: representación del ensamblaje incorrecto en los modelos técnica *poka yoke*

Como se ha comentado anteriormente, se posee una zona común a los dos tipos de piezas (zona comentada hasta el momento) y una línea de proceso independiente para cada uno de los tipos de piezas. La disposición de los diferentes tipos en una zona u otra la realiza la *maquina5*, que se encarga de colocar las piezas *tipo0* en *división\_tipo0* y *tipo1* en *división\_tipo1*. Hay que tener en cuenta que esta máquina posee un pequeño error a la hora de identificar las piezas, pudiendo definir un tipo de una pieza cuando en realidad es el contrario y, por consiguiente, colocar la pieza en la línea de proceso incorrecta. Este error se produce aleatoriamente, es decir, cuando un número aleatorio entre 0 y 1 es mayor que 0.9, la pieza se colocará en el conveyor que no le corresponde, es decir, si la pieza es *tipo0* la colocará en *división\_tipo1* y cuando es *tipo1* la colocará en *división\_tipo0*, como se puede apreciar en Figura 3.10.

```

Edit OUTPUT RULE FOR MACHINE maquina5
Select Search Editor Print
IF tipo = 0
  IF Random () < 0.9
    PUSH to division_tipo0 at Rear
  ELSE
    PUSH to division_tipo1 at Rear
  ENDIF
ELSEIF tipo = 1
  IF Random () < 0.9
    PUSH to division_tipo1 at Rear
  ELSE
    PUSH to division_tipo0 at Rear
  ENDIF
ELSE
  Wait
ENDIF

```

Figura 3.10: disposición de las piezas en los procesos independientes modelo sin poka yoke

A continuación, se va a describir la línea independiente del *tipo0*, siendo la del *tipo1* igual que ésta, pero refiriéndose al *tipo1*. Esta línea se puede apreciar en la Figura 3.11, correspondiente con la zona superior de la Figura 3.1. Los tiempos de ciclo de las máquinas asociadas a estas líneas, así como la calidad mínima, se ha definido en la Tabla 3.2. Como ocurriese en la zona común, las máquinas aportan cierta calidad a las piezas, pudiendo no llegar al mínimo deseado.

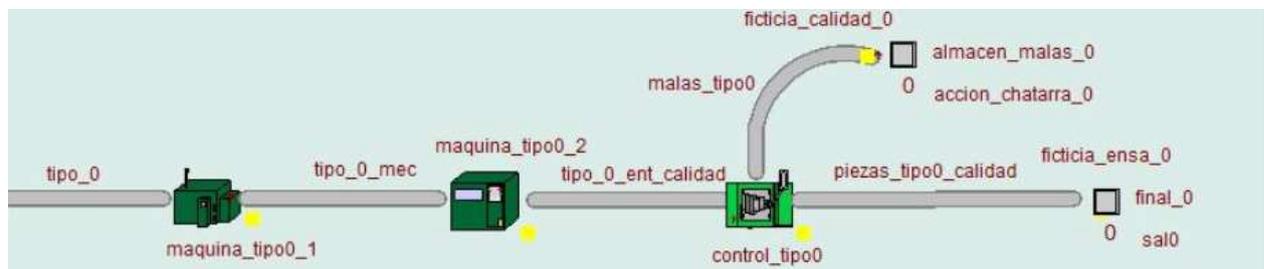


Figura 3.11: representación de la zona específica en el modelo sin poka yoke

Al final de la línea será necesario realizar un control de calidad, para comprobar las piezas defectuosas, ya sea porque el ensamblaje no se ha realizado correctamente, que la calidad mínima a la salida de cada una de las máquinas no se haya conseguido o que el tipo de pieza no sea el correcto. Si alguna de estas condiciones no se cumpliera, las piezas se depositarían en la cinta *malas\_tipo0*. El control de todas estas características la realizará la máquina *control\_tipo0*, la forma que tiene esta máquina de identificar las piezas sin la mínima calidad o con ensamblaje erróneo es mediante el icono que posee, dado que en todas las máquinas se ha ido cambiando el icono de las piezas cuando no se cumple algunas de estas características, ya sea con icono 82 o 44. Además, la identificación de *tipo1* se realizará mediante el atributo *tipo*, como se puede apreciar en la Figura 3.12. Las piezas que poseen la calidad deseada, están ensambladas correctamente y son de *tipo0* se depositarán en *piezas\_tipo0\_calidad*, que indican el final del proceso.

```

Edit OUTPUT RULE FOR MACHINE control_tipo0
Select Search Editor Print
IF ICON <> 44 AND ICON <> 82 AND tipo = 0
  PUSH to piezas_tipo0_calidad at Rear
ELSE
  PUSH to malas_tipo0 at Rear
ENDIF

```

Figura 3.12: salida de control\_tipo0 en los técnica modelos poka yoke

Para conocer las piezas defectuosas que se generan en el proceso, en la cinta `malas_tipo0` se posee una variable contador, denominada `chatarra0`, que incrementará su valor cada vez que entre una pieza defectuosa, en la zona del tipo1 esta variable se denomina `chatarra1`, para conocer la que se produce en esta zona. Para evitar que se acumulen piezas en esta cinta, se dispone de una máquina ficticia denominada `ficticia_calidad0` que se encarga de disponer las piezas en un contenedor denominado `almacen_malas_0` cuando en la cinta `malas_tipo0` hay al menos 7 piezas, como se puede apreciar Figura 3.13. Para mejorar la apariencia visual del modelo en 3D, se ha decidido que en determinados instantes de la simulación se saquen piezas del contenedor `almacen_malas_0`, Figura 3.14; éstas se sacarán del contenedor mediante la máquina `ficticia_acción_chatarra_0`, que se encargará de depositarlas en `scrap`.

```

Edit INPUT RULE FOR MACHINE ficticia_calidad_0
Select Search Editor Print
IF NParts (malas_tipo0) >= 7
  PULL from malas_tipo0 at Front
ELSE
  Wait
ENDIF
    
```

Figura 3.13: caso de sacar piezas de malas\_tipo0 modelos técnica poka yoke

```

Edit INPUT RULE FOR MACHINE accion_chatarra_0
Select Search Editor Print
IF TIME = 2000 OR TIME = 4000 OR TIME= 6000
  PULL from malas_tipo0 at Front
ELSE
  Wait
ENDIF
    
```

Figura 3.14: necesidad de sacar piezas de almacen\_malas\_0 modelos técnica poka yoke

En el caso de que las piezas posean la calidad deseada, que sean del tipo correcto y el ensamblaje se haya realizado correctamente, se depositarán en `piezas_tipo0_calidad`. En esta cinta se ha implementado la variable `salida_0`, que nos indicarán la cantidad de piezas que se generan correctamente en la zona del `tipo0`. Para evitar el almacenamiento de las piezas en la cinta, se ha dispuesto de una máquina ficticia (`ficticia_ensa_0`) que es una máquina de ensamblaje que recoge 5 piezas de la cinta `piezas_tipo0_calidad` cuando sea necesario, según la Figura 3.15. Posteriormente, la pieza que se genera se depositará en `final_0`. Como ocurriese en el caso del `almacen_malas_0`, se sacarán las piezas que hay en el contenedor `final_0` en un determinado instante para evitar la acumulación en la visualización del modelo 3D; esto se realizará con la máquina ficticia `sal0`, cuya regla de entrada se puede apreciar en la Figura 3.16

```

Edit INPUT RULE FOR MACHINE ficticia_ensa_0
Select Search Editor Print
IF NParts (piezas_tipo0_calidad) >= 10
  PULL from piezas_tipo0_calidad at Front
ELSE
  Wait
ENDIF
    
```

Figura 3.15: necesidad de ensamblar piezas a la salida de la piezas\_tipo0\_calidad en los modelos técnica poka yoke

```

Edit INPUT RULE FOR MACHINE sal0
Select Search Editor Print
IF TIME = 4000 OR TIME = 8000
  PULL from final_0
ELSE
  Wait
ENDIF
    
```

Figura 3.16: regla de entrada de sal0, salida de piezas de la zona0 en los modelos técnica poka yoke

### 1.2. IMPLEMENTACIÓN DE POKA YOKE DE ADVERTENCIA

Para evitar la cantidad de chatarra que se genera en el proceso al no realizar controles de calidad hasta el final del proceso, se ha implementado una serie de poka yokes en distintas máquinas para así disminuir los defectos que se producen y, con ello la cantidad, de chatarra. En la Figura 3.17 se puede apreciar el modelo con unos poka yokes, que vendrán dados por: el control de calidad a la salida de la `máquina4` (garantizando que la zona común posee la calidad deseada), un mejor reconocimiento del tipo de piezas en la `máquina5` y un control de tipo y de ensamblaje correcto antes de entrar en cada una de las zonas específicas a cada tipo

(antes de cada una de las *maquina\_tipo1*). En las máquinas donde se han implementado poka yokes se producen una salida de piezas del proceso, haciendo que las piezas que no poseen la característica que se controla con ese poka yoke salgan del proceso. Por esa razón, en la *maquina4* hay una salida hacia desecho (*scrap*). En el caso de la entrada en las zonas específicas de cada tipo, se puede apreciar una salida de piezas del proceso; en este caso, las piezas no van directamente a *scrap*, dado algunas se pueden recuperar. Las piezas que no tengan el tipo correspondiente, se pueden recuperar dado que no se las ha realizado ninguna operación; por el contrario, las que no poseen un ensamblaje correcto se depositaran en *scrap*, dado que no se pueden recuperar.

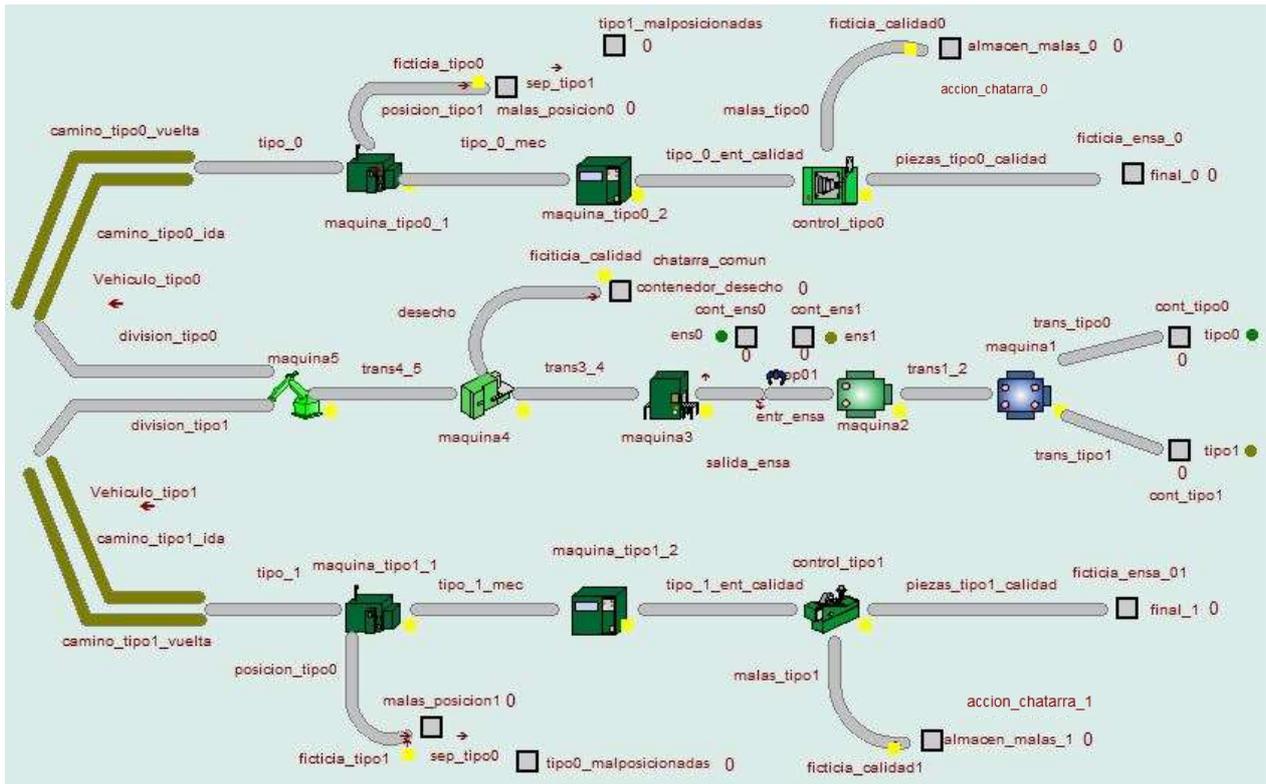


Figura 3.17: representación del modelo poka yoke de advertencia

Cabe destacar que la calidad mínima que hay que asegurar en cada una de las máquinas es la misma que en el modelo sin poka yoke, pero el tiempo de ciclo de algunas de ellas ha aumentado, pudiéndose apreciar los nuevos tiempos de ciclos en la Tabla 3.3. En cuanto a la calidad que se aporta en cada una de ellas, es la misma que en el modelo sin poka yoke

Máquina	Tiempo de ciclo (s)	Calidad mínima admisible	Calidad adquirida
Ficticia_ensamble	4.25		
Maquina4	8	16.5	Tnormal(5,2.5,3.5,7.5)
Maquina5	9		
Maquina_tipo0_1	11	22	Tnormal(4.5,0.8,3,6)

Tabla 3.3: tiempo de ciclo de las máquinas en el modelo poka yoke de advertencia

La primera mejora se ha realizado al implementar un poka yoke en la máquina ficticia\_ensamble, para mejorar el porcentaje de piezas que poseen un ensamblaje correcto. Para implementar este poka yoke en el modelo, se ha jugado con el valor que puede tomar tipo\_acople con referencia al valor que tomaba para el caso sin poka yoke ( Figura 3.2). Es decir, cuanto mayor sea el límite del número aleatorio, mayor probabilidad de que tipo\_acople coincida con tipo, como se puede apreciar en Figura 3.18. De esta forma, la probabilidad de que se acoplen de forma incorrecta será menor. En la representación en 3D, Figura 3.19 y Figura 3.20, se pueden apreciar que los contenedores cont\_ens0 y cont\_ens1 en el modelo 3D poseen los mismos colores que poseen las piezas a las que se deben ensamblar las piezas contenidas.

```

Edit Actions On Create For Part tipo0
Select Search Editor Print
calidad = 0
tipo = 0
IF Random () <= 0.95
tipo_acople = tipo
ELSE
tipo_acople = 1
ENDIF
    
```

Figura 3.18: definición del atributo tipo\_acople en modelo con poka yoke de advertencia

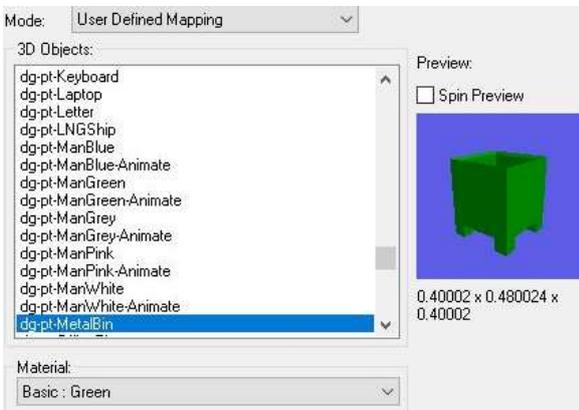


Figura 3.19: representación en 3D de cont\_ens0 modelo poka yoke advertencia

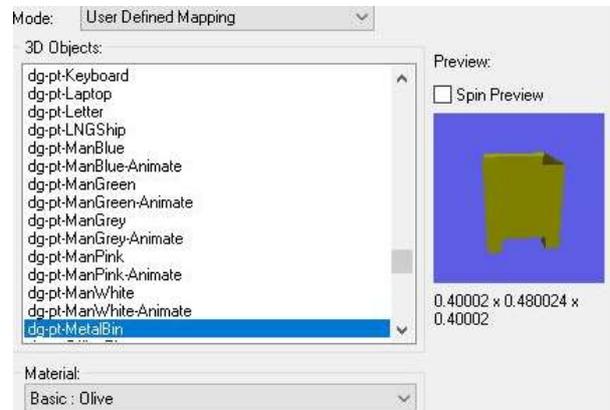


Figura 3.20: representación en 3D de cont\_ens1 modelo poka yoke advertencia

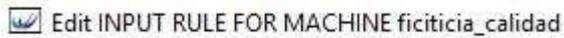
Al finalizar la zona común, es decir, a la salida de la maquina4 se dispone de un poka yoke que controla la calidad que poseen las piezas hasta el momento, desechando las piezas que en alguna de las operaciones no ha alcanzado la calidad mínima. La forma de programar este poka yoke en el modelo ha sido con el icono que se ha asignado a las piezas que no posee en la calidad mínima, es decir, con el icono 44, como se puede apreciar Figura 3.21. Las piezas que no poseen la calidad mínima irán a la cinta de desecho; por el contrario, las que poseen la calidad adecuada se depositarán en la cinta que las lleva a la siguiente máquina del proceso. Cabe destacar que a la entrada de la cinta desecho se ha implementado una variable denominada chatarra\_com para conocer el número de piezas que no han adquirido la calidad deseada en la zona común.

```

Edit OUTPUT RULE FOR MACHINE maquina4
Select Search Editor Print
IF ICON <> 44
PUSH to trans4_5 at Rear
ELSE
PUSH to desecho at Rear
ENDIF
    
```

Figura 3.21: poka yoke en la salida de maquina4 modelo poka yoke advertencia

Para evitar que se acumulen las piezas que se disponen en la cinta desecho, se dispondrá de una máquina ficticia denominada ficticia\_calidad, que coge 5 piezas para situarlas en el contenedor denominado contenedor\_desecho cuando en la cinta hay más de 7 piezas, como indica la Figura 3.22. De forma similar, se procederá con dicho contenedor, haciendo que una máquina ficticia saque piezas de éste en unos determinados instantes, como se hiciese en el modelo sin poka yoke, como se aprecia en la Figura 3.23



Select Search Editor Print

```
IF NParts (desecho) >= 7
PULL from desecho at Front
ELSE
Wait
ENDIF
```

Figura 3.22: salida de piezas de cinta desecho modelo poka yoke advertencia



Select Search Editor Print

```
IF TIME = 2000 OR TIME = 4000 OR TIME = 6000 OR TIME = 8000
PULL from contenedor_desecho
ELSE
Wait
ENDIF
```

Figura 3.23: salida de piezas del contenedor chatarra\_comun modelo poka yoke advertencia

En la **maquina5** se dispone de un poka yoke que mejora la determinación del tipo de la pieza que posteriormente se posicionará en una cinta u otra, haciendo que el número de piezas colocadas de forma errónea disminuya. Por tanto, en la salida de la **maquina5**, como ocurriese en la máquina de ensamblaje, se ha aumentado el límite del valor que puede tomar el número aleatorio. De esta forma, la probabilidad de que la pieza se disponga en la cinta correspondiente es mayor, como se puede apreciar en la Figura 3.24 (en el modelo sin poka yoke el límite se encontraba en 0.9).



Select Search Editor Print

```
IF tipo = 0
IF Random () < 0.98
PUSH to division_tipo0 at Rear
ELSE
PUSH to division_tipo1 at Rear
ENDIF
ELSEIF tipo = 1
IF Random () < 0.98
PUSH to division_tipo1 at Rear
ELSE
PUSH to division_tipo0 at Rear
ENDIF
ELSE
Wait
ENDIF
```

Figura 3.24: disposición de las piezas en la maquina5 en modelo poka yoke advertencia

Una vez que se disponen en cada una de las cintas, antes de entrar en la primera de las zonas correspondientes a cada tipo (**maquina\_tipo0\_1** y **maquina\_tipo1\_1**), se comprobará el tipo que poseen las piezas y que el ensamblaje se haya efectuado correctamente. Esta comprobación se llevará a cabo al final de la cinta donde se depositan las piezas procedentes del vehículo, como se aprecia en la Figura 3.25. Por tanto, cuando el **tipo** y el ensamblaje sean los correctos, se llevarán a la máquina siguiente, para continuar el proceso; por el contrario, cuando el tipo no sea el correcto o el ensamblaje no se haya realizado correctamente se depositarán en la cinta **posición\_tipo**.



Select Search Editor Print

```
IF tipo <> tipo_acople OR tipo = 0
PUSH to posicion_tipo0 at Rear
ELSE
PUSH to maquina_tipo1_1
ENDIF
```

Figura 3.25: poka yoke entrada zona especifica tipo1 en modelo poka yoke advertencia

Como se puede apreciar en la Figura 3.17, de la cinta **posición\_tipo1** se disponen las piezas en el contenedor **malas\_posicion0**, y mediante la máquina ficticia **ficticia\_tipo0** se sacarán las piezas de esta cinta para evitar las acumulaciones que se puedan producir. Una vez que se disponen en este contenedor, mediante otra máquina ficticia, **sep\_tipo01**, se sacan del contenedor de la misma manera que se ha procedido para el defecto de calidad de la zona de común. Se pueden disponer las piezas en dos sitios, Figura 3.26: si no se pueden recuperar, dado que el ensamblaje ha sido incorrecto, hacia **scrap**, o si el ensamblaje es el correcto pero el tipo no corresponde con el de la cinta, pudiéndose recuperar, situándolas en **tipo1\_malposicionadas** debido a que no se ha realizado ninguna operación en la zona no correspondiente con su tipo. Como se produce una salida del proceso de las piezas que no han

sidio ensambladas correctamente, se dispondrá una variable denominada **chatarra\_tipo\_0**, en las que el valor se va incrementando, tal y como se aprecia en la Figura 3.27.

```

Edit OUTPUT RULE FOR MACHINE sep_tipo1
Select Search Editor Print
IF ICON = 82
  PUSH to SCRAP
ELSE
  PUSH to tipo1_malposicionadas
ENDIF
    
```

Figura 3.26: separación de piezas incorrectas tipo modelo poka yoke advertencia

```

Edit Actions On Join For Conveyor posicion_tipo1
Select Search Editor Print
IF ICON = 82
  chatarra_tipo_0 = chatarra_tipo_0 + 1
ENDIF
    
```

Figura 3.27: variable contador de chatarra\_tipo\_0 modelo poka yoke advertencia

El resto del proceso se realiza de igual forma que el modelo sin poka yoke, salvo que en la máquina de **control\_calidad** no se realiza el control de tipo o ensamblaje incorrecto; dado que ya se ha detectado anteriormente, sólo se realizará el control de calidad de la zona específica de cada uno de los tipos.

### 1.3. IMPLEMENTACIÓN DE POKA YOKE DE FIABILIDAD

En el modelo con poka yoke de advertencia se sigue produciendo piezas con defecto, ya sea por defecto de calidad, por ensamblaje incorrecto o por disposición de las piezas en su zona incorrecta. Es, por esta razón, por lo que se ha implementado en cada una de las máquinas un poka yoke de control para garantizar la calidad en cada una de las máquinas. De esta forma, el 100 % de las piezas poseen la calidad deseada, llegando al final del proceso todas las piezas que entran en él. Como se puede apreciar en la Figura 3.28.

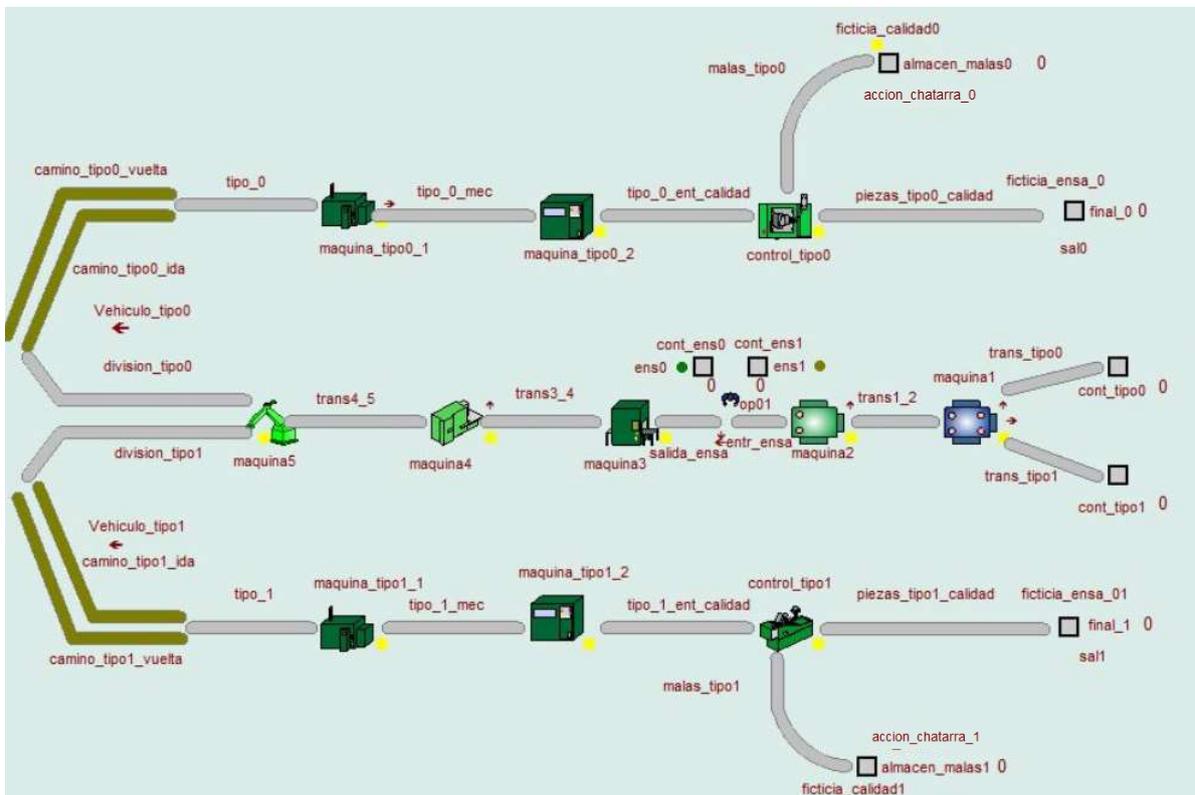


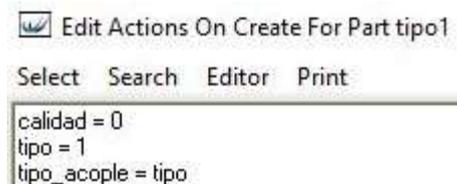
Figura 3.28: representación del modelo poka yoke de fiabilidad

Para comenzar, como en el caso de poka yoke de advertencia, se han modificado los tiempos de ciclo así como las calidades adquiridas por la implantación de los poka yoke, como se pueden apreciar en la Tabla 3.4

Máquina	Tiempo de ciclo (s)	Calidad mínima admisible	Calidad adquirida
Maquina1	12.5	3	10
Maquina2	13.5	7.7	10
Maquina3	10.5	13	10
Ficticia_ensamble	4.5		
Maquina4	8.5	16.5	10
Maquina5	9.5		
Maquina_tipo0_1	11.5	22	10
Maquina_tipo0_2	15.5	26.5	10
Maquina_tipo0_3	10		

Tabla 3.4: tiempos de ciclo modelo poka yoke de fiabilidad

Se comenzará a describir el proceso por la definición del atributo de las piezas, que indicarán cómo se realiza el acople, `tipo_acople`. Como se ha comentado anteriormente, el ensamblaje se va a realizar correctamente en todas las piezas; por esa razón, `tipo_acople` tiene que coincidir con `tipo`, como se puede apreciar en la Figura 3.29, donde se define la pieza que corresponde con `tipo1`. Haciendo que el operario `op01` realice correctamente la operación, siendo las reglas de la máquina ficticia las mismas que en el modelo sin poka yoke, Figura 3.7.



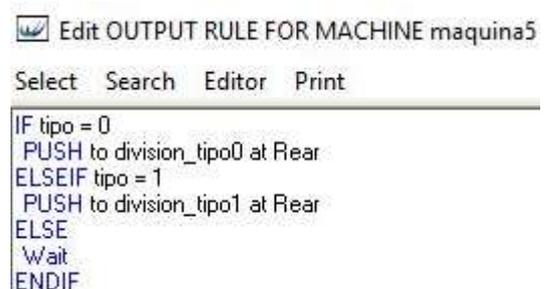
```

Edit Actions On Create For Part tipo1
Select Search Editor Print
calidad = 0
tipo = 1
tipo_acople = tipo

```

Figura 3.29: definición de atributo `tipo_acople` en modelo poka yoke de fiabilidad

En las máquinas en las que se van adquiriendo calidad (`maquina1`, `maquina2`, `maquina3`, `maquina4` y las correspondientes a la zona específica a cada tipo) el comportamiento es igual que en el modelo sin poka yoke, pero en este caso adquiriendo la calidad en cada una de las máquinas según la Tabla 3.4. Cabe destacar que en la `maquina5`, donde se produce la clasificación de las piezas según su `tipo`, también el reconocimiento de las piezas siempre es el correcto; es decir, las de `tipo0` las va a colocar siempre en la zona superior y las de `tipo1` en la zona inferior, como se aprecia en la Figura 3.30



```

Edit OUTPUT RULE FOR MACHINE maquina5
Select Search Editor Print
IF tipo = 0
PUSH to division_tipo0 at Rear
ELSEIF tipo = 1
PUSH to division_tipo1 at Rear
ELSE
Wait
ENDIF

```

Figura 3.30: clasificación de las piezas en modelo con poka yoke de fiabilidad

Al finalizar el proceso se encuentra la máquina que realiza el control de calidad; debido a la presencia de poka yokes de control, la calidad, el tipo y el ensamblaje siempre serán los correctos. Como en los otros casos, para evitar la acumulación de piezas en la cinta `piezas_tipo0_calidad` existirá una máquina ficticia que las depositará en el contenedor `final_0`, `ficticia_ensa0`, además de la salida de piezas del proceso en un instante determinado, `sal0`.

## 2. TÉCNICA CÉLULA EN U (OPERARIO ACOMPAÑANDO A PIEZA)

### 2.1. DESCRIPCIÓN DEL MODELO CON DISPOSICIÓN LINEAL

El siguiente modelo representa un proceso productivo en el que un operario tiene que mover la pieza, es decir, es el propio operario el que lleva la pieza de una máquina a otra. Además de llevar la pieza, se ha supuesto que el operario debe realizar una operación previa a la máquina antes de que comience la actividad, teniendo que estar el operario en la máquina mientras que ésta realice la actividad. El proceso está compuesto por 10 máquinas y, como en el caso de teoría de las limitaciones, una entrada y una salida del proceso. En dicha técnica, los paths entre las máquinas se utilizaban para realizar las tareas de mantenimiento; en este caso, los paths se usan para el traslado de la pieza con cada uno de los operarios (3 operarios, uno para cada zona). El modelo es el indicado en la Figura 3.31.

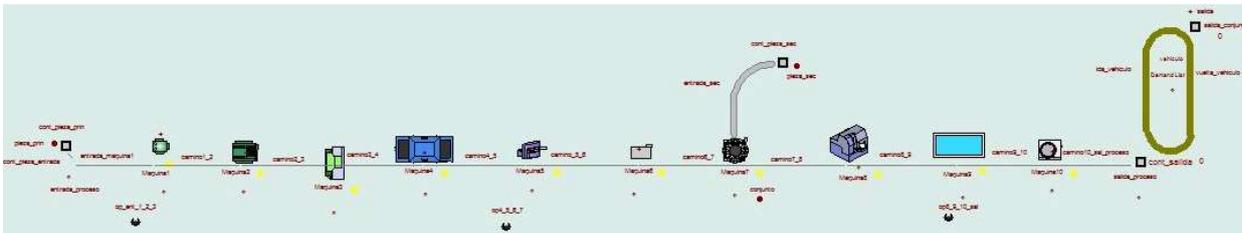


Figura 3.31: representación del modelo acompañar pieza lineal

Como se puede apreciar, se trata de un proceso muy extenso en tamaño, con mucho trayecto a realizar por los operarios. Cada operario se encargará de cuatro máquinas, contando como máquinas la entrada y salida de piezas del proceso; por tanto, se va a dividir el proceso en tres zonas, dadas por cada uno de los operarios: la zona1 vendrá dado por la entrada\_proceso, y las maquinas1-2-3, como se indicada en la Figura 3.32; la zona2 por las maquinas4-5-6-7, visualizándose en la Figura 3.33; y la zona3 vendrá dada por las maquinas8-9-10 y la salida\_proceso, como se puede apreciar en la Figura 3.34 . Además, se ha indicado cómo se realiza la salida de piezas hacia otro proceso, Figura 3.35, donde vehículo las lleva hasta un contenedor.



Figura 3.32: máquinas pertenecientes a la zona1 modelo acompañar pieza lineal

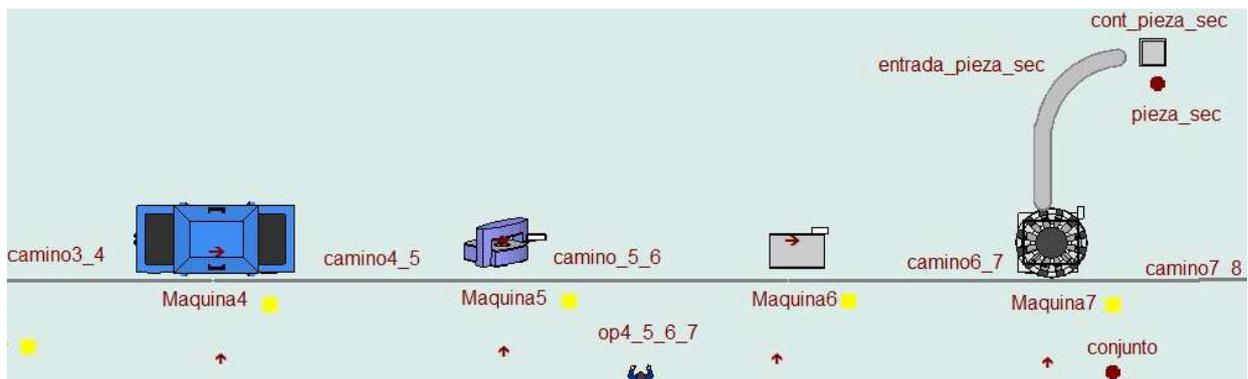


Figura 3.33: máquinas pertenecientes a la zona2 modelo acompañar pieza lineal



Figura 3.34: máquinas pertenecientes a la zona3 modelo acompañar pieza lineal

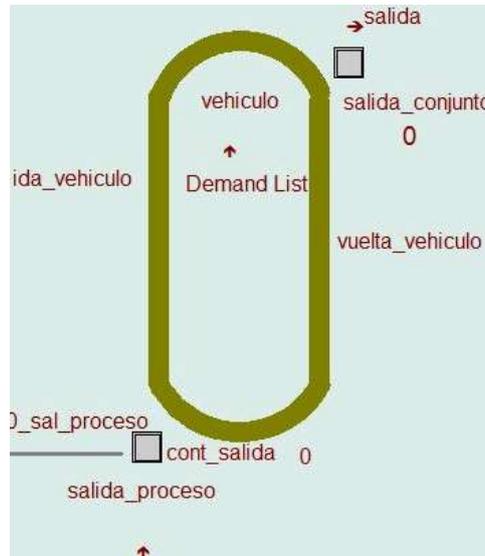


Figura 3.35: salida del proceso modelo acompañar pieza lineal

Una vez que se conocen las máquinas que forman parte del proceso y cómo se disponen, se procederá a definir las piezas que intervienen en el proceso. Existen tres piezas: **pieza\_prin**, que es la pieza principal del proceso y a la que se realizarán una serie de operaciones hasta llegar a la **maquina7**; **pieza\_sec**, que es la pieza que se ensamblará con **pieza\_prin**; y **conjunto**, que es la unión resultante del ensamblaje (**máquina7**). La visualización de estas piezas se puede apreciar en las Figura 3.36, Figura 3.37 y Figura 3.38.



Figura 3.36: representación de la pieza\_prin modelos acompañar pieza

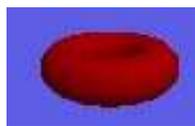


Figura 3.37: representación de la pieza\_sec modelos acompañar pieza



Figura 3.38: representación del conjunto modelos acompañar pieza

Cabe destacar que la entrada de piezas, tanto la secundaria como la principal, se realizará con una frecuencia elevada, para que no se saturen **cont\_pieza\_prin**, y **cont\_pieza\_sec**. Estos tiempos elevados se deben a que la entrada de **piezas\_prin** la realiza el **op\_1\_2\_3**. Éste tiene que moverse entre las máquinas consecutivas, además del tiempo que tarda cada máquina en realizar su operación (el operario tiene que estar mientras realiza la operación la máquina) así como el tiempo que tarda en volver desde la **maquina3** hasta el **cont\_pieza\_prin**. Este elevado tiempo hace que la **pieza\_sec** también tendrá que llegar con una frecuencia mayor, para estar sincronizada con la **pieza\_prin**, mostrándose las frecuencias en las Figura 3.39 y Figura 3.40.

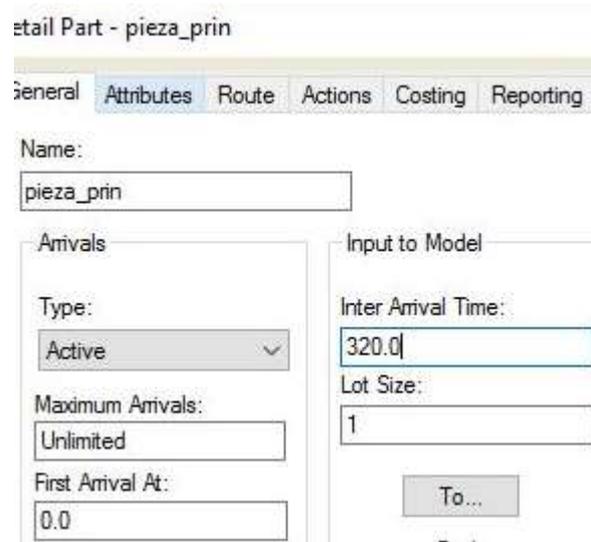


Figura 3.39: Llegada de pieza\_prin modelos acompañar pieza

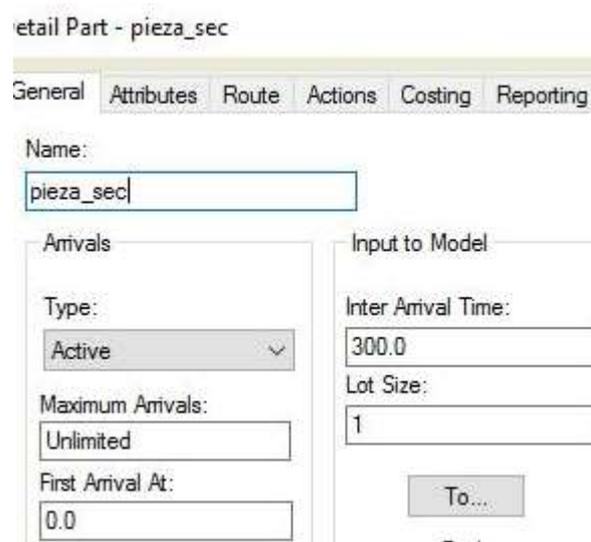


Figura 3.40: Llegadas de pieza\_sec modelos acompañar pieza

Una vez descritas las piezas que forman parte del proceso, se procederá a describir cómo se lleva a cabo el proceso productivo. Se han definido tres variables, **operación\_op1**, **operación\_op2** y **operación\_op3**, asociadas a cada uno de los operarios; es decir, **operación\_op1** al operario **op\_ent\_1\_2\_3**, **operación\_op2** asociada al operario **op4\_5\_6\_7** y, por último, **operación\_op3** asociada al operario **op8\_9\_10\_sal**. Estas variables determinarán en qué máquina están los operarios; es decir, las variables van a poder tomar cuatro valores, que irán cambiando cíclicamente cuando se acabe la actividad de una máquina y hagan que el operario se tenga que mover su máquina consecutiva. En la Tabla 3.5 se puede apreciar la localización de cada uno de los operarios en función del valor de la variable. Cabe destacar que éstas variables toman valor 0 en el momento inicial.

Valor	Máquina correspondiente		
	Operación_op1	Operación_op2	Operación_op3
0	Entrada_proceso	4	8
1	1	5	9
2	2	6	10
3	3	7	Salida_proceso

Tabla 3.5: valores que toman las variables que condicionan los movimientos de los operarios modelo acompañar pieza lineal

Se procederá a explicar cómo es el movimiento del **op\_ent\_1\_2\_3** en la zona1, comportándose igual los otros operarios en las otras zonas con sus máquinas correspondientes.

En la zona1 está la entrada al proceso de **pieza\_prin**, realizando esta operación la máquina ficticia **entrada\_proceso**, que coge **pieza\_prin** del **cont\_pieza\_prin**; en este elemento se ha implementado una variable denominada **piezas\_entrada** para conocer cuántas piezas entran en el proceso. Como se ha comentado anteriormente, para que el operario realice la actividad el valor de la **operación\_op1** será 0, como se puede apreciar en la Figura 3.41. Además, como se ha explicado, los operarios se tienen que trasladar por paths; en este caso, el operario usará el path **cont\_pieza\_entrada** (Figura 3.42). En la descripción del modelo se ha definido que el operario (Figura 3.44) debe realizar una acción previa para simular la acción de la máquina; en este caso, la de coger la pieza, como se aprecia en la Figura 3.43. Al finalizar la actividad,

habrá que indicar que la variable **operacion\_op1** toma el siguiente valor, es decir, 1, como indica la Figura 3.45, para que el operario pueda llevar la pieza hacia **maquina1**.

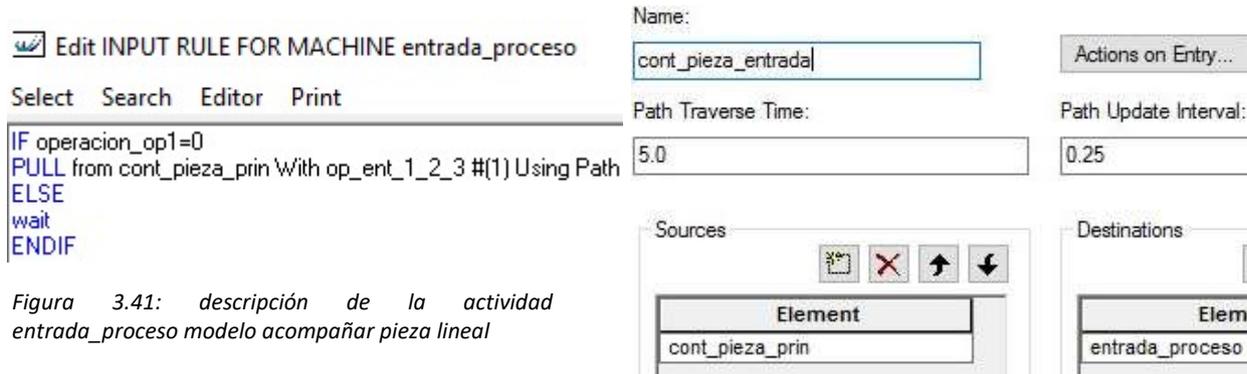


Figura 3.41: descripción de la actividad entrada\_proceso modelo acompañar pieza lineal

Figura 3.42: descripción del path cont\_pieza\_entrada en modelo acompañar pieza lineal

ail Machine - entrada\_proceso

General Setup Breakdowns Fluid Rules Shift Actions Costing Reporting Notes

	Description	Setup Mode				Setup Duration					
		Mode	No. of Operations	Ops to First Setup	Expression	Setup with no part	Station Number	Actions on Start	Labor Rule	Setup Time	Action Fini
1	New Setup	No. of Op	1	0		<input type="checkbox"/>		N	Y	50.0	N

Figura 3.43: definición del tiempo de preparación de la actividad entrada\_proceso en modelo acompañar pieza lineal

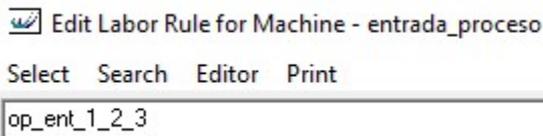


Figura 3.44: definición de operario en entrada\_proceso modelo acompañar pieza lineal

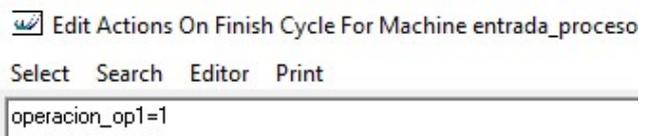


Figura 3.45: cambio de valor operacion\_op1 modelo acompañar pieza lineal

En este punto, nos encontramos que el valor de la **operación\_op1** es 1, la regla de entrada de la **maquina1** es la mostrada en la Figura 3.46. Es decir, cuando **operación\_op1** valga 1, el **op\_ent\_1\_2\_3** deberá ir hacia la **maquina1** por el path correspondiente, es decir, **entrada\_maquina1** para coger la pieza. Como en el caso anterior, cuando la pieza llegue a la **maquina1**, el **op\_ent\_1\_2\_3** deberá realizar una acción previa, dada por el **setup**. Cuando acabe la actividad, el valor de **operación\_op1** tomará valor 2, haciendo que se repita lo mismo en la **maquina2**, usando el path **camino\_1\_2** y haciendo que, al finalizar la actividad, el valor de **operación\_op1** tome valor 3. En la última máquina que forma parte de esta zona, la **maquina3**, el operario cogerá la **pieza\_prin** de la **maquina2** cuando el valor **op\_operacion1** toma valor 3, es decir, a la salida de la **maquina2**. Cabe destacar que cuando acaba la actividad de la **maquina3**, el valor de **operación\_op1** tomará valor 0, para continuar con el ciclo de la zona, como muestra la Figura 3.47. Por tanto, a la salida de la **maquina3**, como el valor de **operación\_op1** es 0, el **op\_ent\_1\_2\_3** tendrá que ir a coger piezas del **con\_pieza\_prin**, el trayecto lo realizará por los paths indicados, dado que los paths forman un único trayecto, el final de un path es el comienzo de otro.

```

Edit INPUT RULE FOR MACHINE Maquina1
Select Search Editor Print
IF operacion_op1 = 1
PULL from pieza_prin out of entrada_proceso With op_ent_1_2_3#(1) Using Path
ELSE
Wait
ENDIF
    
```

Figura 3.46: entrada de piezas en la maquina1 en el modelo acompañar pieza lineal

```

Edit Actions On Finish Cycle For Machine Maquina3
Select Search Editor Print
operacion_op1=0
    
```

Figura 3.47: definición del nuevo ciclo a realiza por el op\_ent\_1\_2\_3 en modelo acompañar pieza lineal

Una vez que se ha descrito la zona1, y como se ha comentado anteriormente, el procedimiento de acompañar a las piezas en la zona2 es el mismo que en la zona1, pero haciendo una serie de matizaciones. En primer lugar, en la maquina4 (máquina en la que comienza la zona2) se ha implementado una condición que no posee las máquinas de la zona1. Esta condición se debe al tiempo que tardaría en llegar la primera pieza a la maquina4 procedente de la zona1; por este motivo, se ha supuesto que en el instante 60 llega una pieza\_prin a la maquina4 procedente de world (Figura 3.48), para que el op\_4\_5\_6\_7 no esté sin realizar ninguna actividad. En el resto de los casos, el operario cogerá piezas de la maquina3 cuando la variable operación\_op2 tome valor 0.

```

Edit INPUT RULE FOR MACHINE Maquina4
Select Search Editor Print
IF TIME = 60
PULL from pieza_prin out of WORLD
ELSEIF operacion_op2 = 0
PULL from Maquina3 With op4_5_6_7#(1) Using Path
ELSE
Wait
ENDIF
    
```

Figura 3.48: descripción de la entrada a la maquina4 en modelo acompañar pieza lineal

Otra consideración a tener en cuenta en esta zona es relativa a la maquina7, en donde se produce el ensamblaje de la pieza\_prin con la pieza\_sec para obtener el conjunto. La regla de entrada de estas dos piezas a la maquina7 se muestra en la Figura 3.49. Se puede apreciar que la máquina coge pieza\_sec de entrada\_pieza\_sec y la pieza\_prin la cogerá de la maquina6 y el op\_4\_5\_6\_7 irá por el camino6\_7. Al finalizar la operación se realizarán dos acciones: se hará que la variable operación\_op2 tome valor 0 para volver a realizar el ciclo de la zona2 y que el ensamblaje de las dos piezas se convierta en la pieza conjunto.

```

Edit INPUT RULE FOR MACHINE Maquina7
Select Search Editor Print
IF operacion_op2 = 3
MATCH/ANY
Maquina6 With op4_5_6_7 Using Path #(1)entrada_sec at Front #(1)
ELSE
Wait
ENDIF
    
```

Figura 3.49: definición de maquina7 en modelo acompañar pieza lineal

Para finalizar, falta la descripción de la zona3. Como en el caso de la zona2, se necesita coger de world para evitar que el op\_8\_9\_10\_sal tuviera que esperar casi todo el proceso, dado que tendría que pasar la pieza por las dos zonas anteriores. Por esta razón, se ha implementado que en el instante 20 coja la pieza conjunto (dado que a la salida de la maquina7 ya no es pieza\_prin, es conjunto) del world, como se puede apreciar en la Figura 3.50. Como ocurriese en el caso de la maquina4, en el resto de los casos, el ciclo comenzará

```

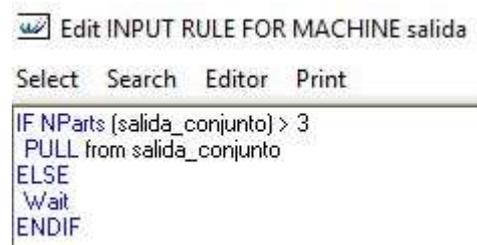
Edit INPUT RULE FOR MACHINE Maquina8
Select Search Editor Print
IF TIME = 20
PULL from conjunto out of WORLD
ELSEIF operacion_op3 = 0
PULL from conjunto out of Maquina7 With op8_9_10_sal#(1) Using Path
ELSE
Wait
ENDIF
    
```

Figura 3.50: descripción de la maquina8 en el modelo acompañar pieza lineal

cuando **operación\_op2** tome valor 0. Para finalizar esta zona, se dispone la salida de piezas del proceso, se dispone de una máquina ficticia, **salida\_proceso**, haciendo que el **op\_8\_9\_10\_sal** coja las piezas de la **maquina10** cuando la variable **operación\_op3** toma valor 3, depositando las piezas en el **cont\_salida**. En este contenedor se ha implementado una variable denominada **piezas\_salida** para conocer las piezas se han generado en el proceso. Cuando acabe la actividad de **salida\_proceso**, la variable **operación\_op3** tomará valor 0 para que el operario vuelva a la **maquina8**.

Después de haber descrito cómo funcionan la **maquina4** y **maquina8**, cabe destacar que los tiempos que se han indicado para realizar la primera actividad (80 y 20) han sido determinados para que las tres zonas se encontrasen sincronizadas, es decir, para que ninguno de los operarios tuviera que esperar en el siguiente ciclo la pieza. En algunas ocasiones se puede apreciar cómo el operario acude a su posición de reposo en vez de ir recoger la pieza a la máquina anterior del comienzo de su zona. Esto es debido a que cuando el operario acaba la actividad y su correspondiente variable **operación\_op** toma valor 0, el operario de la zona anterior no ha acabado su actividad, haciendo que no haya pieza en la máquina y no pueda ir el operario a recogerla. En estos casos, los operarios no usarán los paths que se han implementado, ni para ir desde la última máquina a la zona de reposo ni desde la zona de reposo a coger la pieza.

Para acabar la descripción del modelo acompañar pieza lineal, se ha decidido que un vehículo saque las piezas que se depositen en el **cont\_salida\_proceso**, sacando una pieza cada vez y depositándola en el contenedor **salida\_proceso**. Para evitar acumulaciones en este contenedor se dispone de una máquina ficticia denominada **salida**, que envía una pieza a **ship** cuando en el contenedor **salida\_proceso** haya al menos 3 piezas, como muestra la Figura 3.51.



```

Edit INPUT RULE FOR MACHINE salida
Select Search Editor Print
IF NParts (salida_conjunto) > 3
PULL from salida_conjunto
ELSE
Wait
ENDIF

```

Figura 3.51: salida de piezas de los modelos acompañar pieza

## 2.2. IMPLEMENTACIÓN DE LA CÉLULA EN U

Una vez que se ha descrito cómo es el modelo en el que los operarios tienen que llevar la pieza de una a otra máquina, estando éstas dispuestas de forma lineal, se ha podido apreciar cómo de largo se hace el proceso y, por tanto, la elevada cantidad de tiempo que se pierde en los trayectos entre el final y el comienzo de cada una de las zonas. Por esta razón, se ha dispuesto el proceso en forma de U. Este proceso se puede apreciar en la Figura 3.52, en donde se puede comprobar cómo las máquinas se encuentran dispuestas en U. En este modelo se puede apreciar más fácilmente las zonas que componen el proceso, estando la primera zona compuesta por las máquinas ficticias **entrada\_proceso** y **salida\_proceso**, y las **maquina1** y **10**, estando estas máquinas gobernadas por el **op\_ent\_1\_10\_sal** y la variable **operación\_op1**. La segunda zona está compuesta por las **maquinas2-3-8-9**, necesitando al **op2\_3\_8\_9** y siendo realizadas las actividades por el valor de la variable **operación\_op2**. La última zona es la zona3, dada por las **maquinas4-5-6-7** y por el operario **op4\_5\_6\_7** y la variable **operación\_op3**.

Como en el caso anterior, se dispone de una salida de piezas del proceso, realizada con el **vehículo**, las vías **ida\_vehiculo** y **vuelta\_vehiculo**, así como el contenedor **salida\_conjunto** y la máquina ficticia **salida**.

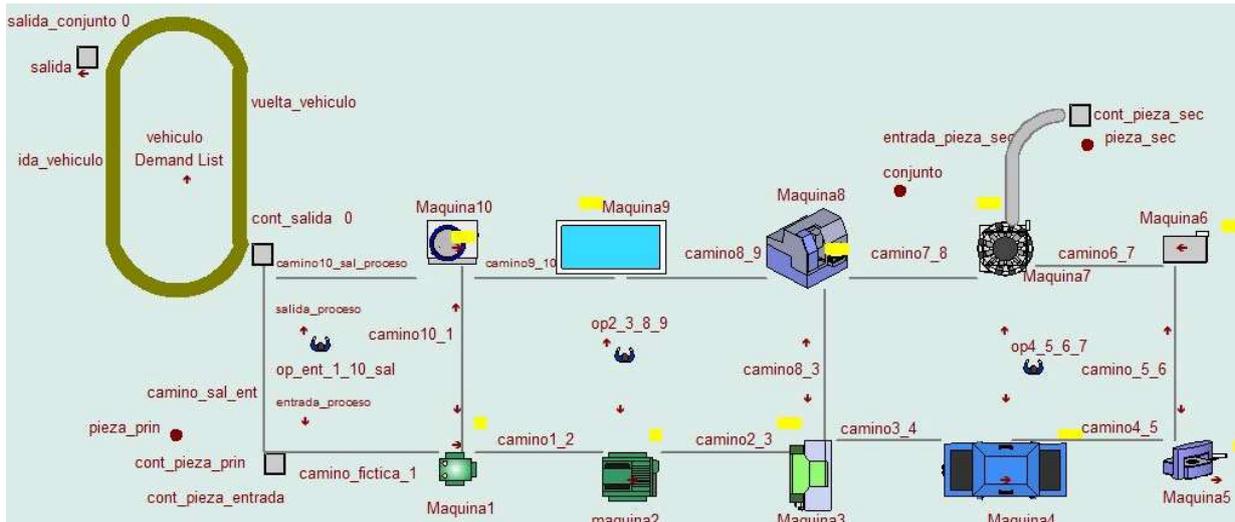


Figura 3.52: representación del modelo acompañar pieza en U

Cabe destacar que, para un mejor funcionamiento del proceso, se ha decidido que éste comience con piezas en la **maquina2**, en la **maquina6** y en la **maquina10**, de esta forma, se evitan largos tiempos de espera hasta que el proceso comience a funcionar estacionariamente. Como en el caso anterior, se ha considerado que el operario debe realizar una acción previa en las máquinas, para simular que el operario realiza una acción en éstas y sea necesario en el transporte de la pieza de una a otra máquina.

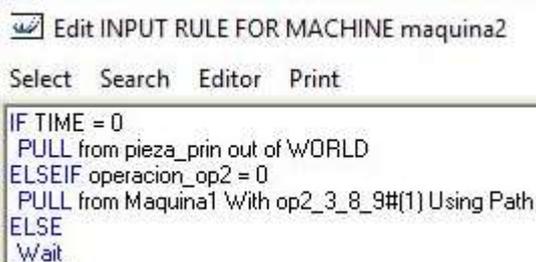
Una vez que se conoce la distribución del proceso, se procederá a explicar cómo funciona, dado que posee ciertas diferencias con el modelo anterior. Para comenzar, al variar la disposición de las máquinas, se ha modificado las máquinas que están asignadas a cada una de las variables, la nueva asignación de variables se puede apreciar en la Tabla 3.6. Además, cabe destacar que existe un valor más que puede tomar la variable, asignado a la última zona, usado para llevar la pieza desde la última máquina de la correspondiente zona a la primera de la siguiente zona.

Valor	Maquina correspondiente		
	Operación_op1	Operación_op2	Operación_op3
0	Entrada_proceso	2	4
1	1	3	5
2	Salida_proceso	9	6
3		Llevar a maquina10	7
4	-		Llevar a maquina8

Tabla 3.6: valores que toman las variables que condicionan los movimientos de los operarios modelo acompañar pieza en U

En este caso, dado que las máquinas de las zonas no se encuentran consecutivas, se procederá a explicar el proceso siguiendo a la pieza, no la zona de la que se encarga cada operario (en el caso anterior coincidían las zonas con las máquinas consecutivas). Como en el caso anterior, la acción de introducir piezas en el proceso la realiza la máquina ficticia **entrada\_proceso**, gobernada en este caso por el operario **op\_ent\_1\_10\_sal**, haciendo que saque piezas del **con\_pieza\_entrada** cuando la variable **operación\_op1** toma valor 0, como en el caso lineal, cuando acaba la actividad, la variable toma valor 1, haciendo que el operario coja la pieza y la lleve a la **maquina1** mediante el path **entrada\_maquina**.

La zona 2 comienza con la **maquina2**, cuya entrada de piezas vendrá dada por la variable **operación\_op2** (Tabla 3.6), por tanto, el operario **op2\_3\_8\_9** se dirigirá a la **maquina1** a través del path **camino1\_2** para coger la **pieza\_prin**, como se aprecia en la Figura 3.53 y disponerla en la **maquina2**, al finalizar la actividad, **operación\_op2** tomará valor 1, Figura 3.54. Además, como se ha comentado anteriormente, se desea que la primera acción del **op2\_3\_8\_9** comience en la **maquina2**, para evitar grandes tiempos de espera hasta que llega la primera pieza, por este motivo, se ha implementado en la acción de entrada de esta máquina, que cuando el modelo se encuentre en el instante 0, coja **pieza\_prin** del **world**. Cuando acabe la actividad, el **op2\_3\_8\_9** tendrá que llevar la **pieza\_prin** a la **maquina3**, dado que se encuentra consecutivamente a ella y en la misma célula. La regla de entrada de la **maquina3** viene dada por 1 en el valor de la variable **operación\_op3**, realizando el trayecto por **camino2\_3** y cuando realice esta actividad.

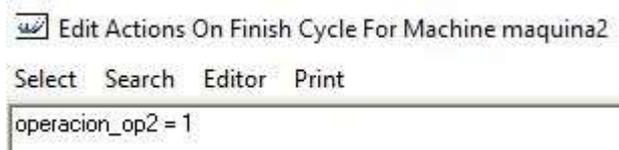


```

Edit INPUT RULE FOR MACHINE maquina2
Select Search Editor Print
IF TIME = 0
PULL from pieza_prin out of WORLD
ELSEIF operacion_op2 = 0
PULL from Maquina1 With op2_3_8_9#(1) Using Path
ELSE
Wait

```

Figura 3.53: entrada de piezas en maquina2 modelo acompañar pieza en U



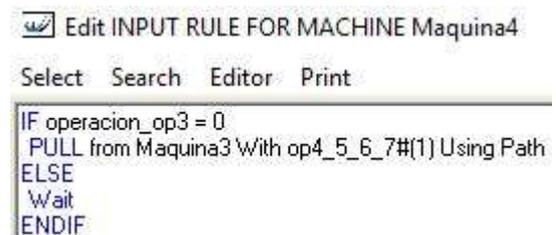
```

Edit Actions On Finish Cycle For Machine maquina2
Select Search Editor Print
operacion_op2 = 1

```

Figura 3.54: definición de próximo valor de la operacion\_op2 en modelo acompañar pieza U

Cuando se acabe la actividad en la **maquina3**, la pieza deberá pasar a la siguiente zona, es decir, a la zona3, y el operario **op4\_5\_6\_7** deberá coger la pieza mediante el path **camino3\_4**, Figura 3.55, para llevar la pieza desde la **maquina3** a la **maquina4**. Cabe destacar que el operario sólo cogerá la pieza de la **maquina3** cuando la variable **operación\_op3** tome valor 0; es decir, puede que la **maquina3** ya haya procesado la **pieza\_prin** y **op4\_5\_6\_7** esté en otra máquina porque la variable **operación\_op3** no posea el valor 0 y, por tanto, no irá a recoger la pieza hasta que no acabe el ciclo de la zona y la variable **operación\_op2** vuelva a tomar valor 0. Además, cuando el **op4\_5\_6\_7** realice la operación de la **maquina4**, la variable **operación\_op3** tomará valor 1.



```

Edit INPUT RULE FOR MACHINE Maquina4
Select Search Editor Print
IF operacion_op3 = 0
PULL from Maquina3 With op4_5_6_7#(1) Using Path
ELSE
Wait
ENDIF

```

Figura 3.55: regla de entrada maquina4 en modelo acompañar pieza U

Cuando **op4\_5\_6\_7** acabe la operación de la **maquina4**, tendrá que realizar la operación previa de la **maquina5** (de igual forma que la **maquina4**, siendo la condición de entrada con el valor de la variable **operación\_op3**, 1 y al finalizar la actividad ésta tomará valor 2). En cuanto a la **maquina6**, como se ha comentado anteriormente, la primera operación que se realiza en esta zona del **op4\_5\_6\_7**, se lleva a cabo en esta máquina, por tanto, se ha estimado que debe llegar una **pieza\_prin** (que es la que se procesa en la **maquina6**) en el instante 11 (Figura 3.56), para que quede sincronizado con el resto de las zonas. En el resto de los casos, cogerá la pieza de la **maquina5** a través el **camino5\_6**. Al finalizar esta máquina, la variable **operación\_op3** tomará valor 3, como muestra la Figura 3.57.

```

Edit INPUT RULE FOR MACHINE Maquina6
Select Search Editor Print
IF TIME = 11
PULL from pieza_prin out of WORLD
ELSEIF operacion_op3 = 2
PULL from Maquina5 With op4_5_6_7#(1) Using Path
ELSE
Wait
ENDIF
    
```

Figura 3.56: definición de entrada de piezas maquina6 modelo acompañar piezas U

```

Edit Actions On Finish Cycle For Machine maquina6
Select Search Editor Print
operacion_op2=3
    
```

Figura 3.57: variación variable operacion\_op2 salida maquina6 modelo acompañar pieza U

Cuando la **maquina6** acaba su actividad, se procederá a realizar el ensamblaje en la **maquina7**, que se realiza de la misma forma que en el modelo lineal, salvo que cogerá ambas piezas cuando la variable **operación\_op3** tome valor 3. Cuando se realice el ensamblaje, el operario tendrá que llevar el conjunto a la **maquina8** (Figura 3.58), dado que el **op4\_5\_6\_7** se debe de encargar de todas las acciones que se encuentran en la zona y **camino7\_8** se encuentra en la zona 3, como indica la Figura 3.52. Por tanto, a la salida de la **maquina7**, la variable **operación\_op3** debe cambiar su valor (Figura 3.59).

```

Edit INPUT RULE FOR MACHINE Maquina8
Select Search Editor Print
IF operacion_op3 = 4
PULL from Maquina7 With op4_5_6_7#(1) Using Path
ELSE
Wait
ENDIF
    
```

Figura 3.58: entrada de piezas en la maquina8 modelo acompañar pieza U

```

Edit Actions On Finish Cycle For Machine maquina7
Select Search Editor Print
operacion_op3=4
    
```

Figura 3.59: variación de la variable operacion\_op3 en la maquina7 modelo acompañar pieza U

En el momento en el que el **op4\_5\_6\_7** deje la pieza en la **maquina8** comienza de nuevo la zona2, haciendo que el **op2\_3\_8\_9** se desplace hacia la zona superior, para realizar la tarea previa de la **maquina8**. Pero antes de continuar el **op2\_3\_8\_9** con su actividad en la **maquina8**, será necesario finalizar el ciclo de la zona3; esto se consigue cuando la variable **operación\_op3** toma valor 0 (Figura 3.60), haciendo que se inicie de nuevo el ciclo de la zona 3 (moviéndose **op4\_5\_6\_7** por el **camino8\_3**, llegando a la **maquina3** para coger la pieza y posteriormente llevarla a la **maquina4** por su path correspondiente).

```

Edit Actions On Input to Cycle For Machine Maquina8
Select Search Editor Print
operacion_op3 = 0
    
```

Figura 3.60: fin de ciclo zona3 modelo acompañar pieza U

Una vez que se ha definido cómo finalizar el ciclo de la zona 3, se procederá a explicar cómo es el funcionamiento de la zona 2: el funcionamiento de la **maquina8** ya se ha explicado, el **op4\_5\_6\_7** lleva el conjunto a la **maquina8**, según la regla de entrada que se puede apreciar en la Figura 3.58. Además, al finalizar la actividad, habrá que definir que el conjunto pase a la **maquina9**, haciendo que la variable **operación\_op2** tome valor 2, el **op2\_3\_8\_9** se trasladará a la **maquina9** dado que en la regla de entrada indica que, debe coger el **conjunto** cuando esta variable toma valor 2 y llevarlo a la **maquina9**. Como ocurriese en la zona3, el camino para llevar el **conjunto** hasta la **maquina10**, se encuentra en la zona2; por tanto, **op2\_3\_8\_9** deberá llevar la pieza a la **maquina10**. De esta forma, en la salida de la **maquina9** la variable **operación\_op2** toma valor3 (Figura 3.61), y la entrada de la **maquina10** se realiza cuando la variable tome ese valor.

```

Edit Actions On Finish Cycle For Machine maquina9
Select Search Editor Print
operacion_op2=3
    
```

Figura 3.61: acción de salida maquina9 modelo acompañar pieza U

Para finalizar, se procederá a explicar cómo es el final de la zona1 que comienza en la **maquina10**. Como se ha comentado anteriormente, el conjunto llegará a través del **op2\_3\_8\_9** a la **maquina10**, igual que en el caso de la zona3, y, por tanto, el final del ciclo de la zona2 se produce cuando se realiza la entrada del **conjunto** en la **maquina10**, haciendo que la variable **operación\_op2** tome valor 0, Figura 3.62, y que el **op2\_3\_8\_9** vaya a la **maquina1** para coger la **pieza\_prin**. En la zona1, la primera actividad que se realiza es en la **maquina10** en el instante 5, cogiendo la pieza que se procesa en esa máquina, el **conjunto**, como muestra la Figura 3.63. Al finalizar esta operación la variable **operación\_op1** tomará valor 2 (Figura 3.64), haciendo que el operario lleve el conjunto a la máquina ficticia **salida\_proceso**; al finalizar esta acción, el operario volverá a coger la **pieza\_prin** y seguirá con el ciclo de la zona1

Edit Actions On Input to Cycle For Machine Maquina10

Select Search Editor Print

operacion\_op2 = 0

Figura 3.62: acción a la entrada de la maquina10 modelo acompañar pieza U

Edit Actions On Finish Cycle For Machine Maquina10

Select Search Editor Print

operacion\_op1 = 2

Figura 3.64: acción a la salida maquina10 modelo acompañar pieza U

Edit INPUT RULE FOR MACHINE Maquina10

Select Search Editor Print

```
IF TIME = 5
PULL from conjunto out of \WORLD
ELSEIF operacion_op2 = 3
PULL from Maquina8 \With op2_3_8_9#{1} Using Path
ELSE
Wait
ENDIF
```

Figura 3.63: regla de entrada en la maquina10 modelo acompañar pieza U

El final del proceso se realiza de la misma forma que en el modelo lineal, sacando el **conjunto** del proceso mediante un **vehículo** y depositándolo en un contenedor denominado **salida\_conjunto**. Posteriormente, el conjunto saldrá del proceso hacia **ship**.

### 3. TÉCNICA HEIJUNKA

#### 3.1. DESCRIPCIÓN DEL MODELO

Se posee un proceso productivo al que llegan tres piezas distintas: **pieza1**, **pieza2** y **pieza3**. También se poseen dos zonas: una común, en la que se realizan operaciones a los tres tipos de piezas, y una específica, de almacenaje previo, constituida por dos contenedores para cada tipo de pieza, que permitirán realizar una serie de operaciones a cada tipo de piezas. En la Figura 3.65 se puede apreciar cómo es el modelo y las dos zonas que se han definido.

Al finalizar la zona de procesado específico, se dispone de una máquina que realiza conjuntos de las piezas que se han producido, dependiendo de las necesidades de unos clientes; es decir, el número de piezas de cada tipo que formarán el conjunto dependerá de la demanda de unos clientes. Para finalizar, se dispondrán los distintos conjuntos en su contenedor correspondiente, es decir, en el contenedor de cada cliente. Cuando en el contenedor haya el número de conjuntos deseados por el cliente, se les sacará del contenedor hacia **ship**.

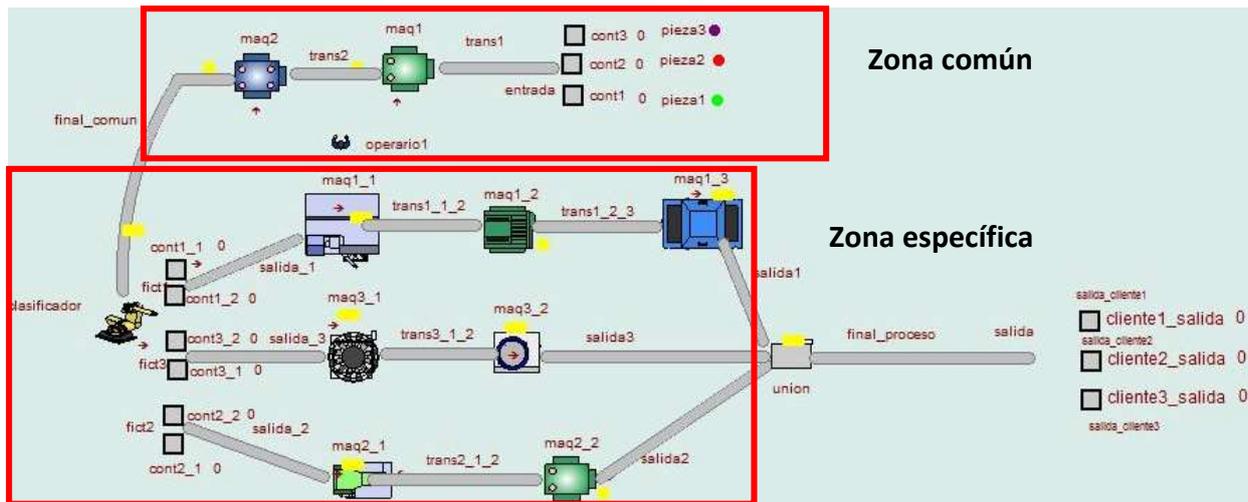


Figura 3.65: representación del modelo técnicas Heijunka y Smed

Para comenzar a explicar el proceso, se indicarán las necesidades de los clientes, que serán los que determinen las piezas que hay que realizar. Como se ha comentado anteriormente, se poseen tres clientes, dados por las variables tipo vectores: **cliente1**, **cliente2** y **cliente3**. Cada uno de los clientes posee unas necesidades distintas, siendo distinto el número de piezas de cada tipo que forman el conjunto (la primera posición del vector se refiere a la **pieza1**, la segunda a la **pieza2** y la tercera a la **pieza3**), el número total de piezas que lo forman y el número de conjuntos que van a necesitar. En la Tabla 3.7 se puede apreciar las necesidades de cada cliente, así como qué piezas forman sus necesidades.

Cliente	Pieza1	Pieza2	Pieza3	Numero de conjuntos
Cliente1	3	2	1	10
Cliente2	4	3	2	6
Cliente3	4	1	2	4

Tabla 3.7: necesidades de los clientes modelos técnica Heijunka y Smed

Una vez que se conocen cuáles son las necesidades de los clientes, se podrá proceder a explicar las piezas que llegan al proceso. Como se ha comentado anteriormente, llegan tres piezas, **pieza1**, **pieza2** y **pieza3**. Éstas poseen diferente frecuencia de llegada al modelo, dado que, como se ha podido apreciar en la Tabla 3.7, hay piezas de las que se posee una mayor demanda que de otras, como es el caso de la **pieza1**. Por esta razón, la llegada de piezas al modelo no es la misma para todas las piezas, así como el lote de piezas que llegan, como indica la Tabla 3.8. Además, para poder visualizar mejor qué tipo de pieza es, tanto en el modelo 3D como en el 2D, se han definido tres representaciones distintas, cada una asociada a un tipo de pieza, como se aprecia en la Figura 3.66, Figura 3.67 y Figura 3.68. Cabe destacar, que se ha tenido que crear un atributo denominado **tipo**, para que en la entrada de las piezas a las máquinas se sepa de qué tipo de pieza se trata; por esta razón, a la hora de crear las piezas, se ha definido el atributo que debe tener cada una para asociarla un tipo de pieza, que se puede apreciar en la Tabla 3.8.

Pieza	Tamaño lote	Tiempo entre llegadas (s)	Acciones al crear
Pieza1	30	750	tipo=1
Pieza2	20	750	tipo=2
Pieza3	15	750	tipo=3

Tabla 3.8: definición de entrada de piezas al modelo sin heijunka ni smed

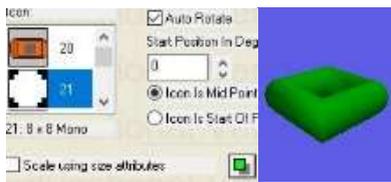


Figura 3.66: representación de la pieza1 modelos técnica heijunka



Figura 3.67: representación de la pieza2 modelos técnica heijunka

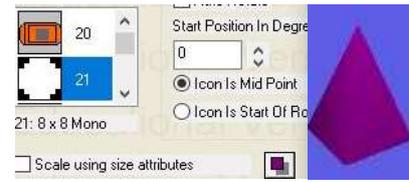


Figura 3.68: representación de la pieza3 modelos técnica heijunka

Antes de comenzar las máquinas de la zona común, como se aprecia en la Figura 3.65, se disponen tres contenedores, uno asociado a cada tipo de pieza. Como se trata de un modelo en el que no se ha implementado ni técnicas Smed ni Heijunka, se trabajará con grandes lotes de piezas. Para realizar la entrada de piezas al proceso se dispone de una máquina ficticia denominada **entrada**, que cogerá un tipo de pieza u otro dependiendo de cuántas haya cogido ya de cada tipo, Figura 3.69. Para ello se han dispuesto tres variables contador, denominadas **entrada\_tipo1**, **entrada\_tipo2** y **entrada\_tipo3**, que irán acumulando el número de piezas que se han cogido de cada tipo, como se aprecia el incremento en la Figura 3.70. Así, cuando se llegue al límite de cada tipo de pieza estipulado por la regla de entrada, se deja de coger de ese tipo y se coge de otro. Cuando se hayan cogido 9 **piezas3**, se iniciarán los contadores, Figura 3.71 y se comenzará de nuevo el ciclo.

```

Edit INPUT RULE FOR MACHINE
IF entrada_tipo1 <= 29
  PULL from cont1
ELSEIF entrada_tipo2 <= 19
  PULL from cont2
ELSEIF entrada_tipo3 <= 14
  PULL from cont3
ELSE
  Wait
ENDIF

Edit Actions On Input to Cycle
IF tipo = 1
  entrada_tipo1 = entrada_tipo1 + 1
ELSEIF tipo = 2
  entrada_tipo2 = entrada_tipo2 + 1
ELSEIF tipo = 3
  entrada_tipo3 = entrada_tipo3 + 1
ENDIF

Edit Actions On Output from Cycle
IF entrada_tipo3 = 15
  entrada_tipo1 = 0
  entrada_tipo2 = 0
  entrada_tipo3 = 0
ENDIF
    
```

Figura 3.69: regla de entrada al modelo sin heijunka ni smed

Figura 3.70: incremento contador entrada modelo sin heijunka ni smed

Figura 3.71: inicialización de contadores entrada modelo sin heijunka ni smed

Llegados a este punto, ya se disponen las piezas en la cinta de entrada a la **maq1**. Como se ha comentado anteriormente, las máquinas de la zona común realizan la operación a los diferentes tipos de piezas; por esta razón, antes de realizar el procesamiento de una pieza que no posee el mismo **tipo** que las anteriores, será necesario realizar una actividad previa en la máquina. Esta actividad vendrá dada por un **setup** que hará que el operario tenga que acudir a la máquina cuando cambia el atributo **tipo**. Como se trata de un modelo en el que no se han implementado Smed, la actividad asociada al cambio de pieza tardará mucho tiempo en realizarse, como se puede apreciar en la Figura 3.72. Además, será necesario que lo realice el **operario1**. En la **maq2** se procederá de igual forma, realizándose una actividad en la máquina cuando cambie el **tipo** de pieza.

Machine - maq1

Description	Setup Mode				Setup Duration					
	Mode	No. of Operations	Ops to First Setup	Expression	Setup with no part	Station Number	Actions on Start	Labor Rule	Setup Time	Action Finish
1 New Setup	Value Ch			tipo	<input type="checkbox"/>		N	Y	60.0	N

Figura 3.72: operación cambio de pieza maq1 en modelo sin heijunka ni smed

Después de realizar las operaciones de la zona común, se procederá a explicar el funcionamiento de la zona específica a cada uno de los tipos. Para comenzar, se dispondrá de una máquina denominada **clasificador** que es la encargada de colocar cada tipo de pieza en su zona correspondiente. A la entrada de cada línea asociada a un tipo de pieza, se disponen dos contenedores para disponer las piezas en modo de stock intermedio. Para comenzar a explicar el funcionamiento del clasificador, será necesario conocer cómo se comportan los contenedores. La salida de piezas de los contenedores las realizarán máquinas ficticias, denominadas **fict1**, **fict2** y **fict3**: actuarán cuando en la cinta de entrada de cada zona específica haya menos de 3 piezas, como se puede apreciar en la Figura 3.73. Para conocer de qué contenedor han salido las últimas piezas (para no colocar siempre las piezas en el mismo contenedor) se ha implementado una variable en la salida de piezas de cada tipo de contenedor, tomando un valor u otro dependiendo de qué contenedor salga. Por ejemplo, cuando salgan piezas del **cont1\_2** porque ya hay 3 piezas, la variable **cant\_cont\_1\_1** tomará valor 0, como indica la Tabla 3.9.

```

Edit INPUT RULE FOR MACHINE fict1
Select Search Editor Print
IF NParts (salida_1) < 3
IF NParts (cont1_1) = 3
MATCH/ANY
cont1_1 #[3]
ELSEIF NParts (cont1_2) = 3
MATCH/ANY
cont1_2 #[3]
ENDIF
ELSE
Wait
ENDIF
    
```

Figura 3.73: salida de piezas en los contenedores en el modelo heijunka

Variable	Contenedor asociado a valor 0	Contenedor asociado a valor 1
Cant_cont_1_1	Cont1_2	Cont1_1
Cant_cont_2_1	Cont2_2	Cont2_1
Cant_cont_3_1	Cont3_2	Cont3_1

Tabla 3.9: valor de las variables a la salida de los contenedores almacenaje modelo heijunka

Conocido cómo funcionan los contenedores, se retomará la explicación del **clasificador**. Éste primero reconocerá el tipo de pieza que es, para disponerlo en la zona que le corresponde y, posteriormente, lo depositará en un contenedor u otro dependiendo de la cantidad de piezas que hay en cada uno. Se ha querido simular que cuando acabe de llenar uno de los contenedores, pase a rellenar el siguiente; de ahí la necesidad de indicar que lo deposite en el que tenga menos de 3 piezas y que, además, deba cumplir la condición de las últimas piezas que han salido sean del otro contenedor (asegurando que no siempre se coloca en el primer contenedor, dado que es el primero en la condición), como se puede apreciar en la Figura 3.74.

```

Edit OUTPUT RULE FOR MACHINE clasificador
Select Search Editor Print
IF tipo = 1
IF NParts (cont1_1) < 3 AND NParts (cont1_2) = 0 AND cant_cont_1_1 = 0
PUSH pieza1 to cont1_1
ELSEIF NParts (cont1_2) < 3 AND NParts (cont1_1) = 0 AND cant_cont_1_1 = 1
PUSH pieza1 to cont1_2
ENDIF
ELSEIF tipo = 2
IF NParts (cont2_1) < 3 AND NParts (cont2_2) = 0 AND cant_cont_2_1 = 0
PUSH pieza2 to cont2_1
ELSEIF NParts (cont2_2) < 3 AND NParts (cont2_1) = 0 AND cant_cont_2_1 = 1
PUSH pieza2 to cont2_2
ENDIF
ELSEIF tipo = 3
IF NParts (cont3_1) < 3 AND NParts (cont3_2) = 0 AND cant_cont_3_1 = 0
PUSH pieza3 to cont3_1
ELSEIF NParts (cont3_2) < 3 AND NParts (cont3_1) = 0 AND cant_cont_3_1 = 1
PUSH pieza3 to cont3_2
ENDIF
ELSE
Wait
ENDIF
    
```

Figura 3.74: funcionamiento del clasificar modelos técnica heijunka

En cada una de las zonas específicas se van a realizar una serie de operaciones en las piezas (no será necesario realizar operaciones previas a las máquinas dado que siempre procesan el mismo tipo de pieza). Al finalizar cada una de las zonas específicas, las piezas se disponen en unas cintas denominadas **salida1**, **salida2** y **salida3**. Estas cintas llevarán las piezas a la entrada de la máquina **unión**, que es la encargada de realizar los conjuntos en función de las

necesidades de los clientes. Además, como se ha comentado en la Tabla 3.7, los clientes necesitan un número específico de conjuntos. Cuando se cumplan las necesidades de todos los clientes, se iniciará el ciclo de nuevo. Cabe destacar que los conjuntos que se generan tomarán diferente representación dependiendo de a quién va dirigido, tanto en 2D como en 3D, como se puede apreciar en las Figura 3.75, Figura 3.76 y Figura 3.77.

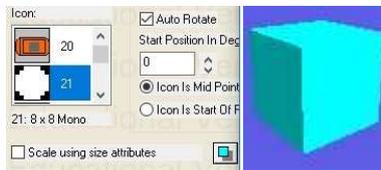


Figura 3.75: representación lote\_cliente1 modelo heijunka

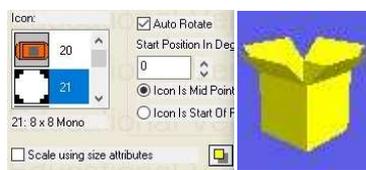


Figura 3.76: representación lote\_cliente2 modelo heijunka



Figura 3.77: representación lote\_cliente3 modelo heijunka

Cada uno de los conjuntos que se generan lleva asociado un atributo denominado **tipo\_cliente**, tomando valor 1 cuando el conjunto es el asociado al **cliente1**, valor 2 cuando es para el **cliente2** y valor 3 para el **cliente3**.

Una vez que se conocen las características que deben poseer los conjuntos, se procederá a explicar cómo es el funcionamiento de la máquina de ensamblaje **unión**. Se ha definido una variable denominada **ensamblaje**, dado que el número de piezas que se necesita en cada conjunto es diferente (el número de piezas para el **cliente1** son 6, para el **cliente2** son 9 y para el **cliente3** 7). De esta forma, el número que define las piezas que entran en la máquina viene determinado por la variable **ensamblaje**; así, el número de piezas de entrada a la máquina puede variar. Además, en la **unión** también se ha definido una variable denominada **numero\_operaciones**, que nos indicará el número de operaciones que se han realizado y, en consecuencia, qué tipo de conjunto hay que realizar (al tratarse de un modelo sin Smed ni Heijunka, se realizarán 10 operaciones para el **cliente1**, 6 para el **cliente2** y 4 para el **cliente3**).

Una vez hechas todas estas consideraciones, se explicará cómo es la entrada de estas piezas en **union**. Para comenzar, el primer conjunto que se realiza en esta máquina es para **cliente1**; por esta razón, la acción inicial de **unión** es para indicar que la cantidad que debe realizar es la del **cliente1**, como muestra la Figura 3.78.

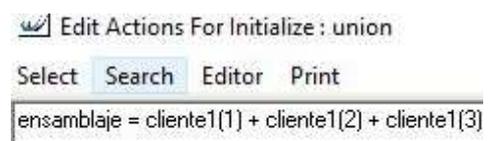


Figura 3.78: primer tipo a realizar por unión modelos técnica heijunka

La regla de entrada de esta máquina viene dada por la Figura 3.79. Como se puede apreciar, siempre coge las piezas del mismo sitio, pero la cantidad de éstas varían (dadas por qué cliente es) en función de la variable **numero\_operacion**. Como se ha comentado anteriormente, se ha indicado el valor del número de piezas que debe coger la máquina para la primera acción mediante la variable **ensamblaje**, pero hay que definir cuántas coge para las operaciones posteriores; para ello, se ha implementado esta variable en las acciones de entrada de la máquina. Dependiendo de **numero\_operaciones**, el valor de **ensamblaje** variará. Además, se ha definido una variable denominada **tipocliente**, que variará cuando ya se han realizado todos los conjuntos para cada cliente, como se puede apreciar en la Figura 3.80. De esta forma, cuando se comience a fabricar para otro cliente, variará la variable **tipocliente** y habrá que realizar la acción previa en la máquina, debido al cambio del tipo de conjunto que hay que realizar. Como en los casos de la zona común, al no estar implementada ni la técnica Smed ni

Heijunka, se tardará un elevado tiempo para realizar la acción previa, como indica la Figura 3.81.

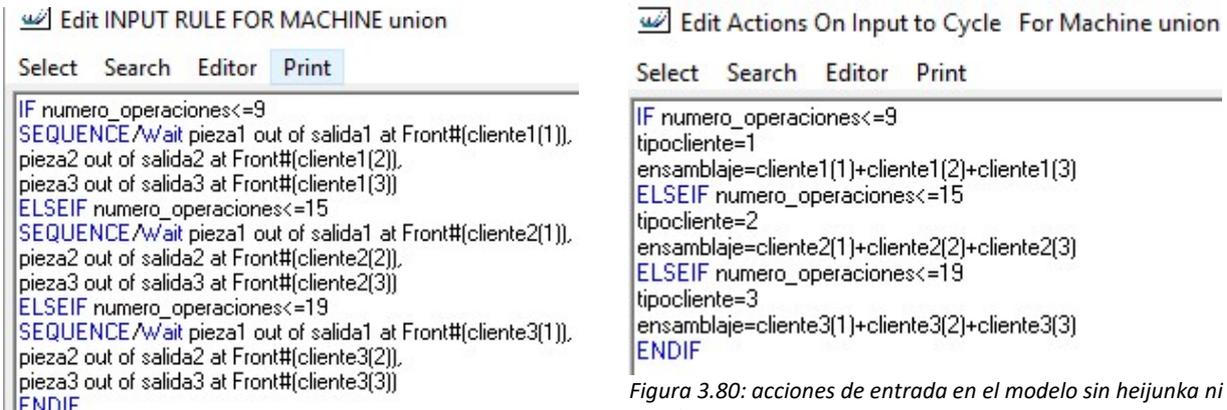


Figura 3.79: regla de entrada en el modelo sin heijunka ni smed

Figura 3.80: acciones de entrada en el modelo sin heijunka ni smed

Detail Machine - union

Setup Mode											
Description	Mode	No. of Operations	Ops to First Setup	Expression	Setup with no part	Station Number	Actions on Start	Labor Rule	Setup Time	Actions on Finish	Setup Duration
1 New Setup	Value Ch			tipocliente	<input type="checkbox"/>		N	N	50.0	N	

Figura 3.81: acción previa cambio de tipo de pack en unión en el modelo sin heijunka ni smed

Para finalizar de explicar la **unión** de las piezas, se procerá a explicar lo que ocurre al finalizar la actividad de esta máquina. Lo primero que hay que indicar es que al finalizar el proceso, las piezas que aparezcan no serán **pieza1**, **pieza2** y **pieza3**, sino **lote\_cliente1**, **lote\_cliente2** y **lote\_cliente3** (cabe destacar que a estas nuevas piezas se las ha asociado un atributo, **tipo\_cliente**), pudiéndose apreciar en la Figura 3.82 cómo es el cambio de pieza. Para mejorar la visualización en el modelo 2D, se ha definido que el icono tome el mismo color que el que posee en el modelo 3D; de esta forma, se ha indicado que el conjunto para el **cliente1** tome color azul (PEN=22), para el **cliente2** amarillo (PEN =19) y para el **cliente3** magenta(PEN =21). Además, al finalizar la actividad habrá que indicar que la variable **numero\_operaciones** ha aumentado en una unidad su valor, haciendo que cuando llegue a 20 se inicialice a cero, para comenzar de nuevo el ciclo. Una conclusión importante en este modelo es conocer en qué momento se comienza a producir para cada cliente, así como cuándo le llega el último lote. Para ello, se han implementado tres variables de tipo vector, **tiempo\_cliente()**, con dos valores, en donde se almacenará, en el primer dato, la

```

IF numero_operaciones <= 9
CHANGE ALL to conjunto_cliente1
PEN = 22
ELSEIF numero_operaciones <= 15
CHANGE ALL to conjunto_cliente2
PEN = 19
ELSEIF numero_operaciones <= 20
CHANGE ALL to conjunto_cliente3
PEN = 21
ENDIF
IF numero_operaciones = 0
tiempo_cliente1(1) = 0
tiempo_cliente1(2) = 0
tiempo_cliente2(1) = 0
tiempo_cliente2(2) = 0
tiempo_cliente3(1) = 0
tiempo_cliente3(2) = 0
ENDIF
numero_operaciones = numero_operaciones + 1
IF numero_operaciones = 20
tiempo_cliente3(2) = TimeEnteredModel ()
numero_operaciones = 0
ELSEIF numero_operaciones = 1
tiempo_cliente1(1) = TimeEnteredModel ()
ELSEIF numero_operaciones = 10
tiempo_cliente1(2) = TimeEnteredModel ()
ELSEIF numero_operaciones = 11
tiempo_cliente2(1) = TimeEnteredModel ()
ELSEIF numero_operaciones = 16
tiempo_cliente2(2) = TimeEnteredModel ()
ELSEIF numero_operaciones = 17
tiempo_cliente3(1) = TimeEnteredModel ()
ENDIF
    
```

Figura 3.82: acciones al finalizar la acción unir en modelo sin heijunka ni smed

primera pieza que sale de ese cliente y, en el segundo valor, cuándo se produce la última pieza del lote. Cuando se han realizado todos los lotes, se inicia el nuevo ciclo para realizar los 20 nuevos conjuntos y se inicializarán estas variables a 0, para que se conozcan los nuevos tiempos.

Ya se disponen los conjuntos en la salida de la máquina. Por tanto, habrá que colocarlos a su cliente correspondiente. Para ello, se dispone de una máquina ficticia denominada *salida*, que dispondrá las piezas en el contenedor correspondiente a cada cliente, *cliente1\_salida*, *cliente2\_salida* y *cliente3\_salida*. Para disponer los conjuntos en uno u otro contenedor, dependiendo del tipo de conjunto, se usará el atributo *tipo\_cliente* que se ha definido a la hora de la creación de los *lotes\_clientes*. Esto se puede apreciar en la Figura 3.83.

```

Edit OUTPUT RULE FOR MACHINE salida
Select Search Editor Print
IF tipo_cliente = 1
  PUSH to cliente1_salida
ELSEIF tipo_cliente = 2
  PUSH to cliente2_salida
ELSE
  PUSH to cliente3_salida
ENDIF
    
```

Figura 3.83: disposición de los lotes en su contenedor correspondiente en modelo heijunka

Posteriormente, cuando en el contenedor se dispongan las cantidades necesarias para cada cliente (10 en el caso del *cliente1*, 6 para el *cliente2* y 4 para el *cliente3*), se sacarán las piezas de los contenedores mediante máquinas ficticias, como se pueden apreciar en la Figura 3.65. Cada máquina sacará el número de piezas que necesitan cada cliente, como se puede apreciar en la Figura 3.84; en el caso del *cliente1*, se sacarán 10, que se depositarán en *ship*.

```

Edit INPUT RULE FOR MACHINE salida_cliente1
Select Search Editor Print
IF NParts (cliente1_salida) = 10
  MATCH/ANY
  cliente1_salida #(10)
ELSE
  Wait
ENDIF
    
```

Figura 3.84: disposición de los lotes en su contenedor correspondiente en modelo heijunka

### 3.2. IMPLEMENTACIÓN DE LA TÉCNICA HEIJUNKA

En el modelo anterior se trabajaba con lotes muy grandes de piezas y sin que se aplicasen técnicas Smed, lo que suponía grandes tiempos de preparación de las máquinas asociadas al cambio de tipo de piezas o de lotes en las máquinas. Por esta razón, se ha decidido implementar la técnica Smed, disminuyendo los tiempos asociados a los cambios del *tipo* de pieza, así como la técnica Heijunka, disminuyendo el tamaño de los lotes, tanto en el tamaño de los lotes de las piezas, Tabla 3.10, como el tamaño de los lotes de los clientes, Tabla 3.11. Cabe destacar que la implementación de las técnicas Smed y Heijunka no modificará la representación del modelo, puesto que son técnicas que se aplican en las máquinas sin modificar ningún otro aspecto.

Pieza	Tamaño lote	Tiempo entre llegadas	Acciones al crear
Pieza1	6	150	tipo=1
Pieza2	4	150	tipo=2
Pieza3	3	150	tipo=3

Tabla 3.10: definición de entrada de piezas al modelo con heijunka y smed

Cliente	Tamaño lote total	Lote realizado en cada etapa
Cliente1	10	5
Cliente2	6	3
Cliente3	4	2

Tabla 3.11: definición tamaños lote en union con heijunka y smed

Como se ha comentado anteriormente, el tamaño de los lotes que se procesan cada vez son menores; por esta razón, la entrada de piezas se realizará de forma similar a la descrita anteriormente aunque, en este caso, el número de piezas de cada tipo que se ha reducido, como se puede apreciar en la Figura 3.85, haciendo que la inicialización varíe, según la Figura 3.86.

```

Edit INPUT RULE FOR MACHINE entrada
Select Search Editor Print
IF entrada_tipo1 <= 5
  PULL from cont1
ELSEIF entrada_tipo2 <= 3
  PULL from cont2
ELSEIF entrada_tipo3 <= 2
  PULL from cont3
ELSE
  Wait
ENDIF
    
```

Figura 3.85: regla de entrada al modelo con heijunka y smed

```

Edit Actions On Finish Cycle For Machine entrada
Select Search Editor Print
IF entrada_tipo3 = 3
  entrada_tipo1 = 0
  entrada_tipo2 = 0
  entrada_tipo3 = 0
ENDIF
    
```

Figura 3.86: inicialización de contadores entrada modelo con heijunka y smed

Al aplicar la técnica Smed, se han disminuido los tiempos asociados a las actividades previas por la diferencia entre dos piezas consecutivas, ya sea debido al cambio del tipo de piezas, Figura 3.87, o al cambio del tipo de conjunto que se genera en la máquina **union**, Figura 3.88.

ail Machine - Maquina2

ail Machine - Maquina2											
General Setup Breakdowns Fluid Rules Shift Actions Costing Reporting Notes											
ID	Description	Setup Mode				Setup Duration					
		Mode	No. of Operations	Ops to First Setup	Expression	Setup with no part	Station Number	Actions on Start	Labor Rule	Setup Time	Actions on Finish
1	New Setup	Value Ch	Undefined	Undefined	tipo	<input type="checkbox"/>		N	N	5	N

Figura 3.87: operación cambio de pieza maq2 en modelo con heijunka y smed

tail Machine - union

tail Machine - union											
General Setup Breakdowns Shift Actions Costing Reporting Notes											
ID	Description	Setup Mode				Setup Duration					
		Mode	No. of Operations	Ops to First Setup	Expression	Setup with no part	Station Number	Actions on Start	Labor Rule	Setup Time	Actions on Finish
1	New Setup	Value Ch			tipocliente	<input type="checkbox"/>		N	N	20.0	N

Figura 3.88: acción previa cambio de tipo de conjunto en unión en el modelo con heijunka y smed

Como en el caso sin Heijunka, al acabar la zona común se dispone el **clasificador**, que actúa de igual forma que en el modelo sin Heijunka y Smed. La zona específica a cada tipo de pieza se comporta de igual forma que en el caso sin Heijunka ni Smed.

En el caso anterior, sin técnica Smed y Heijunka, la máquina encargada de hacer el ensamblaje de las piezas, **union**, no cambiaba de tipo de ensamblaje hasta que no se acababa con las necesidades de ese cliente. En el caso del modelo con Heijunka ni Smed, esta condición se ha cambiado, haciendo que los ensamblajes también los realice por lotes, como se indica en la Tabla 3.11. Por tanto, primero se realizarán

```

IF numero_operaciones <= 4
  tipocliente = 1
  ensamblaje = cliente1(1) + cliente1(2) + cliente1(3)
ELSEIF numero_operaciones <= 7
  tipocliente = 2
  ensamblaje = cliente2(1) + cliente2(2) + cliente2(3)
ELSEIF numero_operaciones <= 9
  tipocliente = 3
  ensamblaje = cliente3(1) + cliente3(2) + cliente3(3)
ELSEIF numero_operaciones <= 14
  tipocliente = 1
  ensamblaje = cliente1(1) + cliente1(2) + cliente1(3)
ELSEIF numero_operaciones <= 17
  tipocliente = 2
  ensamblaje = cliente2(1) + cliente2(2) + cliente2(3)
ELSEIF numero_operaciones <= 19
  tipocliente = 3
  ensamblaje = cliente3(1) + cliente3(2) + cliente3(3)
ENDIF
    
```

Figura 3.89: acción a la entrada modelo con heijunka y smed

5 ensamblajes para el **cliente1**, posteriormente otros 3 para el **cliente2** y, por último, 2 para el **cliente3**; una vez realizados éstos, se volverá a realizar los 5 para el **cliente1**, hasta completar las necesidades de 20 conjuntos en total. La acción a la entrada que definirá para qué cliente es el conjunto que se está fabricando (indicará la cantidad de piezas que debe coger la máquina, la variable **ensamblaje**) es la mostrada en la Figura 3.89.

Para finalizar, a la salida de la máquina hay que indicar que se ha generado el conjunto mediante una variación de la representación, tanto en 2D como en 3D, como ocurriese en el modelo sin Heijunka ni Smed. En este caso, se indicará que cambia el conjunto cuando ya han realizado los pequeños lotes correspondientes a cada cliente, dado por la variable **numero\_operaciones**. Como en el modelo sin Heijunka, también se modificará el icono que posee en la representación en 2D, mediante la función PEN. A la hora de conocer en qué instante se realizan los distintos conjuntos, se ha decidido crear un vector como en el otro modelo; en este caso, el vector posee tres valores, dando el valor intermedio el tiempo en el que se genera la mitad del lote correspondiente a cada cliente.

Cuando se inicia de nuevo el ciclo para generar otros 20 conjuntos, los vectores que almacenan el tiempo toman valor cero, para conocer el tiempo asociado a los siguientes lotes de 20, como se puede apreciar en la Figura 3.90.

Para finalizar, la salida de los conjuntos del proceso se realiza de la misma forma que para el modelo sin Heijunka ni Smed, retirando las piezas cuando en los contenedores se posee la cantidad total para cada cliente.

```

IF numero_operaciones <= 4
  PEN = 22
  CHANGE ALL to conjunto_cliente1
ELSEIF numero_operaciones <= 7
  PEN = 19
  CHANGE ALL to conjunto_cliente2
ELSEIF numero_operaciones <= 9
  PEN = 21
  CHANGE ALL to conjunto_cliente3
ELSEIF numero_operaciones <= 14
  PEN = 22
  CHANGE ALL to conjunto_cliente1
ELSEIF numero_operaciones <= 17
  PEN = 19
  CHANGE ALL to conjunto_cliente2
ELSEIF numero_operaciones <= 19
  PEN = 21
  CHANGE ALL to conjunto_cliente3
ENDIF
IF numero_operaciones = 0
  tiempo_cliente1(1) = 0
  tiempo_cliente1(2) = 0
  tiempo_cliente1(3) = 0
  tiempo_cliente2(1) = 0
  tiempo_cliente2(2) = 0
  tiempo_cliente2(3) = 0
  tiempo_cliente3(1) = 0
  tiempo_cliente3(2) = 0
  tiempo_cliente3(3) = 0
ENDIF
numero_operaciones = numero_operaciones + 1
IF numero_operaciones = 20
  numero_operaciones = 0
  tiempo_cliente3(3) = TimeEnteredModel ()
ELSEIF numero_operaciones = 1
  tiempo_cliente1(1) = TimeEnteredModel ()
ELSEIF numero_operaciones = 5
  tiempo_cliente1(2) = TimeEnteredModel ()
ELSEIF numero_operaciones = 15
  tiempo_cliente1(3) = TimeEnteredModel ()
ELSEIF numero_operaciones = 6
  tiempo_cliente2(1) = TimeEnteredModel ()
ELSEIF numero_operaciones = 8
  tiempo_cliente2(2) = TimeEnteredModel ()
ELSEIF numero_operaciones = 18
  tiempo_cliente2(3) = TimeEnteredModel ()
ELSEIF numero_operaciones = 9
  tiempo_cliente3(1) = TimeEnteredModel ()
ELSEIF numero_operaciones = 10
  tiempo_cliente3(2) = TimeEnteredModel ()
ENDIF

```

Figura 3.90: acción a la salida de la máquina unión modelo con heijunka y smed



## Capítulo 4.: COMPORTAMIENTO DE LOS MODELOS



En este capítulo se procederá a visualizar los distintos modelos que se han explicado en los Capítulos 2 y 3. En primer lugar, se mostrará la representación en 3D de los modelos sin técnica Lean y, posteriormente, se presentarán aquellas con técnica Lean asociada. Para finalizar, se realizará una comparación de ambos modelos a distintos tiempos.

## 1. TÉCNICA KANBAN

Para comenzar este capítulo, se mostrará cómo es la representación del modelo sin kanban, así como las características más importantes de éste (cómo se llenan los contenedores de la zona de acabado y se dejan de rellenar cuando tienen una cierta cantidad). Por el contrario, en los contenedores de fabricación se llenarán, debido a que la zona de fabricación nunca deja de fabricar.

A continuación, se mostrará el modelo en el que se ha implementado la técnica kanban, mostrándose cómo se fabrica solo cuando hay necesidad, es decir, cuando los kanban se encuentran en su buzón. Para finalizar, se realizará una comparación entre los dos modelos a distintos tiempos, para poder visualizar las diferencias que se generan al implantar una técnica lean.

### 1.1. MODELO SIN KANBAN

Para comenzar, se va a proceder a indicar los elementos del modelo 3D, dado que aunque ya se conocen cómo son en 2D, se van a volver a nombrar en el modelo 3D (Figura 4.1)



Figura 4.1: representación del modelo 3D sin kanban

En el instante inicial, se puede apreciar como en el modelo se poseen las cinco piezas en la entrada de cada una de las máquinas para simular el modelo estacionario, Figura 4.2.

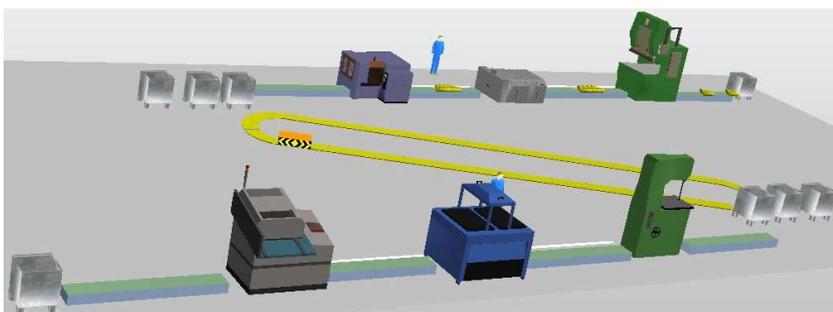


Figura 4.2: representación del modelo 3D sin kanban en el momento inicial

Una vez que se comienza a producir, se puede apreciar como en el contenedor **fabr1** se va a llenar con las 5 piezas (Figura 4.3); en ese momento, será necesario transportar las piezas a la zona de acabado (después de convertir las piezas en una única), Figura 4.4. Las piezas llegan al contenedor **alm1** ya se realiza en lote de 5 piezas, Figura 4.5. Como en la cinta de entrada al proceso de acabado no hay el mínimo de 2 piezas que se exige, éstas se colocarán en entrada (Figura 4.6).

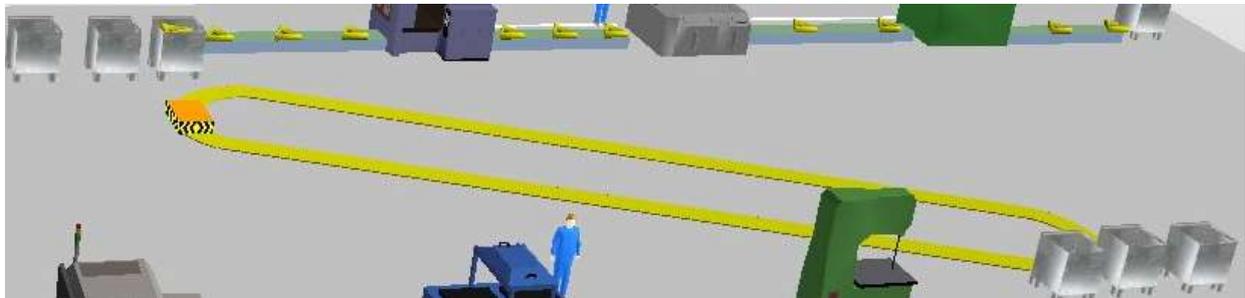


Figura 4.3: representación del modelo 3D sin kanban, llenado de **fabr1**

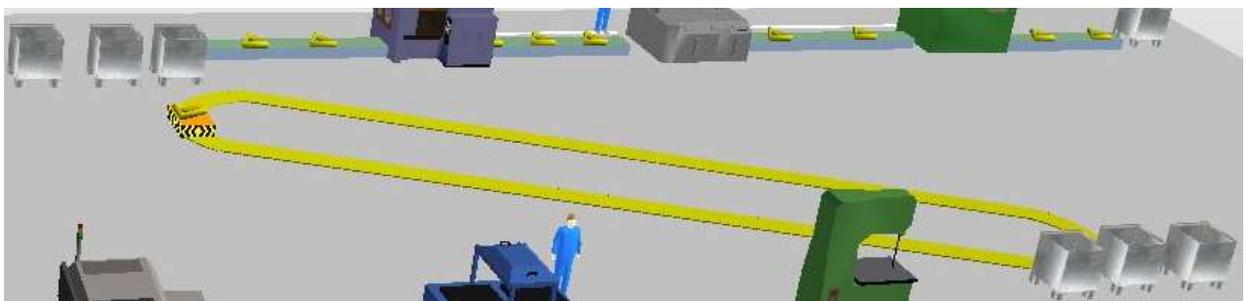


Figura 4.4: representación del modelo 3D sin kanban, transporte de la pieza desde zona de fabricación a la de acabado

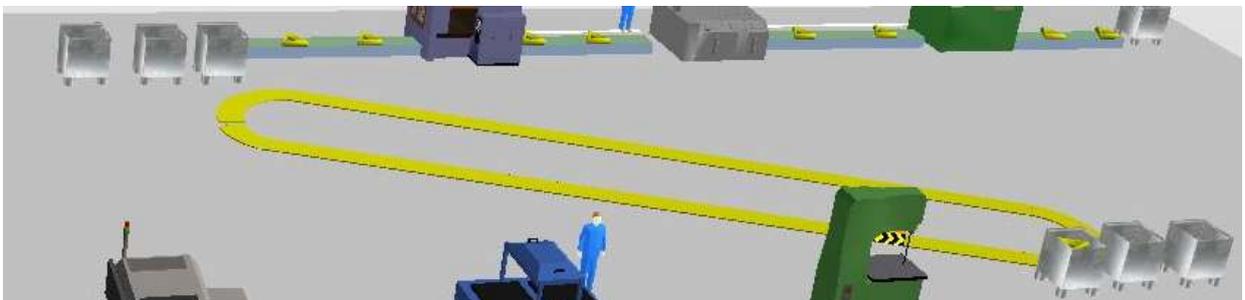


Figura 4.5: representación del modelo 3D sin kanban, almacenaje de las piezas en **alm1**

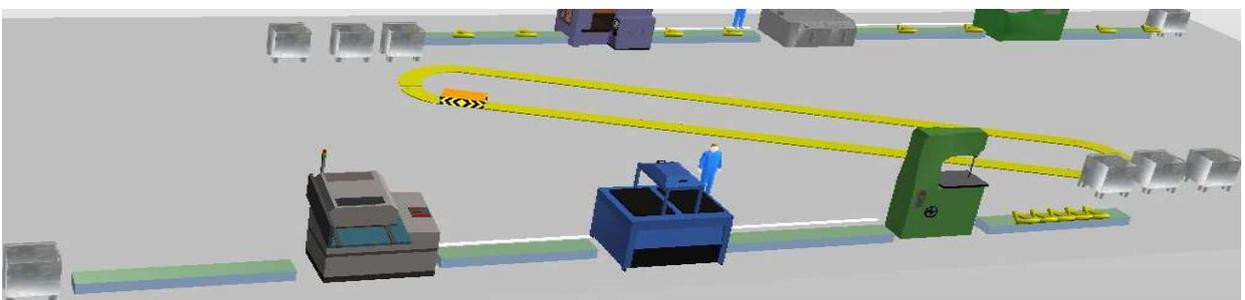


Figura 4.6: representación del modelo 3D sin kanban, colocación de las piezas en la entrada al proceso acabado

Se seguirán produciendo piezas en la zona de fabricación y, como en los contenedores de la zona de acabado no se disponen de 10 piezas en cada uno de ellos, se seguirán llevando piezas a esta zona aunque la cinta de entrada no haya necesidades, como se puede apreciar en la Figura 4.7

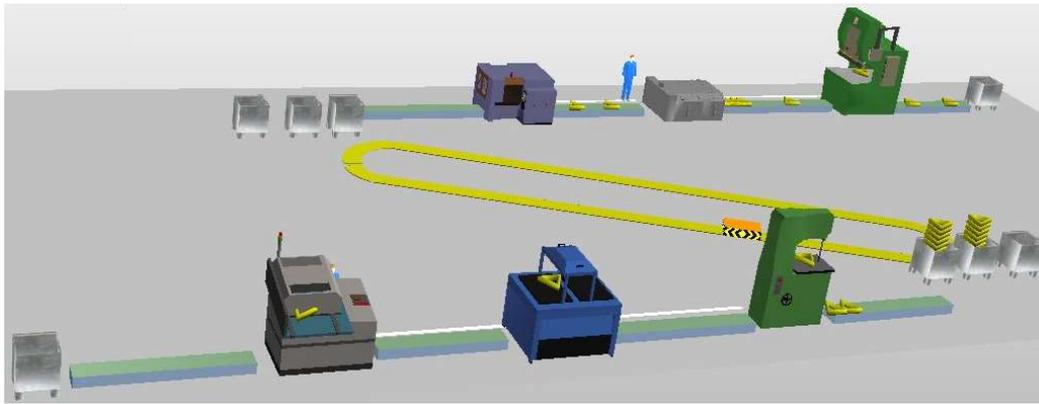


Figura 4.7: representación del modelo 3D sin kanban, llenado de los contenedores de acabado

En el momento en el que la cinta de entrada a la zona de acabado posea menos de 2 piezas, se sacarán 5 piezas de los contenedores y se depositarán en ésta, como se aprecian en la Figura 4.8.

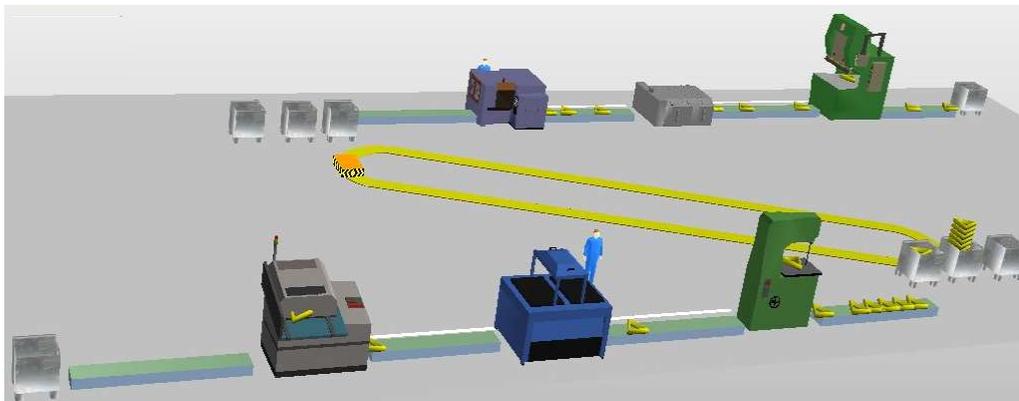


Figura 4.8: representación del modelo 3D sin kanban, llenado de la cinta de entrada a la zona de acabado

Cuando los tres contenedores de la zona de acabado posean 10 piezas, el **vehículo** llevará un lote más de piezas a la zona de acabado, esperando a que uno de los contenedores se quede vacío para depositarlas, como se aprecia en la Figura 4.9.

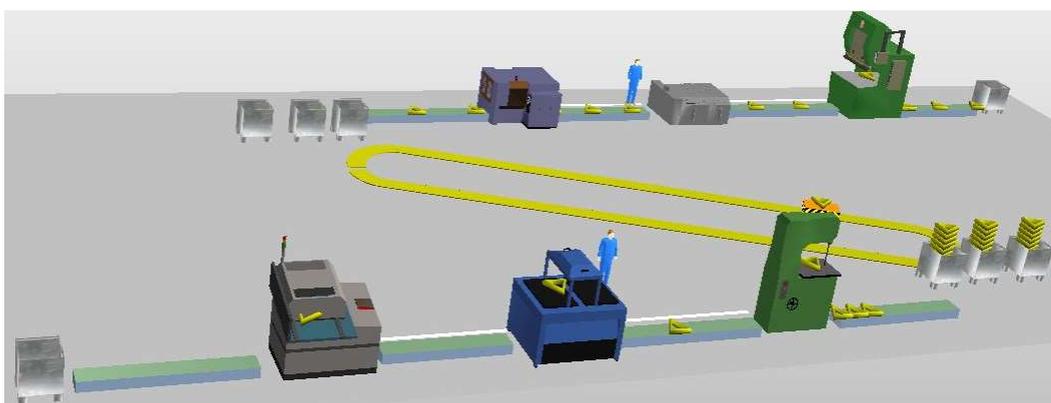


Figura 4.9: representación del modelo 3D sin kanban, transporte de 5 piezas estando los contenedores de acabado llenos

Se seguirán generando piezas, aunque no haya necesidades en la zona de acabado, haciendo que se acumulen en los contenedores de la zona de fabricación. Cuando se necesiten piezas en la cinta de **entrada\_final**, se sacarán del contenedor de acabado e inmediatamente éste se llenará con las 5 piezas que hay en el **vehículo**, Figura 4.10.

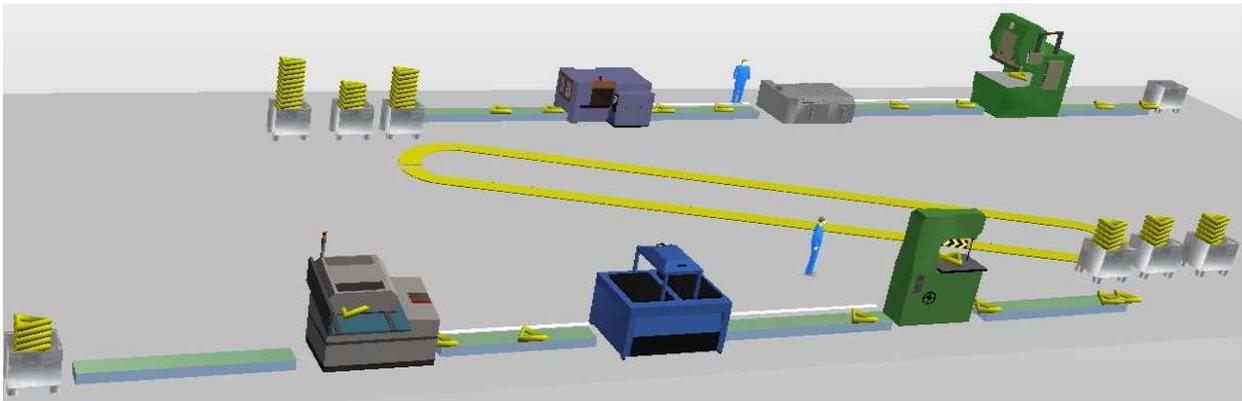


Figura 4.10: representación del modelo 3D sin kanban, llenado de los contenedores de la zona de acabado

Además, se puede apreciar cómo se produce la salida de las piezas del proceso cuando en el contenedor de `almacen_final` hay 10 piezas, como se aprecia en la Figura 4.11.

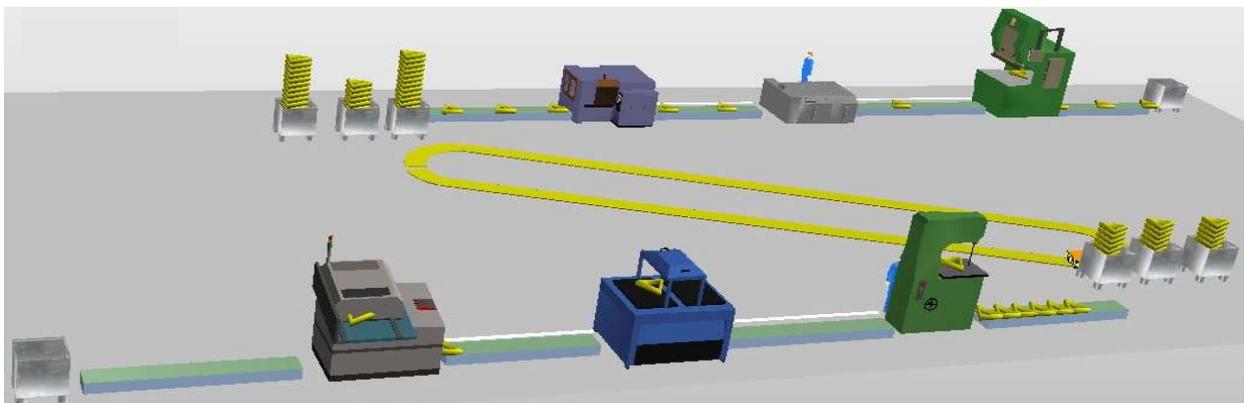


Figura 4.11: representación del modelo 3D sin kanban, salida de piezas del proceso

## 1.2. MODELO CON KANBAN

Antes de comenzar a visualizar el comportamiento del modelo, se procederá como en el caso anterior a indicar los elementos que lo forman, como se aprecia en la Figura 4.12.

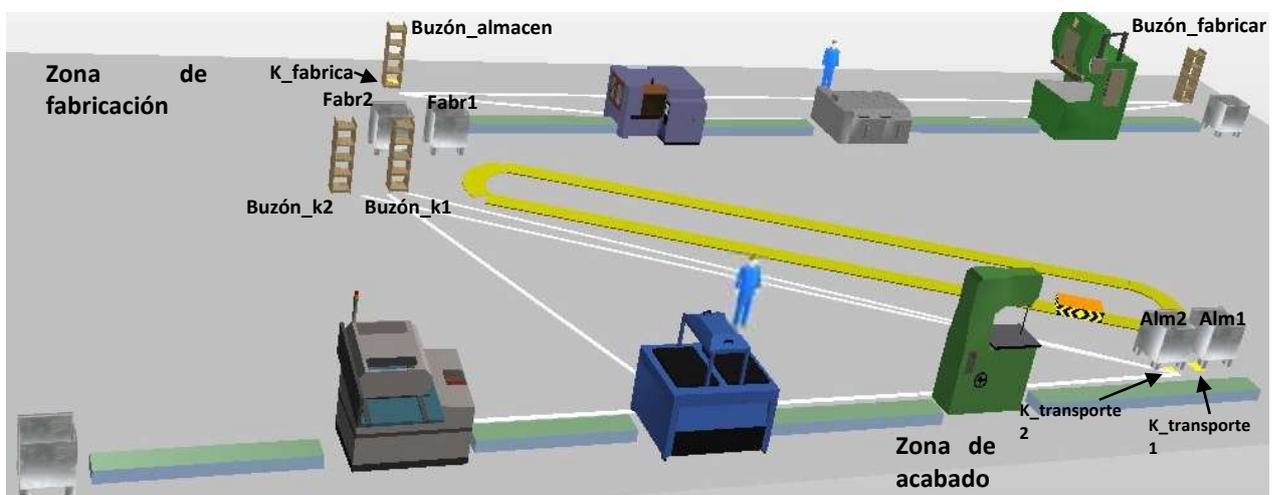


Figura 4.12: representación del modelo 3D con kanban

En el momento inicial, no se poseen de piezas en la zona de acabado ni en la de fabricación; por tanto, los operarios deben trasladar los kanban hasta su buzón correspondiente por los caminos definidos, Figura 4.13. En el momento en el que se deposite el  $k_{fabrica}$  en el  $buzon_fabricar$ , llegarán las 5 piezas a la entrada de cada una de las máquinas, debido a que se trata del instante inicial. El operario habrá tenido que ir a por el  $k_{transporte2}$ , dado que en el contenedor  $alm2$  no hay piezas, como muestra la Figura 4.14.

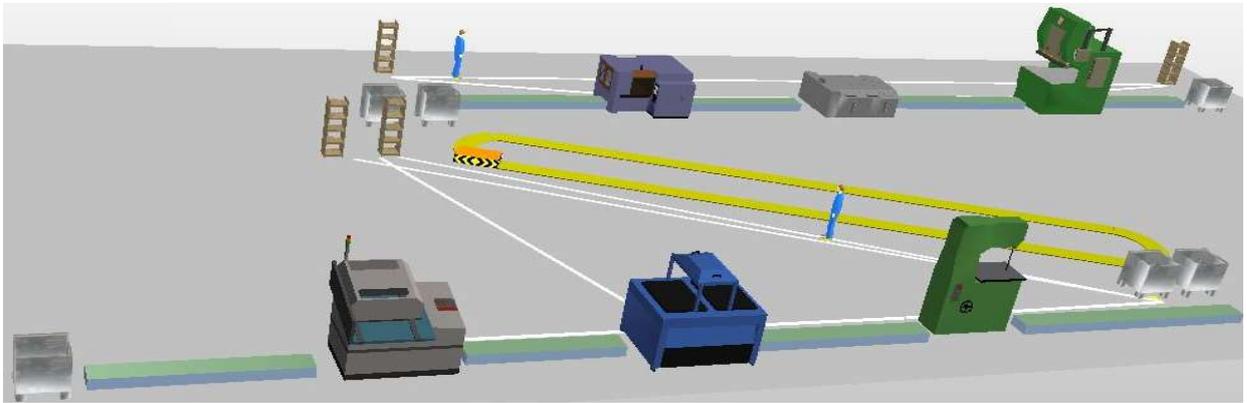


Figura 4.13: representación del modelo 3D con kanban, transporte de kanban

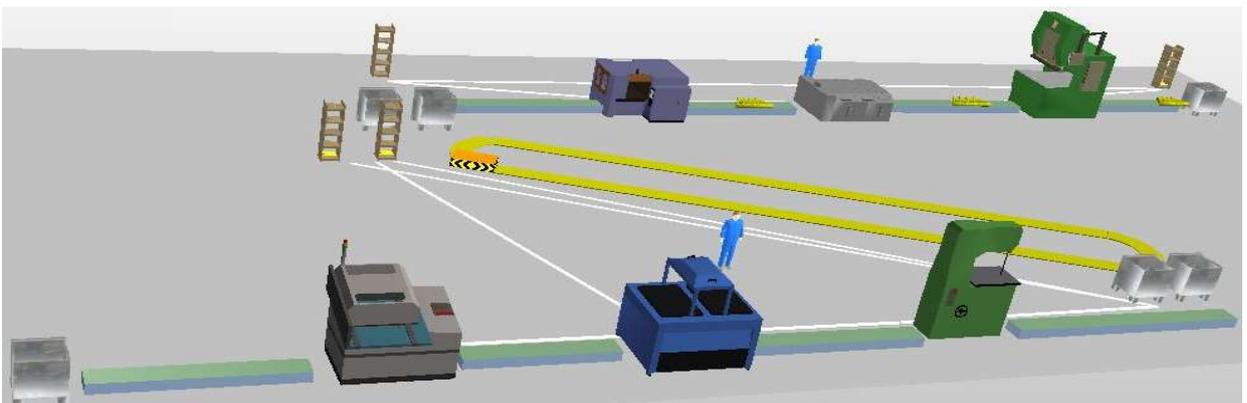


Figura 4.14: representación del modelo 3D con kanban, comienzo de fabricación en el momento inicial

Cuando en uno de los contenedores de fabricación haya 5 piezas, Figura 4.15, se unirán el  $k_{transporte}$  con las 5 piezas, Figura 4.16, para trasladarlas a la zona de acabado. Tanto kanban como piezas se depositarán en el contenedor correspondiente, Figura 4.17. Si no se poseen piezas en la cinta de entrada, las piezas que se han depositado en el contenedor irán a dicha cinta, haciendo que el operario tenga que ir a por el kanban, Figura 4.18.

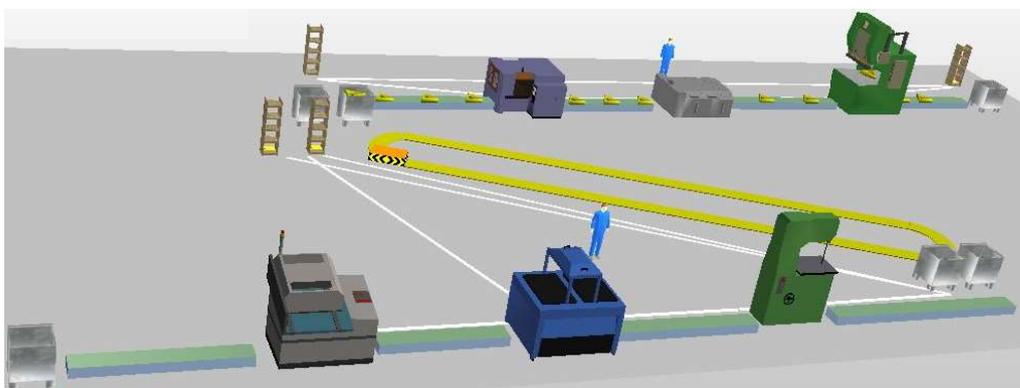


Figura 4.15: representación del modelo 3D con kanban, contenedor fabr1 con 5 piezas

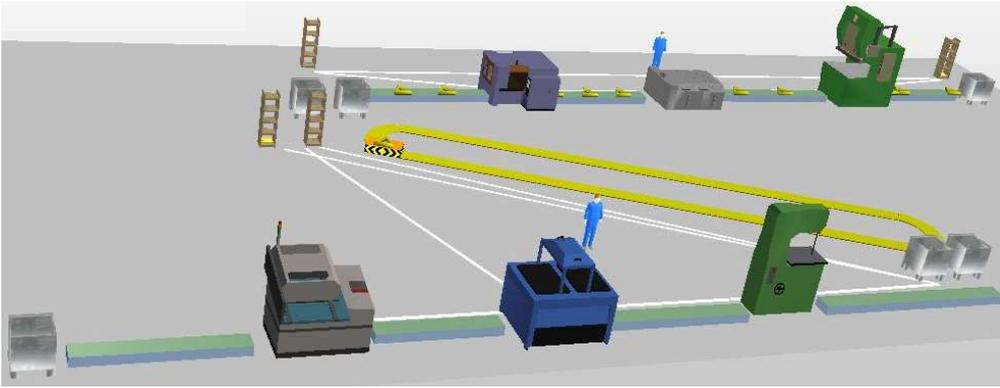


Figura 4.16: representación del modelo 3D con kanban, unión de las 5 piezas con k\_transporte1

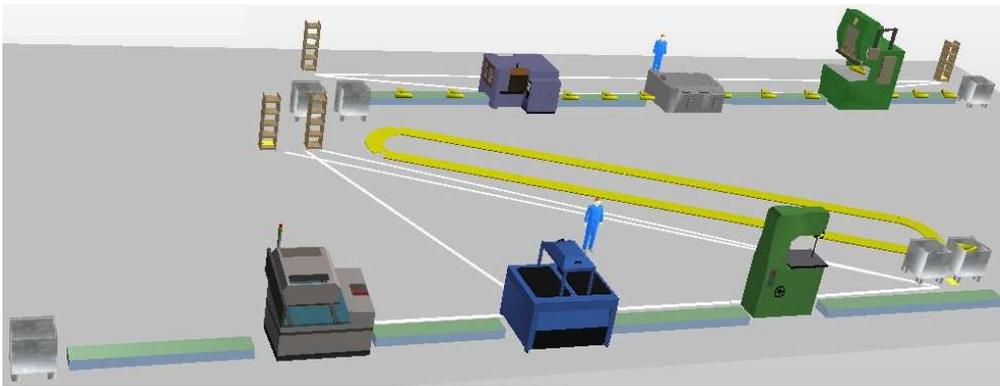


Figura 4.17: representación del modelo 3D con kanban, llenado del contenedor alm1 junto con k\_transporte1

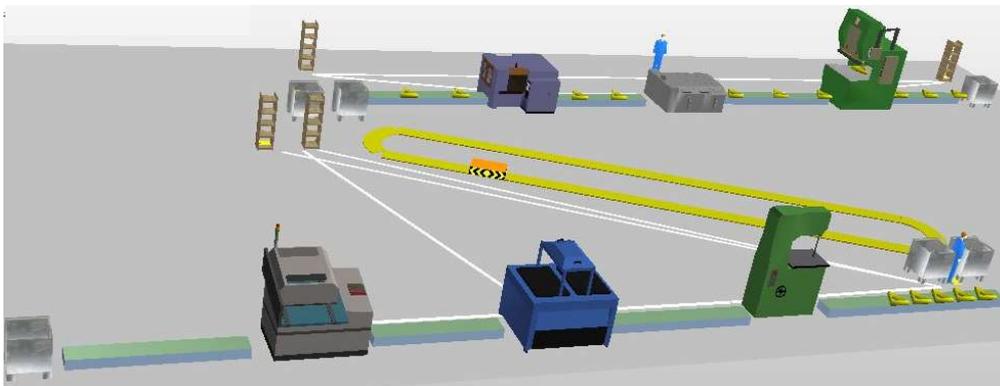


Figura 4.18: representación del modelo 3D con kanban, llenado del cinta entrada y recogida k\_transporte1

Este proceso se repetirá hasta que haya 5 piezas en alm1 y alm2. En este momento no es necesario traer piezas de la zona de fabricación, estando los dos k\_transporte en sus contenedores correspondientes, estando los buzones de transporte vacíos, como se aprecia en la Figura 4.19.

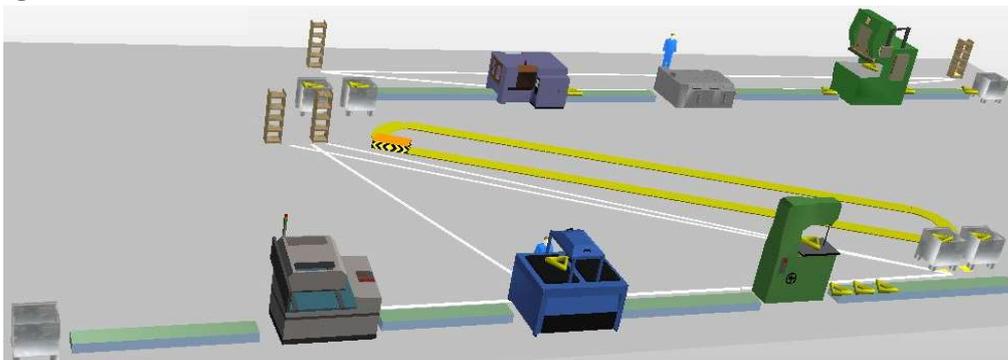


Figura 4.19: representación del modelo 3D con kanban, sin necesidades de transporte pero sí de fabricación

Se seguirán fabricando piezas hasta que haya 5 piezas en la entrada de cada una de las máquinas. En ese momento, será necesario que el operario recoja el *k\_fabrica* y lo lleve al *buzon\_almacen*, Figura 4.20, provocando que se deje de fabricar, dado que el *k\_fabrica* no se encuentra en el *buzon\_fabricar*. En la Figura 4.21 se puede apreciar cómo el *k\_fabrica* se encuentra en el *buzon\_almacen* y las máquinas han dejado de fabricar, hasta que en la zona de acabado haya necesidades, sacando piezas de los contenedores de fabricación.

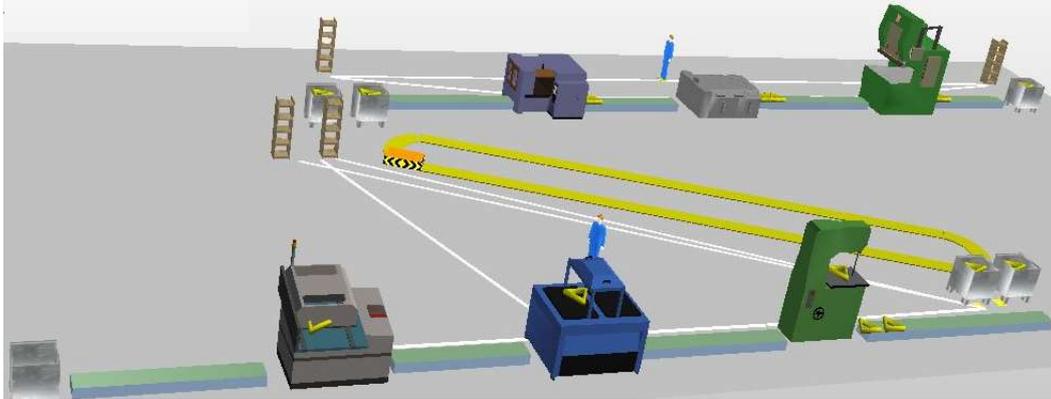


Figura 4.20: representación del modelo 3D con kanban, *k\_fabrica* hacia *buzon\_almacen*

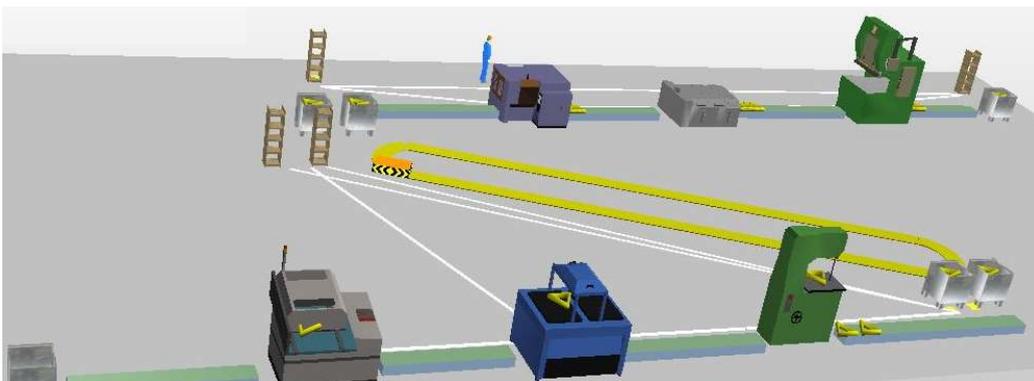


Figura 4.21: representación del modelo 3D con kanban, sin necesidades de fabricación ni de transporte

Como se ha comentado a la hora de explicar los modelos, los distintos paths conectan todas las máquinas para que los movimientos de los operarios estén definidos. De esta forma, en la Figura 4.22 se puede apreciar como el *operario1*, después de dejar el *k\_transporte*, se desplazará por el path para realizar una tarea de mantenimiento en su zona.

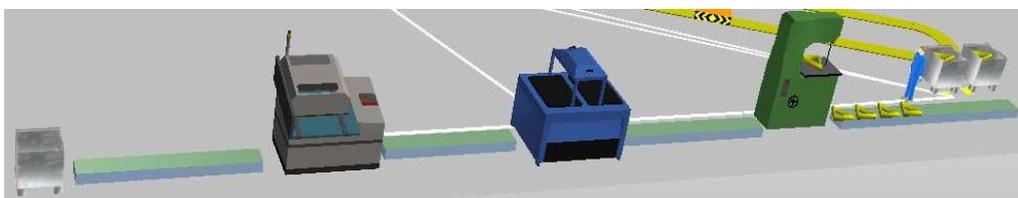


Figura 4.22: representación del modelo 3D con kanban, tareas mantenimiento con movimiento entre paths

### 1.3. COMPARACIÓN DE LOS MODELOS

Una vez que se ha podido visualizar cómo es el funcionamiento de los modelos en 3D, se procederá a hacer una pequeña comparación de los dos modelos en instantes concretos. Además de comparar visualmente cómo son los modelos, también se van a dar cifras de las piezas que han entrado y salido del modelo, Tabla 4.1, debido a que visualmente todas las piezas que se generan en el modelo no se pueden visualizar, pues *salida\_modelo* las saca del modelo.

Tiempo	Con kanban		Sin kanban	
	Piezas entrada	Piezas salida	Piezas entrada	Piezas salida
1000	55	7	106	10
2000	65	20	216	25
2500	73	27	266	32

Tabla 4.1: comparación del número de piezas en modelo kanban a diferentes tiempos

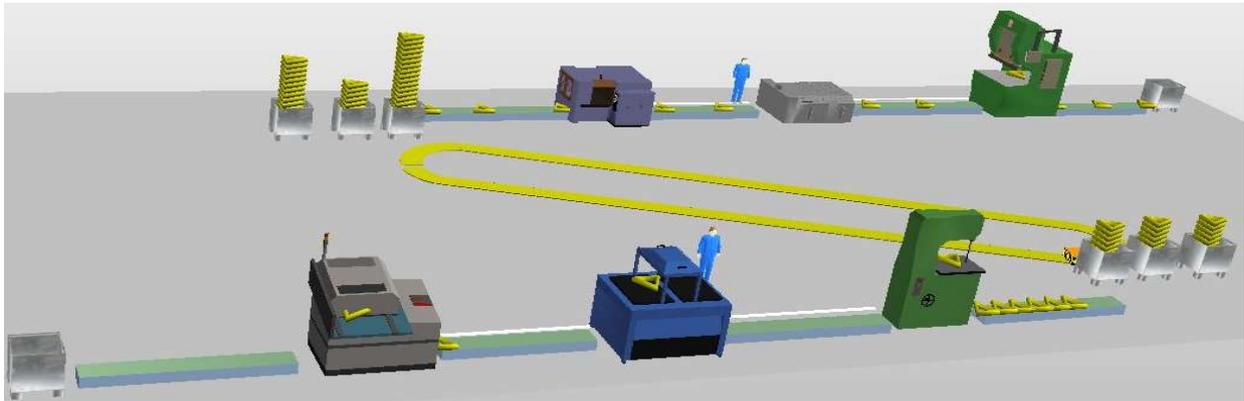


Figura 4.23: representación del modelo 3D sin kanban en el instante 1000

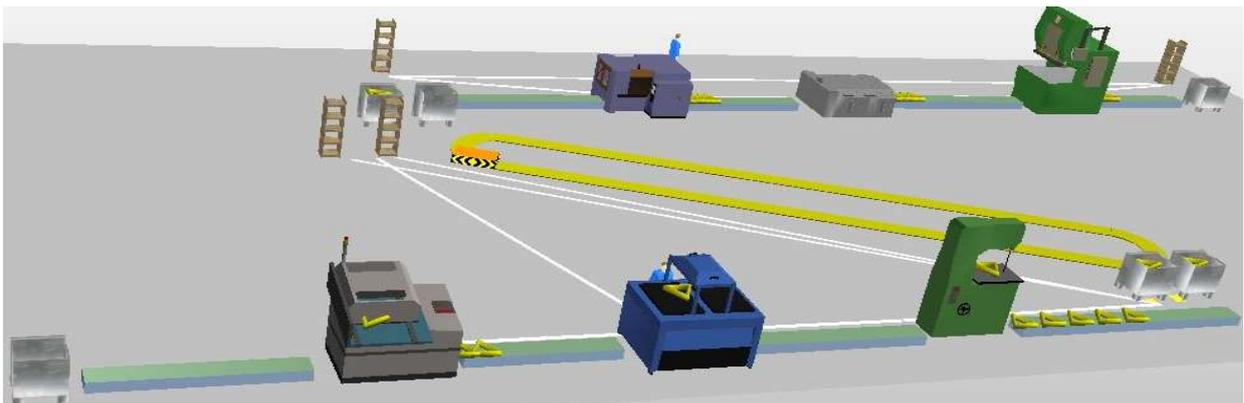


Figura 4.24: representación del modelo 3D con kanban en el instante 1000

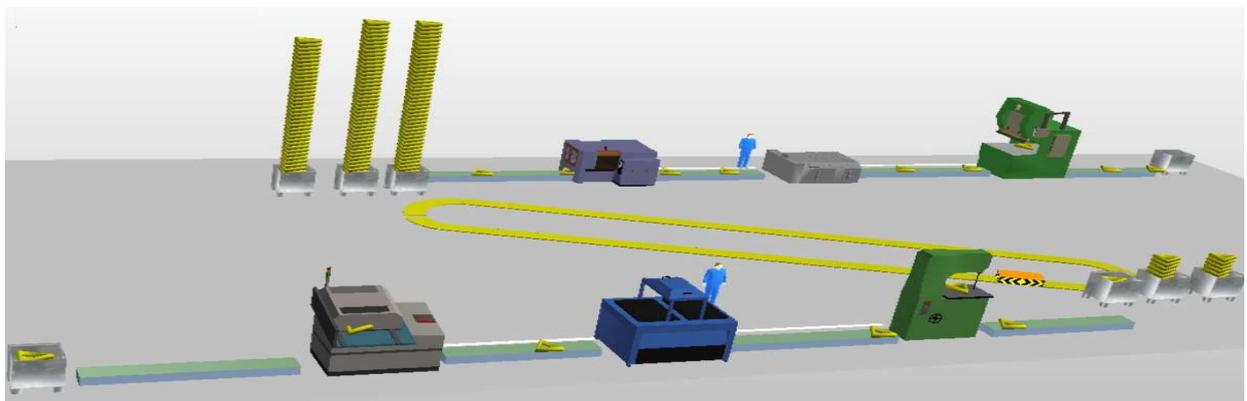


Figura 4.25: representación del modelo 3D sin kanban en el instante 2000

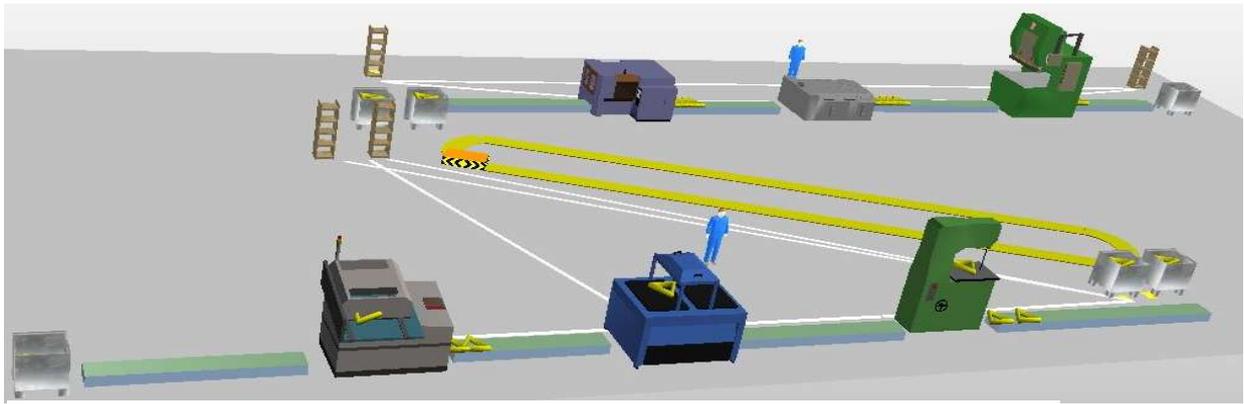


Figura 4.26: representación del modelo 3D con kanban en el instante 2000

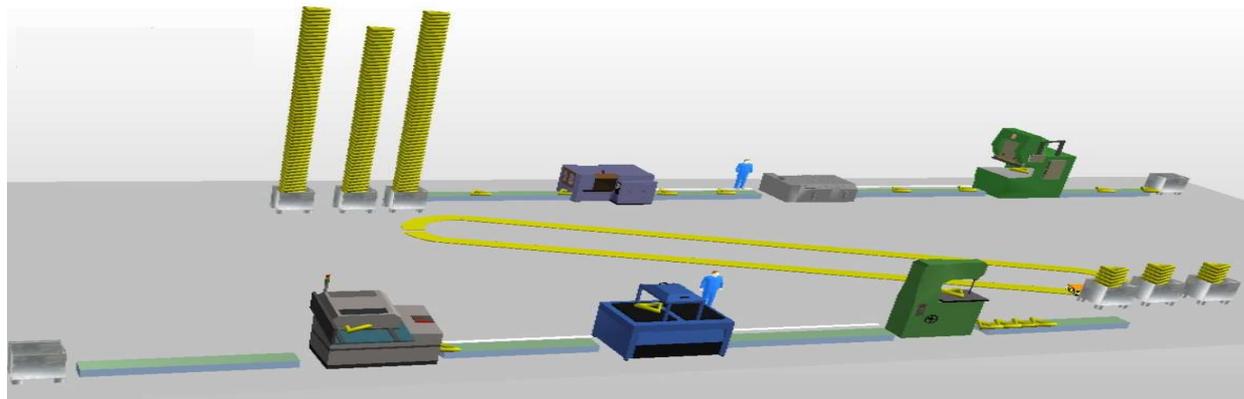


Figura 4.27: representación del modelo 3D sin kanban en el instante 2500

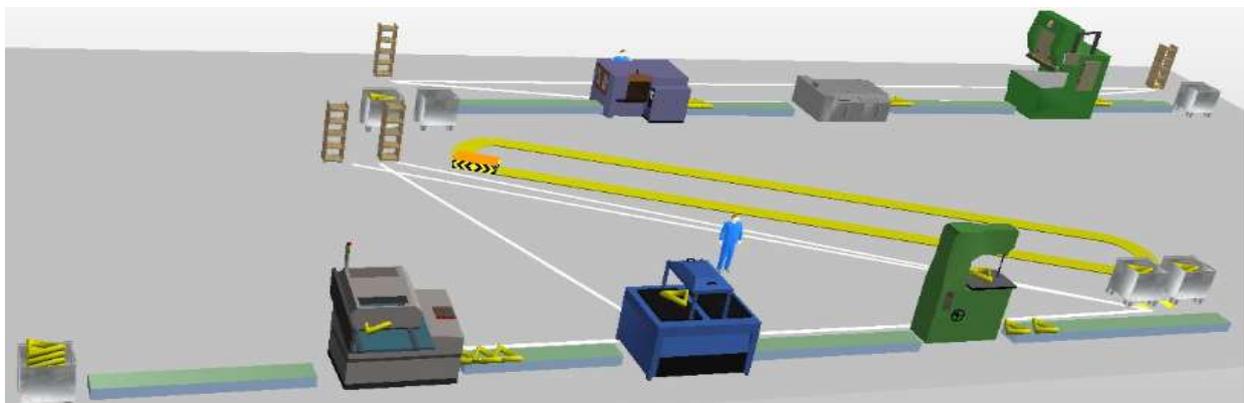


Figura 4.28: representación del modelo 3D con kanban en el instante 2500

## 2. TEORÍA DE LAS LIMITACIONES

En este apartado se procederá a visualizar los modelos asociados a la teoría de las limitaciones. En primer lugar, se apreciará el modelo que posee cuello de botella, pudiéndose apreciar cómo se sacan las piezas de la *cinta1\_2* para depositarlas en el contenedor. Como técnica Lean, se creará un elemento capacitario, aumentando en una unidad la *maq2*, y parando la *maq1* así como la entrada de piezas cuando en la *cinta1\_2* y la *entrada\_a\_maquina1* ya haya las suficientes piezas. Para finalizar, se compararán el modelo con cuello de botella con el modelo con elemento capacitario en diferentes instantes, para conocer cómo se disminuye el nivel de stock al no haber cuello de botella.

## 2.1. MODELO CON CUELLO DE BOTELLA

La visualización del modelo que posee un cuello de botella se puede apreciar en la Figura 4.29, donde la característica más relevante de este modelo es la salida de piezas hacia el contenedor *sal\_cuello*. Como se puede apreciar existen diferentes paths para moverse entre las máquinas, así como entre la *maq2* y el contenedor *sal\_cuello*

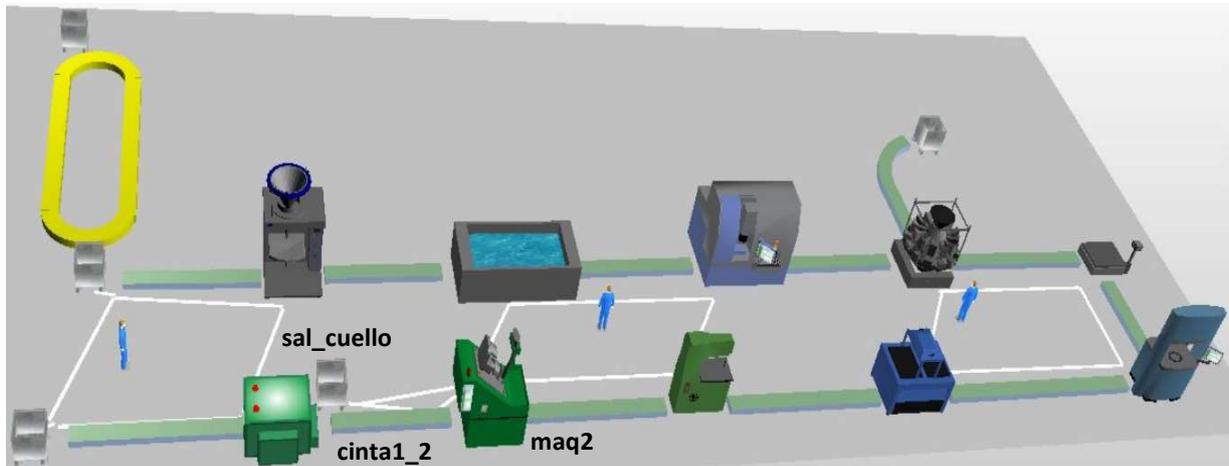


Figura 4.29: representación del modelo 3D con cuello de botella

En la Figura 4.30 se puede apreciar cómo se produce la entrada de piezas en el modelo, siendo realizado por *op\_ent\_sal\_1\_10* mediante el path correspondiente, como se puede apreciar el operario arrastra la *pieza1* por el path. Esta operación se realizará siempre que no sea necesario sacar piezas del modelo. Además, se puede observar cómo se empiezan a acumular piezas en la *cinta1\_2* debido al elevado tiempo de ciclo de la *maq2*.

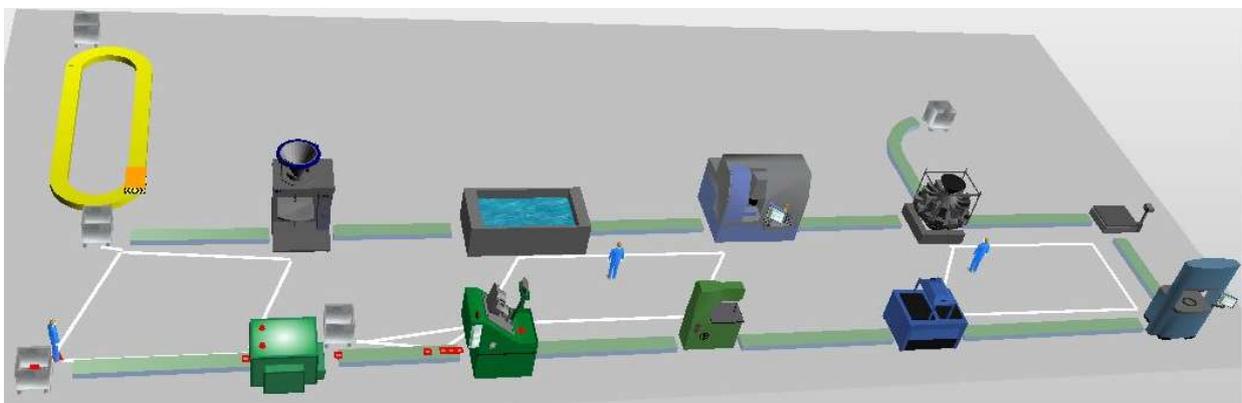


Figura 4.30: entrada piezas al modelo 3D con cuello de botella

En el momento en el que haya más de 5 piezas en la *cinta1\_2*, en este caso, cuando hay 6, el *op\_2\_3\_8\_9* se dirigirá hacia la *cinta1\_2*, Figura 4.31 (como el operario se encontraba en su posición de reposo, no usará ningún path hasta llegar a la *cinta1\_2*). Cuando llegue el *op\_2\_3\_8\_9* a la cinta cogerá una *pieza1*, Figura 4.32, y la trasladará al contenedor *sal\_cuello* mediante el path correspondiente, Figura 4.33. Si en la *cinta1\_2* sigue habiendo más de 5 piezas, el operario se volverá a trasladar por dicho path, Figura 4.34 para recoger más piezas. En el caso de que no sea necesario sacar más piezas y el operario sea requerido en otras máquinas, éste irá por el path mostrado en la Figura 4.35.

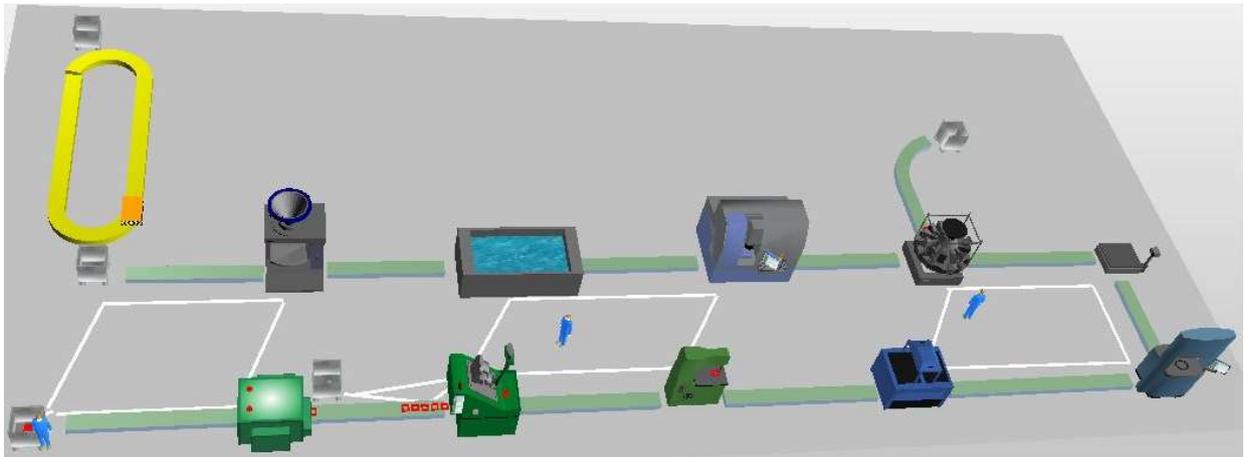


Figura 4.31: operario dirigiéndose a sacar piezas cinta1\_2 modelo 3D con cuello de botella

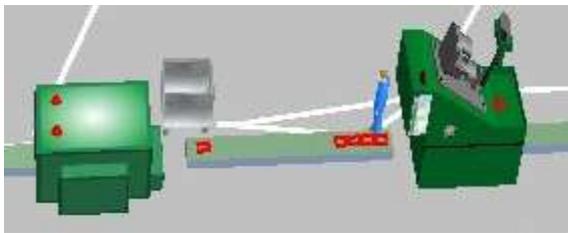


Figura 4.32: operario sacando piezas de la cinta1\_2 cuello de botella, modelo 3D cuello de botella

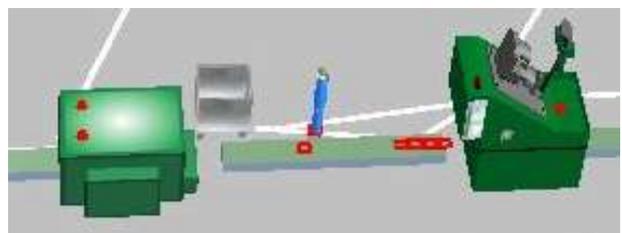


Figura 4.33: transporte de las piezas por el path cuello\_cont hacia el contenedor sal\_cuello, modelo 3D cuello de botella

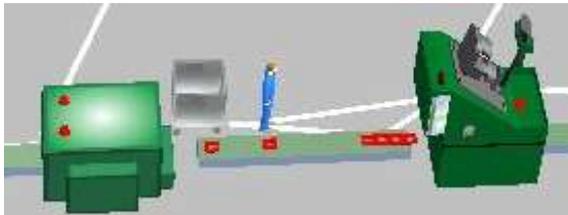


Figura 4.34: operario trasladándose a la cinta1\_2 desde el cuello de botella, modelo 3D cuello de botella

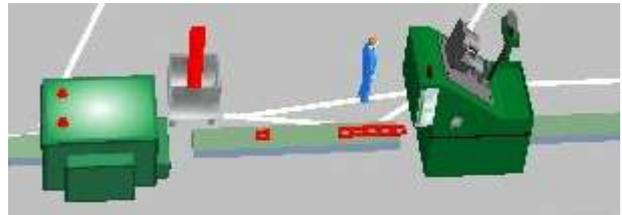


Figura 4.35: operario trasladándose otra máquina desde el cuello de botella, modelo 3D cuello de botella

Cuando en la cinta *salida\_proceso* hay más de 8 piezas, el *op\_ent\_sal\_1\_10* se dirigirá a la cinta para sacar 8 piezas. Como se aprecia en la Figura 4.36, el operario sacará de una en una las 8 piezas de la cinta trasladándolas a través del path correspondiente. Además, en esta imagen se puede apreciar cómo se empiezan a acumular piezas tanto en la *cinta1\_2* como en el contenedor *sal\_cuello*. Cuando se han trasladado las 8 piezas, se podrán visualizar en el *contenedor\_salida*.

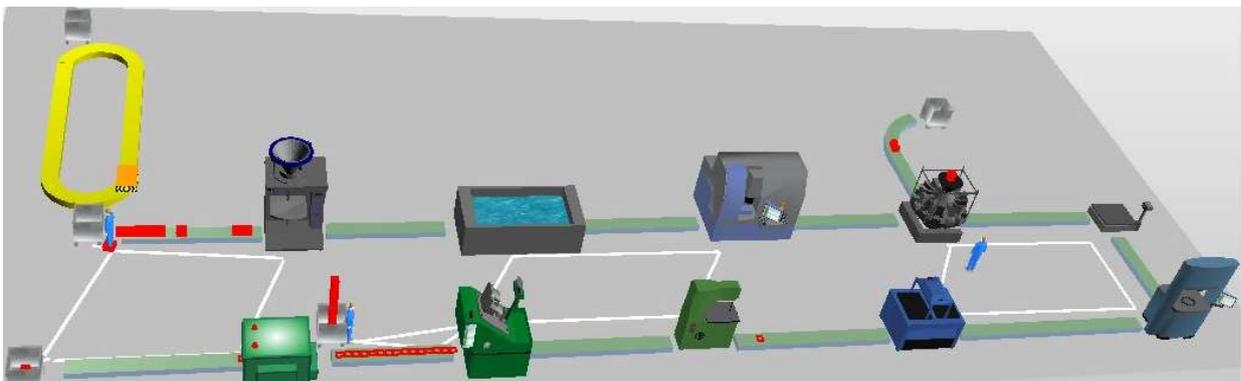


Figura 4.36: operario sacando piezas de la cinta salida\_proceso, modelo 3D cuello de botella

## 2.2. MODELO CON ELEMENTO CAPACITARIO

Como se ha podido visualizar en el modelo con cuello de botella, se produce una gran cantidad de stock intermedio, lo cual no es aconsejable en ningún proceso productivo. Para evitar esta acumulación y para satisfacer la demanda del cliente, se ha aumentado en una unidad la máquina que genera el cuello de botella, la *maq2*. El modelo se puede apreciar en la Figura 4.37.

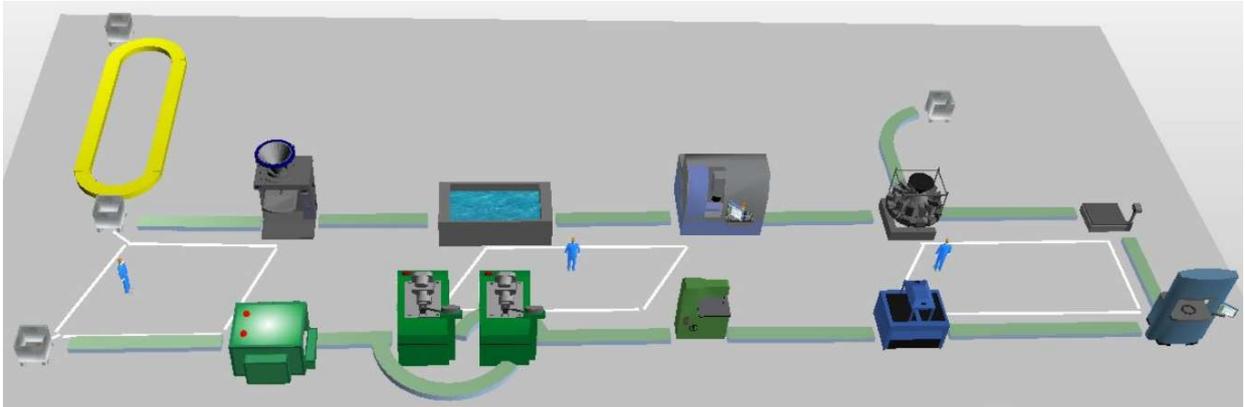


Figura 4.37: representación del modelo 3D capacitario

El proceso de entrada y salida de piezas del modelo es el mismo que en el caso con cuello de botella, realizando ambas actividades el *op\_ent\_sal\_1\_10*. Cabe destacar que el *op\_ent\_sal\_1\_10* introducirá piezas en el modelo hasta que la cantidad de piezas que hay en el conjunto de *entr\_a\_maq1*, *maq1* y *cinta1\_2* sea de 13 piezas. De esta forma, se puede apreciar en la Figura 4.38 como se realiza la entrada de piezas en la cinta *entr\_a\_maq1*. En el momento en el que la *cinta1\_2* tenga 8 piezas, la *maq1* dejará de producir piezas, como se puede apreciar en la Figura 4.39, el *op\_ent\_sal\_1\_10* seguirá introduciendo piezas en la *entr\_a\_maq1* aunque ésta se encuentre parada. En el momento en el que haya 5 piezas en la *entr\_a\_maq1*, el operario dejará de introducir piezas en el proceso, desplazándose a realizar otras actividades, o si no debe realizar ninguna actividad se dirigirá a su posición de reposo, como se aprecia en la Figura 4.40.

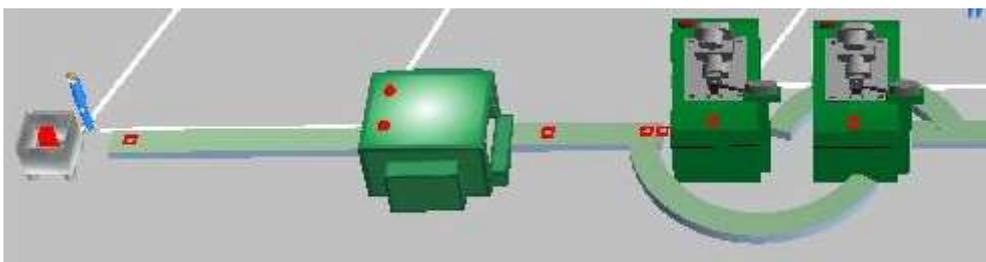


Figura 4.38: entrada de piezas en el modelo 3D capacitario

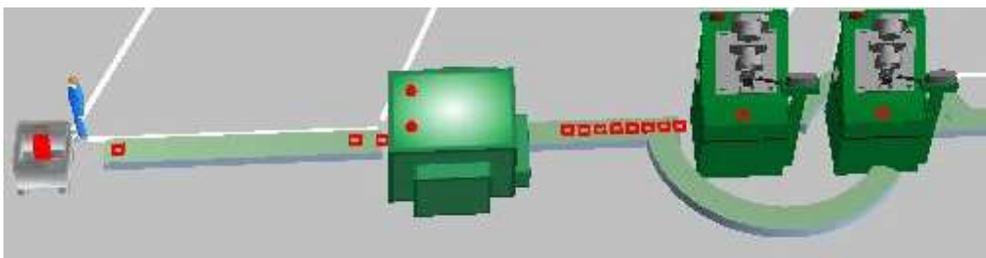


Figura 4.39: parada de la *maq1* con stock necesario en *cinta1\_2* en el modelo 3D capacitario

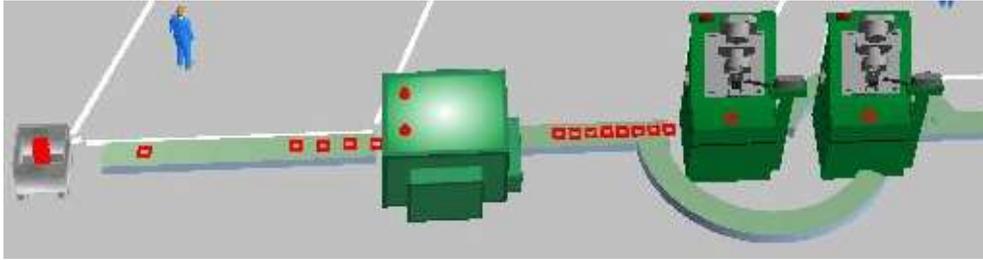


Figura 4.40: finalización de la actividad de introducir piezas en el modelo 3D capacitario

### 2.3. COMPARACIÓN DE LOS MODELOS

Una vez que se conoce cómo es la visualización de ambos modelos, se procederá a compararlos en determinados instantes, los cuales se pueden apreciar en la Tabla 4.2, apreciándose la cantidad de piezas que se acumulan en el contenedor *sal\_cuello*.

Tiempo	Cuello de botella			Con elemento capacitario	
	Piezas entrada	Piezas salida	Sal_cuello	Piezas entrada	Piezas salida
1000	59	12	41	44	33
2000	103	28	76	59	48

Tabla 4.2: comparación del número de piezas en modelo cuello de botella a diferentes tiempos

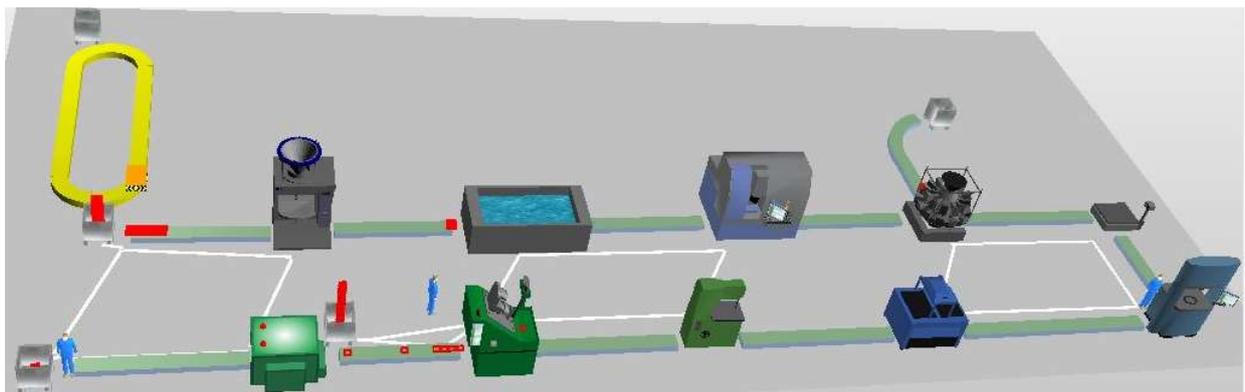


Figura 4.41: visualización del modelo 3D cuello de botella en el instante 1000

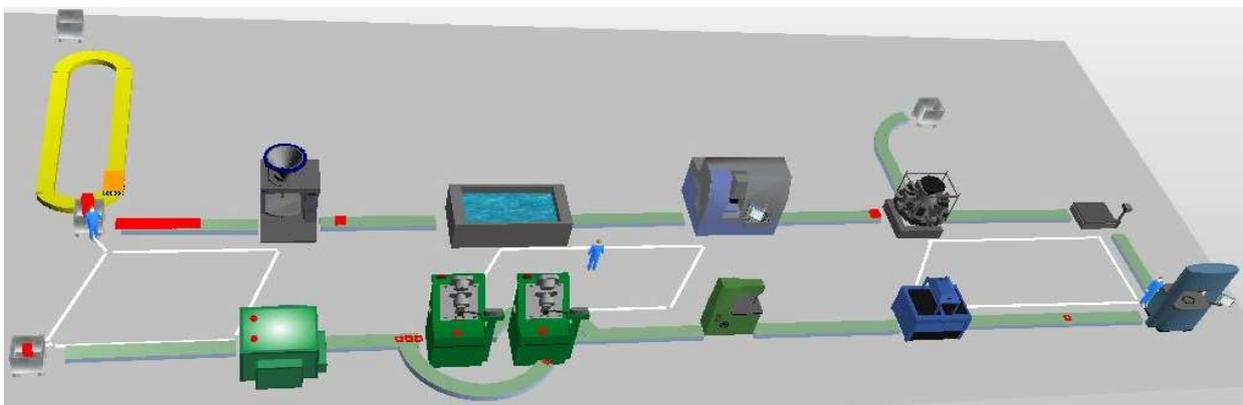


Figura 4.42: visualización del modelo 3D con elemento capacitario en el instante 1000

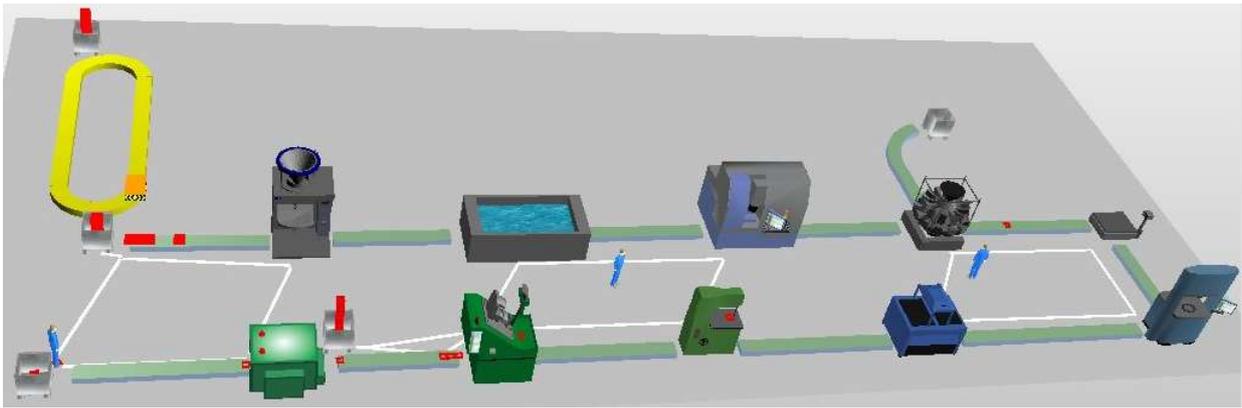


Figura 4.43: visualización del modelo 3D cuello de botella en el instante 2000

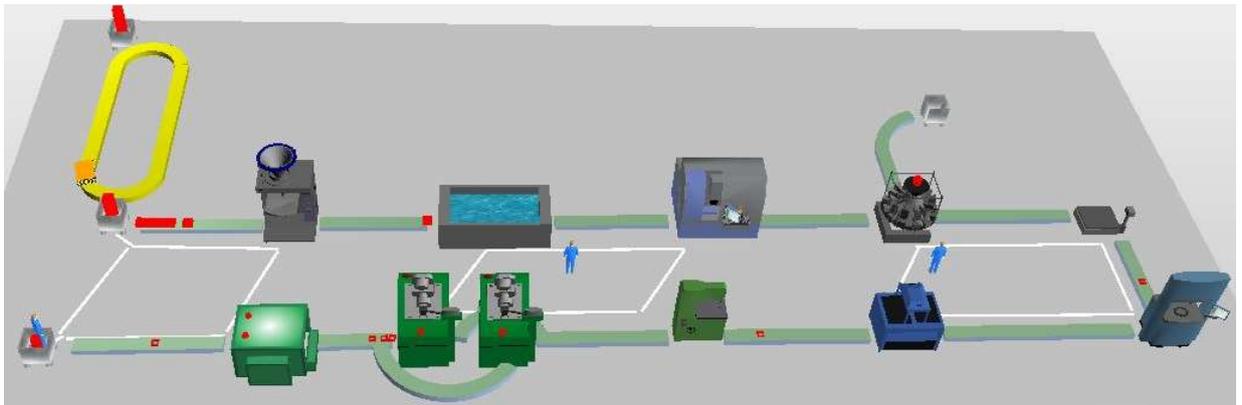


Figura 4.44: visualización del modelo 3D con elemento capacitario en el instante 2000

### 3. TÉCNICA POKA YOKE

Como se ha explicado en el Capítulo 3, se poseen tres modelos asociados a la utilización de la técnica poka yoke. A continuación, se va a proceder a visualizar las características generales del modelo en el modelo sin poka yoke para, posteriormente, apreciar cómo mejora el modelo al aplicar poka yokes de advertencia y de fiabilidad comparando los modelos en determinados instantes.

#### 3.1. MODELO SIN POKA YOKE, CON POKA YOKE DE ADVERTENCIA Y CON POKA YOKE DE FIABILIDAD

Para comenzar, se visualizará el modelo sin poka yoke, Figura 4.45, en donde se aprecian las distintas máquinas indicadas a la hora de describir el modelo, dadas por sus iconos así como la acción de ensamblaje, dada por el operario. Además, se puede apreciar la salida de piezas defectuosas y la salida de piezas correctas al finalizar cada una de las zonas.

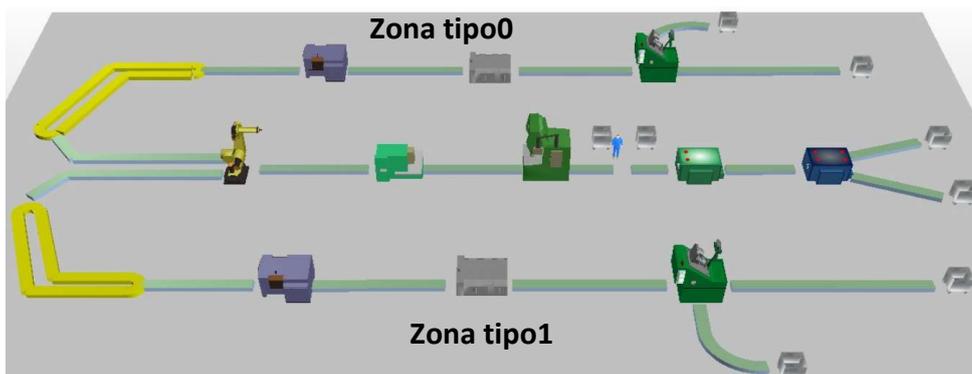


Figura 4.45: representación del modelo 3D sin poka yoke

A la hora de describir el modelo se han hecho una serie de consideraciones referentes a la visualización de las piezas acorde a su ensamblaje o si poseen defecto de calidad. A continuación, se van a describir distintos casos en los que se cambia la visualización de las piezas.

En primer lugar, se indicará la visualización cuando una pieza no posee la calidad deseada. En la Figura 4.46 se puede apreciar la entrada de las piezas a la **maquina2**, pudiéndose apreciar cómo éstas poseen el icono del **tipo0**. A la salida de la **maquina2**, Figura 4.47, la segunda pieza no posee la calidad deseada, por lo que su icono ha cambiado al de pieza defectuosa por calidad.

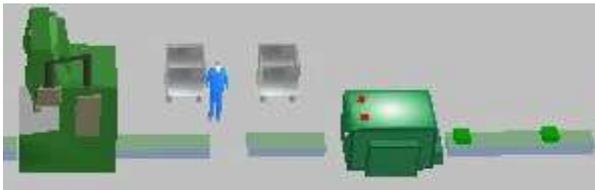


Figura 4.46: representación del modelo 3D sin poka yoke, pieza antes de defecto de calidad

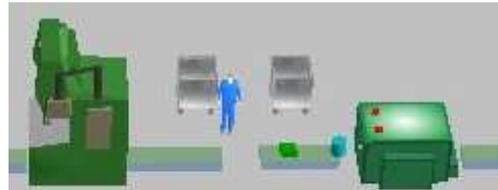


Figura 4.47: representación del modelo 3D sin poka yoke, pieza con defecto de calidad

Una vez que se ha explicado cómo se procede con el defecto de calidad, se explicará cómo se procede con la actividad de ensamblaje. En la Figura 4.48 se aprecia cómo el ensamblaje se ha realizado correctamente, dado que se cogen las dos piezas del mismo tipo; se puede observar cómo es el conjunto que se forma, siendo igual que el **tipo0**, Figura 4.49.



Figura 4.48: representación del modelo 3D sin poka yoke, ensamblaje tipo0

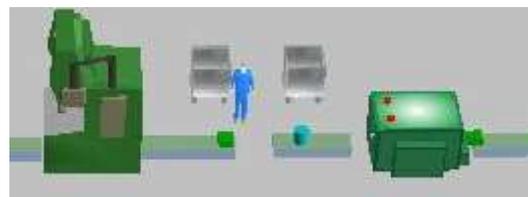


Figura 4.49: representación del modelo 3D sin poka yoke, conjunto ensamblaje tipo0

Un caso que se puede dar es el producido al ensamblar piezas con defectos de calidad, Figura 4.50. En este caso, el defecto de prioridad prevalece sobre el ensamblaje, manteniendo el mismo icono que tenía antes de realizar el ensamblaje, Figura 4.51.



Figura 4.50: representación del modelo 3D sin poka yoke, pieza defecto calidad antes de ensamblaje



Figura 4.51: representación del modelo 3D sin poka yoke, pieza defecto calidad después de ensamblaje

Igual que se realizase para el **tipo0**, en el caso de ensamblaje correcto en el **tipo1**, Figura 4.52, el conjunto generado será como el **tipo1**, Figura 4.53.



Figura 4.52: representación del modelo 3D sin poka yoke, ensamblaje tipo1

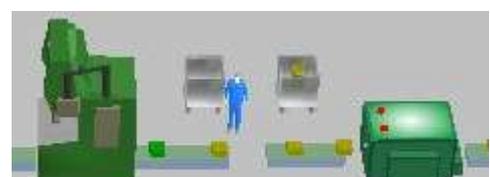


Figura 4.53: representación del modelo 3D sin poka yoke, conjunto ensamblaje tipo1

El último caso que puede producirse en el modelo es el ensamblaje incorrecto: se unen piezas **tipo0** con **ens1**, es decir, **tipo** sea distinto de **tipo\_acople**, como se puede visualizar en la Figura 4.54. Una vez que se genera el conjunto, la visualización de éste será la mostrada en la Figura 4.55.

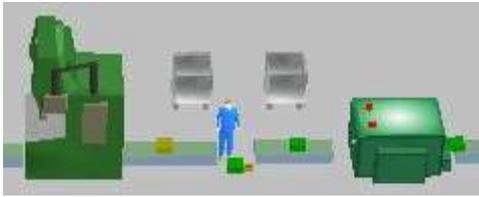


Figura 4.54: representación del modelo 3D sin poka yoke, ensamblaje incorrecto

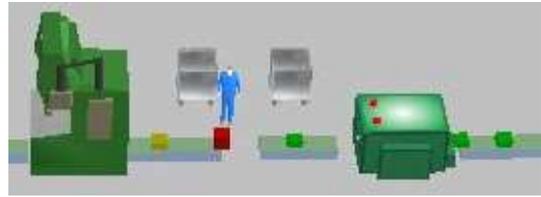


Figura 4.55: representación del modelo 3D sin poka yoke, representación del conjunto incorrecto

Otra acción que hay que controlar en el modelo, es la colocación de las piezas en una zona u otra dependiendo del tipo. Como se ha podido apreciar en la Figura 4.45, se poseen dos zonas: la zona superior para el **tipo0** y la zona inferior para el **tipo1**. La acción de colocar las piezas en una u otra zona la lleva a cabo la **maquina5**, situando los conjuntos **tipo0** en la parte superior, Figura 4.56. Se pueden generar errores en la colocación, colocando el conjunto **tipo0** en la zona inferior y el conjunto **tipo1** en la superior, Figura 4.57.

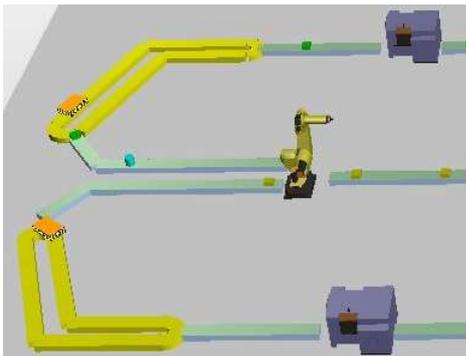


Figura 4.56: representación del modelo 3D sin poka yoke, colocación correcta

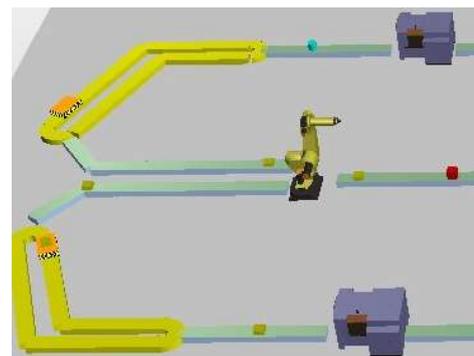


Figura 4.57: representación del modelo 3D sin poka yoke, colocación de forma incorrecta

Al finalizar el proceso, se deberá comprobar que la calidad es la adecuada en cada una de las máquinas, así como que el conjunto se ha realizado correctamente. En la Figura 4.58 se puede apreciar cómo una pieza con defecto de calidad se dispone a entrar en el control de calidad, y en la Figura 4.59 la misma pieza ya colocada en la cinta de piezas defectuosas. Además, se puede apreciar como las piezas que poseen el mismo icono que **tipo0** o **tipo1** se disponen en la cinta de salida.

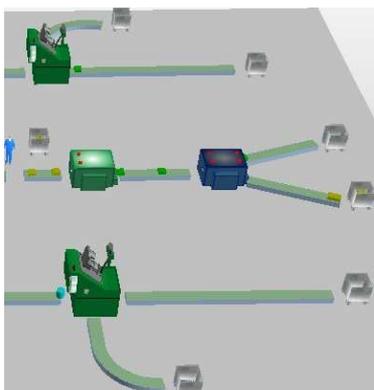


Figura 4.58: representación del modelo 3D sin poka yoke, pieza con defecto de calidad antes de entrar en control

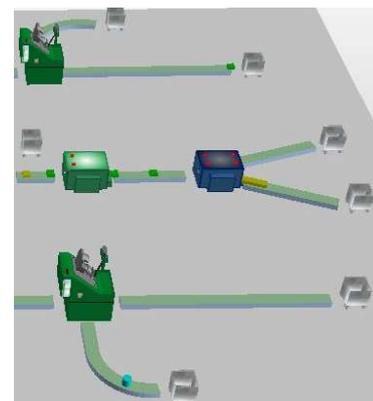


Figura 4.59: representación del modelo 3D sin poka yoke, pieza con defecto de calidad después de entrar en control

Una vez se ha presentado cómo es el modelo sin poka yoke, el funcionamiento del modelo con poka yoke de advertencia es similar, salvo que se produce la salida de piezas que son defectuosas en distintas etapas, como se aprecia en Figura 4.60: salida en la zona común y en las zonas específicas.

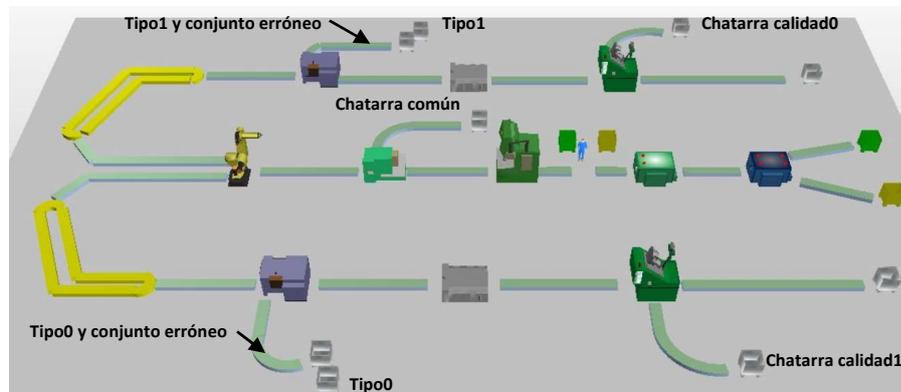


Figura 4.60: representación del modelo 3D con poka yoke advertencia

A continuación, se procederá a visualizar algunos casos en los que se producen salida de piezas del proceso. En la Figura 4.61 se puede apreciar cómo es la salida de piezas en la zona común, dado que son defectuosas de calidad. En la Figura 4.62 se describe cómo se ha producido la salida de piezas en la zona específica a cada uno de los tipos, ya sea porque el ensamblaje no es el correcto o el tipo no es el de esa zona, o ya en el control final donde se realiza el control de calidad de la zona específica a cada tipo.



Figura 4.61: representación del modelo 3D con poka yoke advertencia, control de calidad zona común

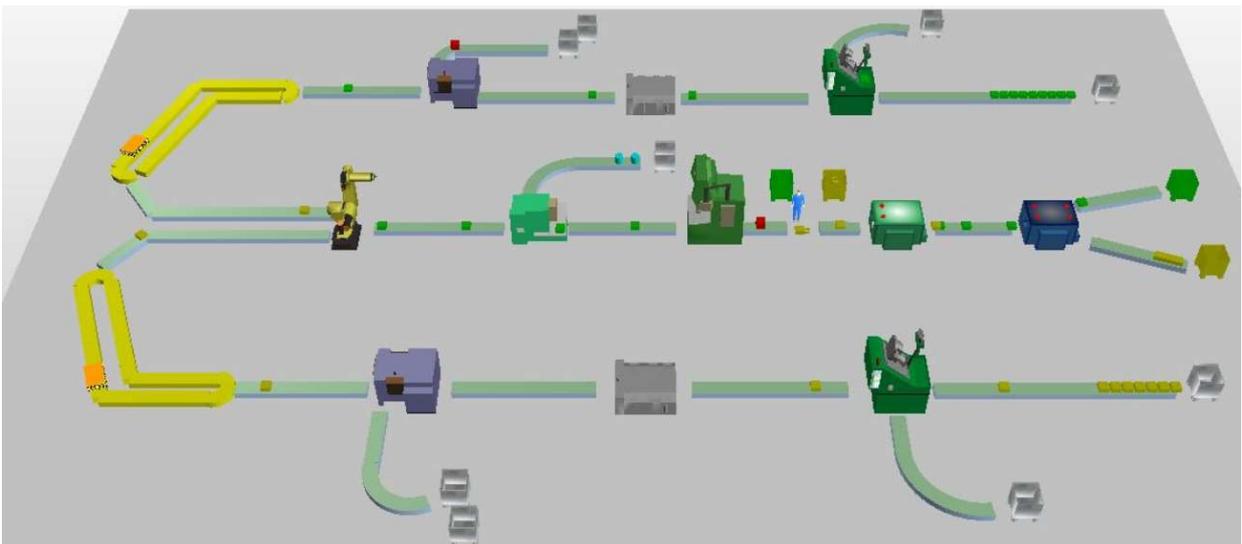


Figura 4.62: representación del modelo 3D con poka yoke advertencia, control de tipo, ensamblaje y calidad en la zona específica

Sólo falta por conocer cómo es la visualización del modelo con poka yoke de fiabilidad, el cuál se puede apreciar en la Figura 4.63. Cabe destacar que, al igual que ocurriese en el modelo sin poka yoke, se posee una salida de piezas al final del proceso para aquellas que no poseen la calidad suficiente, o el ensamblaje o el tipo es erróneo, aunque con los poka yoke de fiabilidad no se deberían generar ninguna pieza defectuosa.

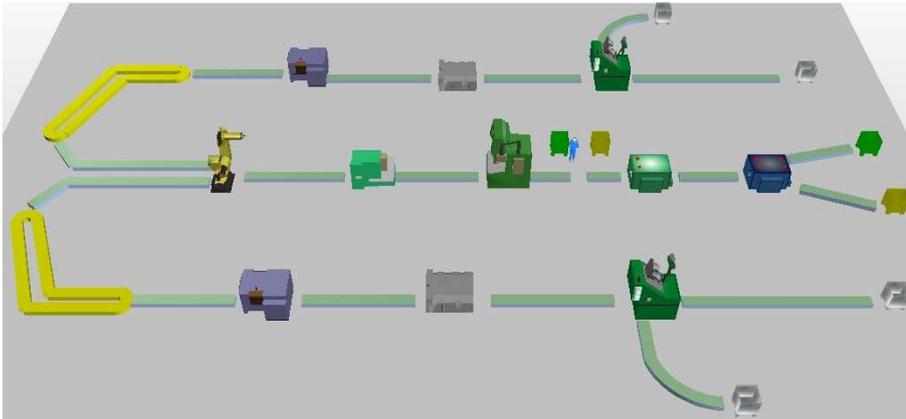


Figura 4.63: representación del modelo 3D con poka yoke de fiabilidad

### 3.2. COMPARACIÓN DE LOS MODELOS

Una vez que se conoce cómo es la visualización de cada uno de los modelos, se procederá a comparar los modelos en instantes determinados (cabe destacar que la implantación de poka yokes hace que aumente un poco el tiempo ciclo, haciendo que los instantes no sean los mismos) para visualizar qué ocurre con ciertas piezas.

Para comenzar, se comprobará qué ocurre en el instante 217, cuando se origina el primer el primer ensamble erróneo. En la Figura 4.64 se puede apreciar cómo se va a realizar el ensamble erróneamente dado que no se posee de poka yoke. Por el contrario, en las Figura 4.65 y Figura 4.66, este error no se genera debido a la presencia del poka yoke de advertencia y el poka yoke de fiabilidad.

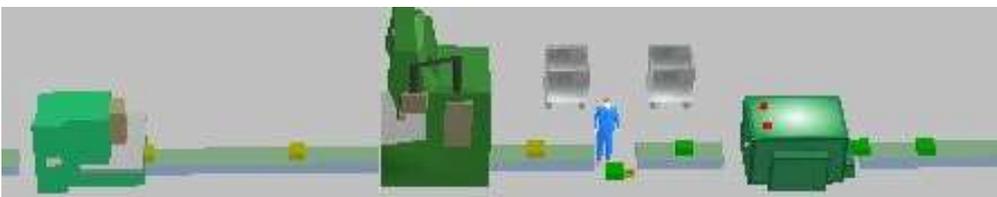


Figura 4.64: representación del modelo 3D sin poka yoke, instante 217



Figura 4.65: representación del modelo 3D con poka yoke de advertencia, instante 217

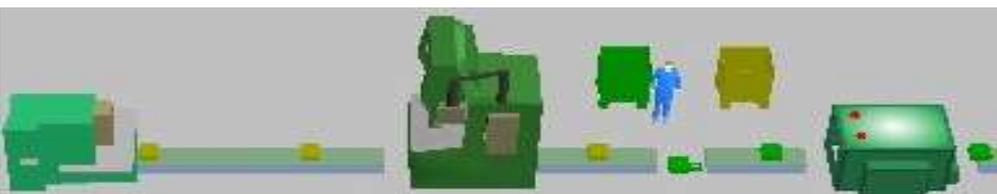


Figura 4.66: representación del modelo 3D con poka yoke de fiabilidad, instante 217

No siempre ocurrirá que el poka yoke de advertencia actúe, algo que se puede apreciar en el instante 387. En el modelo sin poka yoke se produce un ensamblaje incorrecto, **tipo1** con **ens0**, Figura 4.67, aunque en el modelo con poka yoke de advertencia no se soluciona este error, pudiéndose apreciar en la Figura 4.68. En cambio, en el modelo que posee poka yoke de fiabilidad, este error ya no se genera, produciéndose un ensamblaje correcto, Figura 4.69.

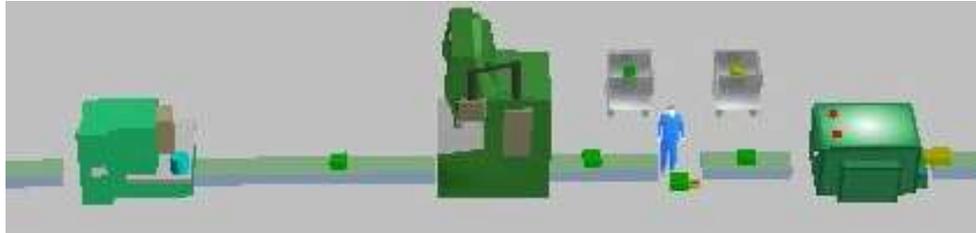


Figura 4.67: representación del modelo 3D sin poka yoke, instante 387

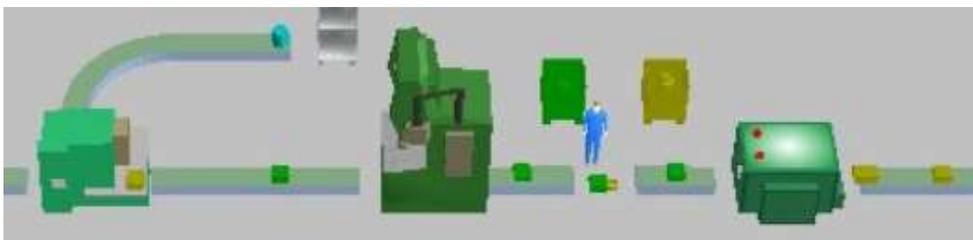


Figura 4.68: representación del modelo 3D con poka yoke de advertencia, instante 387

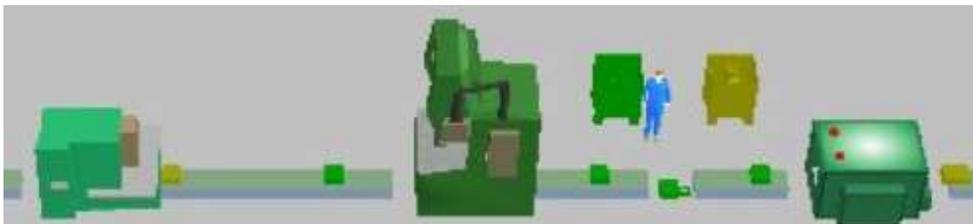


Figura 4.69: representación del modelo 3D con poka yoke de fiabilidad, instante 387

En la **maquina5** también se ha dispuesto de un poka yoke de advertencia; por tanto, será necesario conocer cuál es el impacto de éste en el modelo. En la Figura 4.70 se puede apreciar como una pieza de **tipo1** se dispone en la cinta que lleva a la zona superior. Por el contrario, esta misma pieza en el modelo con poka yoke de advertencia se dispone en su cinta correspondiente, Figura 4.71; igual que ocurre en el modelo con poka yoke de fiabilidad, representado en la Figura 4.72

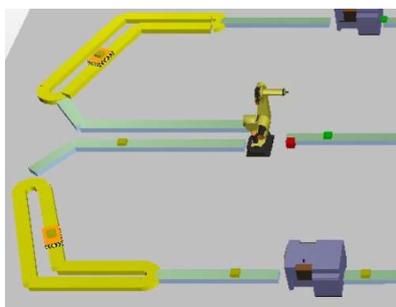


Figura 4.70: representación del modelo 3D sin poka yoke, instante 302

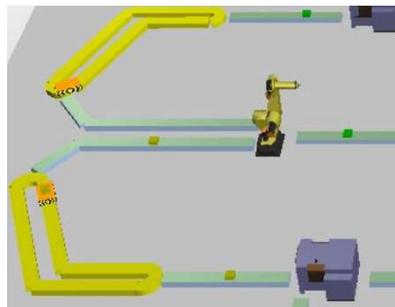


Figura 4.71: representación del modelo 3D con poka yoke de advertencia, instante 302

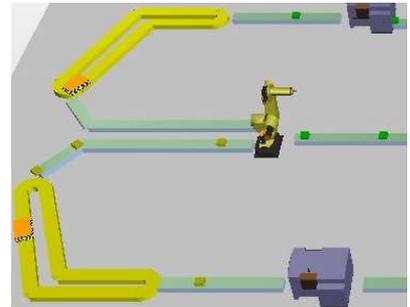


Figura 4.72: representación del modelo 3D con poka yoke de fiabilidad, instante 302

Como en el caso de la máquina de ensamblaje, hay casos en los que el poka yoke de advertencia de la **maquina5** no actúa. Por ejemplo, en el instante 556, en el que se depositan

piezas de tipo1 en la cinta asociada al tipo0, Figura 4.73, actuando de la misma forma que en el modelo sin poka yoke, Figura 4.74. En el modelo con poka yoke de fiabilidad se aprecia cómo estas piezas se depositan de forma correcta en la zona inferior, Figura 4.75.

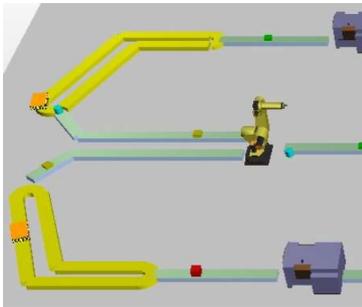


Figura 4.73: representación del modelo 3D sin poka yoke, instante 556

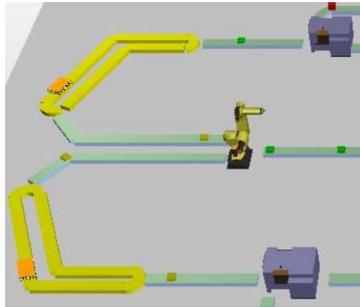


Figura 4.74: representación del modelo 3D poka yoke de advertencia, instante 556

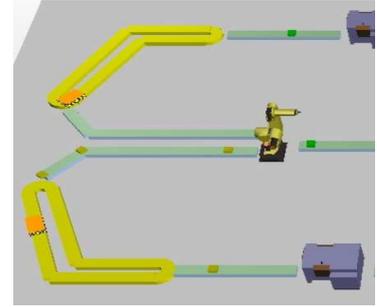


Figura 4.75: representación del modelo 3D con poka yoke de fiabilidad, instante 556

Después de haber visualizado cómo se comportan los distintos modelos para las mismas piezas, se procederá a visualizar el modelo en otros instantes, visualizándose la acumulación de las piezas en las distintas cintas.

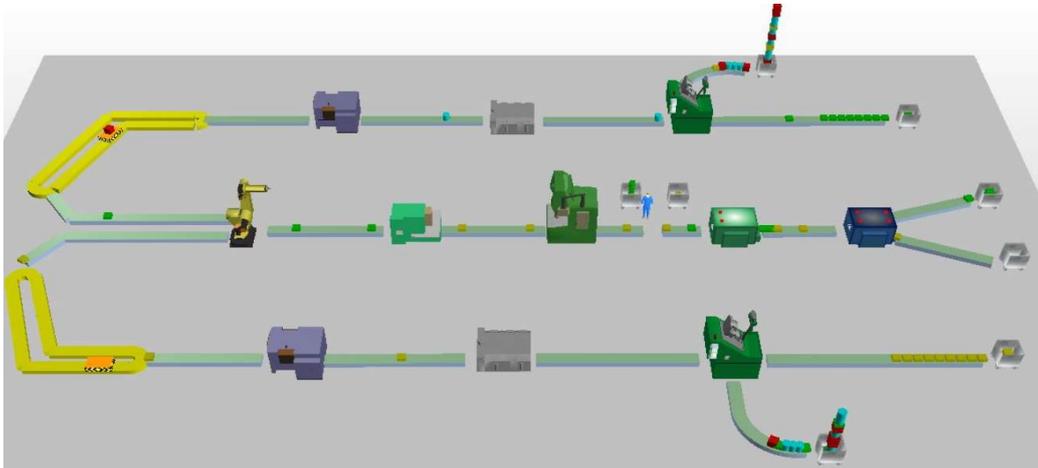


Figura 4.76: representación del modelo 3D sin poka yoke, instante 1500

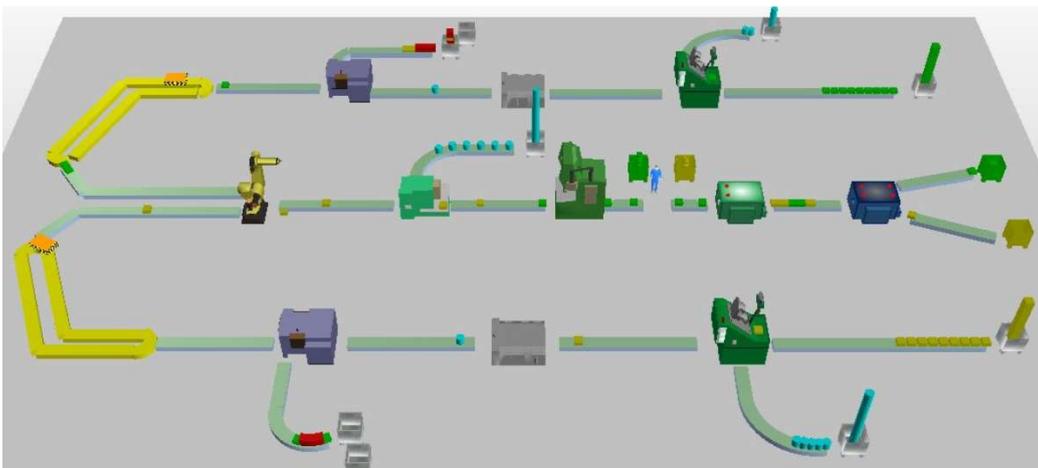


Figura 4.77: representación del modelo 3D con poka yoke de advertencia, instante 1500

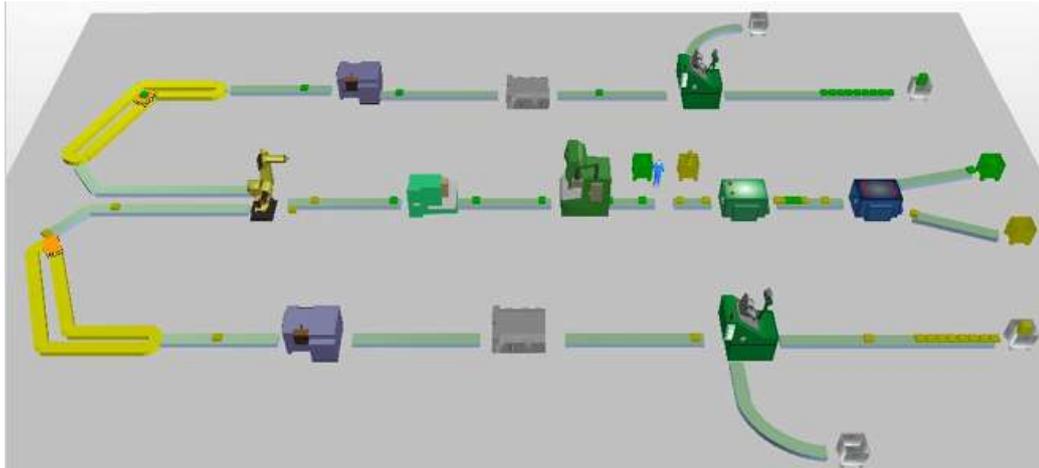


Figura 4.78: representación del modelo 3D con poka yoke de fiabilidad, instante 1500

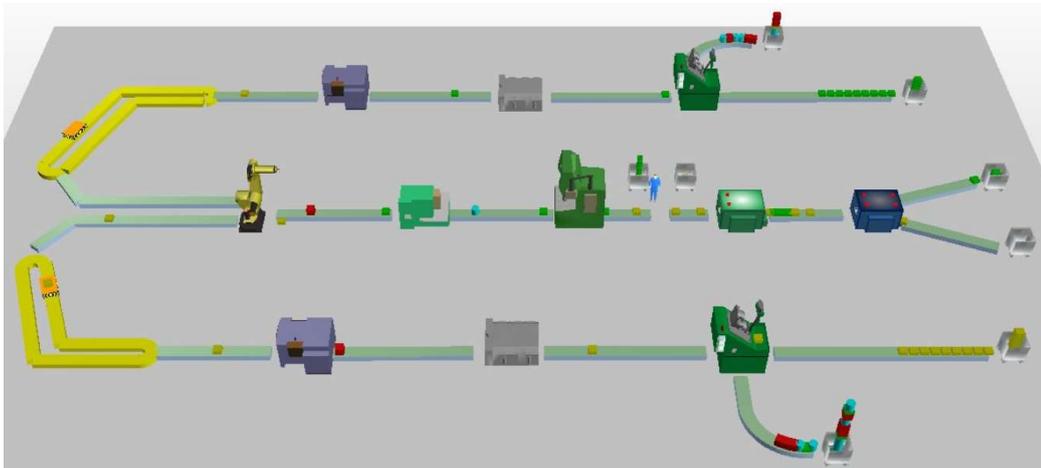


Figura 4.79: representación del modelo 3D sin poka yoke, instante 2500

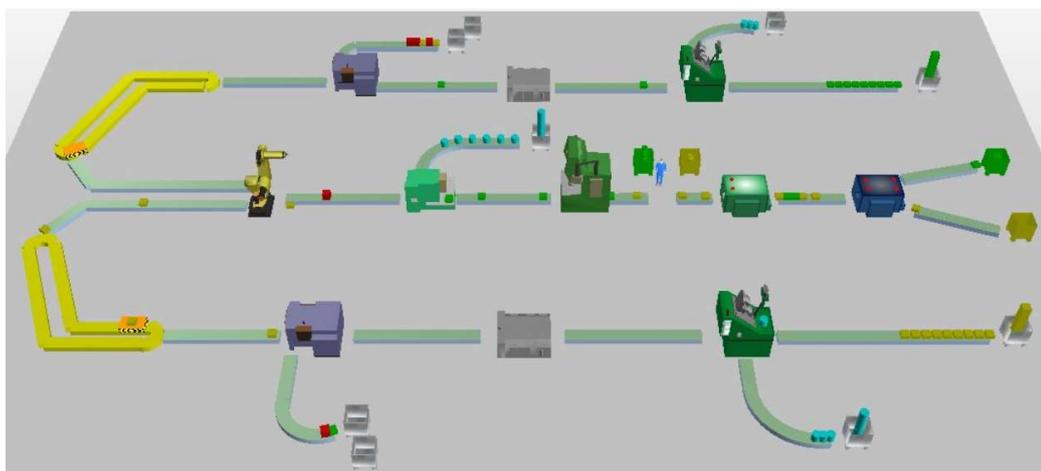


Figura 4.80: representación del modelo 3D con poka yoke de advertencia, instante 2500

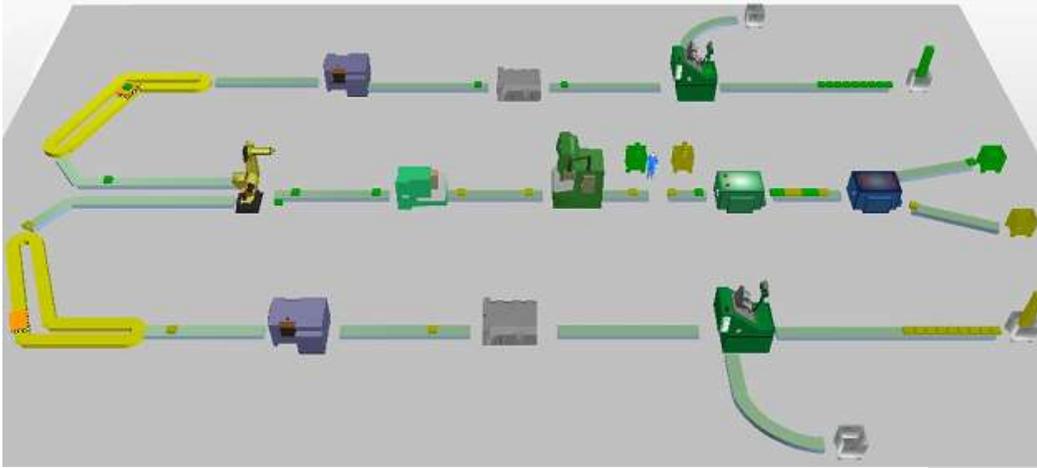


Figura 4.81: representación del modelo 3D con poka yoke de fiabilidad, instante 2500

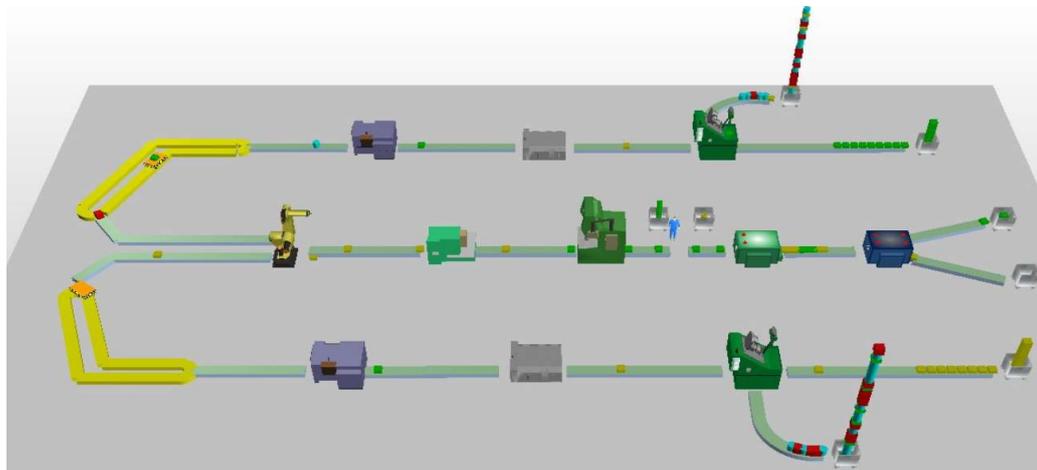


Figura 4.82: representación del modelo 3D sin poka yoke, instante 3500

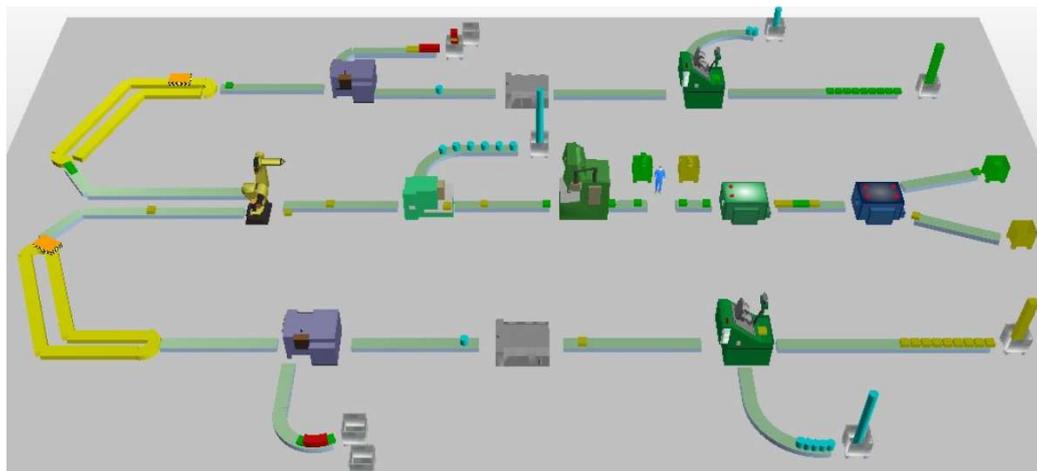


Figura 4.83: representación del modelo 3D con poka yoke de advertencia, instante 3500

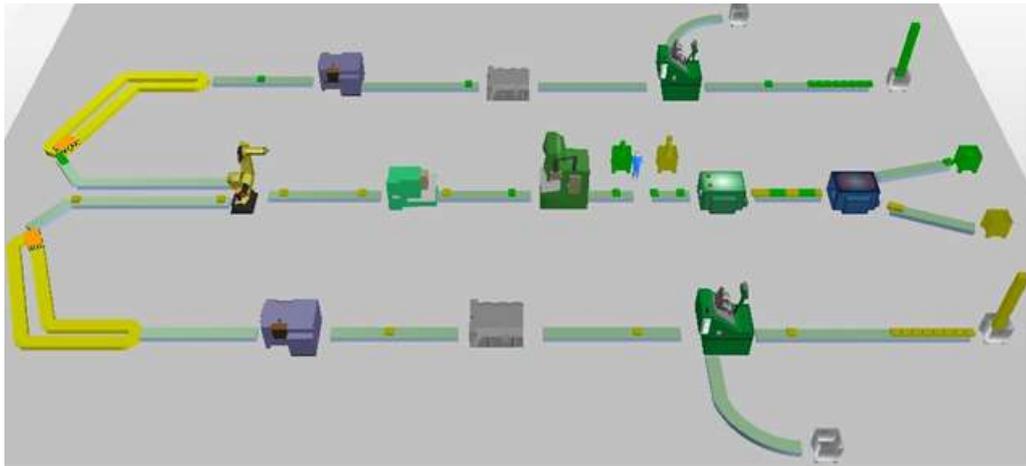


Figura 4.84: representación del modelo 3D con poka yoke de fiabilidad, instante 3500

Una vez visualizados los modelos, y sabiendo que se produce la salida de las piezas del modelo en ciertos instantes, en la Tabla 4.3 se pueden apreciar el número de piezas defectuosas, las piezas que entran en el modelo y las que salen.

Tiempo	Sin poka yoke			Poka yoke advertencia					Poka yoke fiabilidad	
	Piezas entrada	Piezas salida	Piezas chatarra	Piezas entrada	Piezas salida	Chatarra común	Chatarra ensamble	Chatarra zona específica	Piezas entrada	Piezas salida
1500	121	51	41	121	73	8	6	3	121	88
2500	201	97	71	201	135	16	6	11	201	162
3500	281	147	98	281	187	21	9	23	281	237

Tabla 4.3: comparación del número de piezas en modelo poka yoke a diferentes tiempos

#### 4. TÉCNICA CÉLULA EN U

Los modelos que se van a visualizar a continuación son ejemplos en los que los operarios tienen que llevar las piezas de una máquina a otra, en donde se va a poder visualizar la gran diferencia entre ambos modelos, la disposición de las máquinas.

##### 4.1. MODELO CON DISPOSICIÓN LINEAL

Para comenzar, se puede visualizar el modelo en la Figura 4.85, prestando especial atención a los paths que unen las distintas máquinas por las que se tienen que mover los operarios. Una vez que se conoce cómo es la visualización del modelo, se procederá a explicar mediante imágenes cómo es su funcionamiento.



Figura 4.85: visualización del modelo 3D acompañar pieza lineal

Para simular un estado estacionario, en el momento inicial se disponen piezas en la entrada\_proceso, en la maquina4 y en la maquina8, tal y como se aprecia en la Figura 4.86.

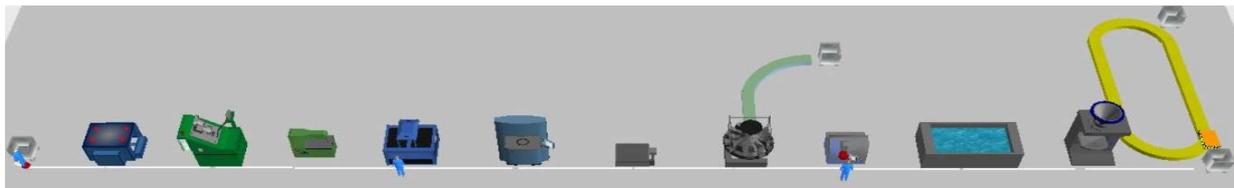


Figura 4.86: visualización del instante inicial modelo 3D acompañar pieza lineal

Cuando se han acabado las operaciones en estas máquinas, los operarios trasladarán las piezas por los paths correspondientes, como se puede apreciar en la Figura 4.87, donde se observa como el `op_ent_1_2_3` está trasladando la `pieza_prin` por el path `entrada_maquina1`, mientras que el `op8_9_10_sal` está cogiendo el `conjunto` de la `maquina8`. El `op4_5_6_7` todavía se encuentra en la `maquina4` debido a que no ha finalizado su operación.



Figura 4.87: visualización del traslado pieza hacia maquina1 y hacia maquina9 instante 70 modelo 3D acompañar pieza lineal

En el momento en el que se finalice la actividad en la `maquina4` (instante 130), el `op4_5_6_7` llevará la `pieza_prin` a la `maquina5` por el path `camino4_5`, Figura 4.88, mientras que el `op_ent_1_2_3` se encuentra en la `maquina1` y el `op8_9_10_sal` en la `maquina9`.

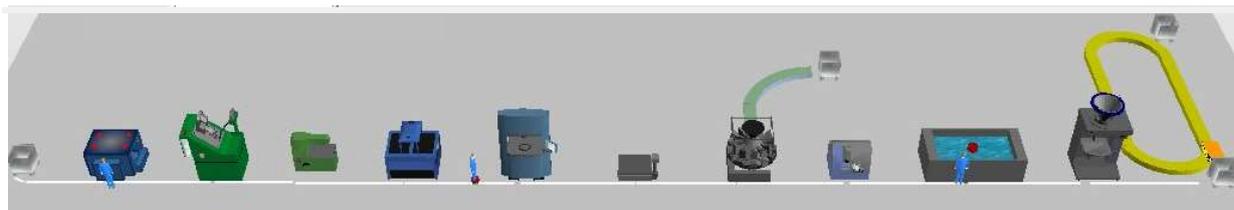


Figura 4.88: visualización del traslado pieza entre maquina4 y maquina5 instante 130.8 modelo 3D acompañar pieza lineal

Los operarios se trasladarán por las máquinas llevando las piezas hasta que lleguen a la última máquina de cada zona. En el instante 280, Figura 4.89, se puede apreciar como el `op_ent_1_2_3` se encuentra en la `maquina3` (última de la zona1) y el `op8_9_10_sal` en la `salida_proceso` (última de la zona3). En ese momento, tendrán que desplazarse por los paths que han recorrido con anterioridad para situarse en la primera máquina, como se puede apreciar en la Figura 4.90 en el caso del `op_ent_1_2_3`. Además, en esta figura se puede apreciar otras actividades: el `op8_9_10_sal` no se dirige a la `maquina7`, debido a que el `op4_5_6_7` no ha acabado su actividad, y, por tanto, el `conjunto` no está listo para llevarse a la `maquina8`, no siendo necesaria la presencia del `op8_9_10_sal`; el `conjunto` que ha depositado el `op8_9_10_sal` en el contenedor `cont_salida` ha salido de éste y el `vehiculo` lo lleva al contenedor `salida_conjunto`. En el momento en el que se acabe la actividad de la `maquina7`, el `op4_5_6_7` se dirigirá hacia la `maquina3` por los paths que unen las máquinas, y el `op8_9_10_sal` se dirigirá a la `maquina7` para coger el `conjunto` como se aprecia en la Figura 4.91.

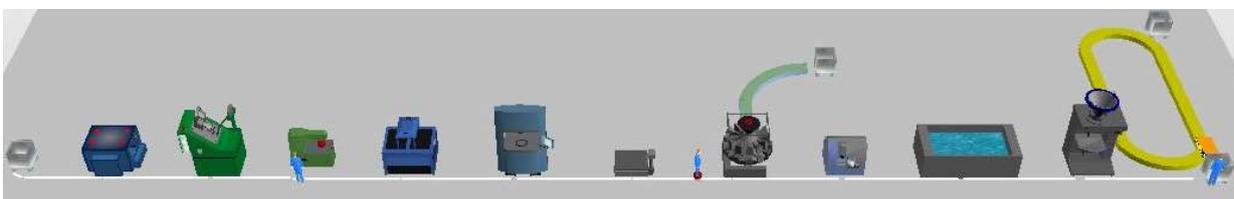


Figura 4.89: visualización del instante 280, finalización de un ciclo en dos zonas, modelo 3D acompañar pieza lineal

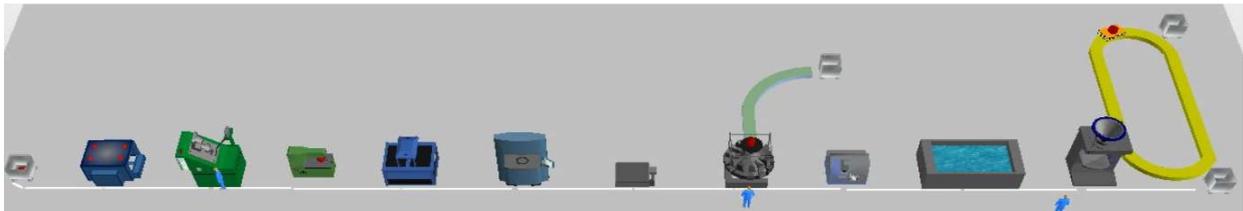


Figura 4.90: visualización del instante 344, desplazamiento a la primera máquina de la zona, modelo 3D acompañar pieza lineal

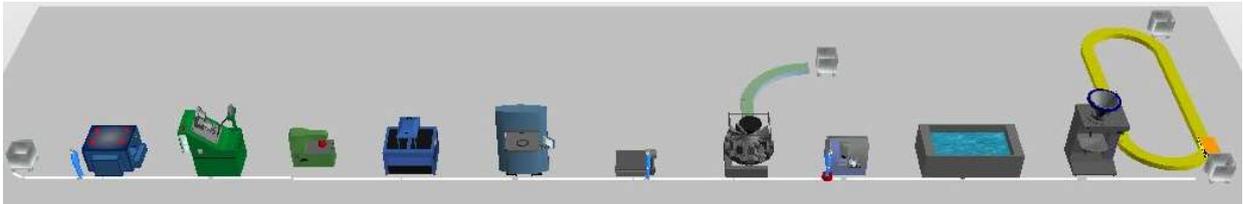


Figura 4.91: visualización del instante 376, desplazamiento a la primera máquina zona1 y 2, modelo 3D acompañar pieza lineal

El proceso que se ha visualizado anteriormente se repetirá de igual forma que la explicada, pudiéndose apreciar diferencias en el caso de finalizar alguna zona y que el operario se tenga que ir a su zona de descanso porque la pieza que deba llevar no se encuentre disponible.

#### 4.2. MODELO CON CÉLULA EN U

En la Figura 4.92 se puede apreciar cómo es la visualización del modelo, pudiéndose observar las distintas zonas en las que se divide el modelo, así como los operarios que transportan las piezas de una máquina a otra.



Figura 4.92: visualización del modelo 3D acompañar pieza en U

Como ocurriese en el modelo descrito anteriormente, en el momento inicial se disponen piezas en determinadas máquinas, como se puede apreciar en la Figura 4.93. Se dispondrán piezas en las *maquina10*, *maquina6* y *maquina2*, estando los operarios dispuestos en dichas máquinas.

A continuación, se va a proceder a explicar cómo se desarrolla el proceso productivo. Cuando acaba la *maquina6* de realizar su operación, el *op4\_5\_6\_7* llevará la pieza por el *camino6\_7* mientras que el *op\_ent\_1\_10\_sal* se encuentra en la *maquina10* y el *op3\_4\_8\_9* se encuentra en la *maquina2*, como se puede apreciar en la Figura 4.94.

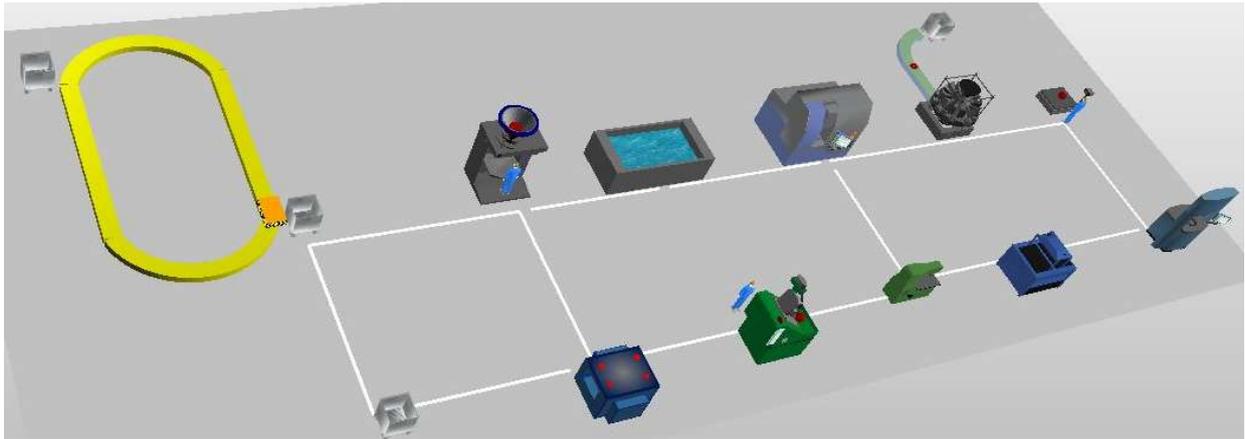


Figura 4.93: visualización del instante inicial modelo 3D acompañar pieza en U

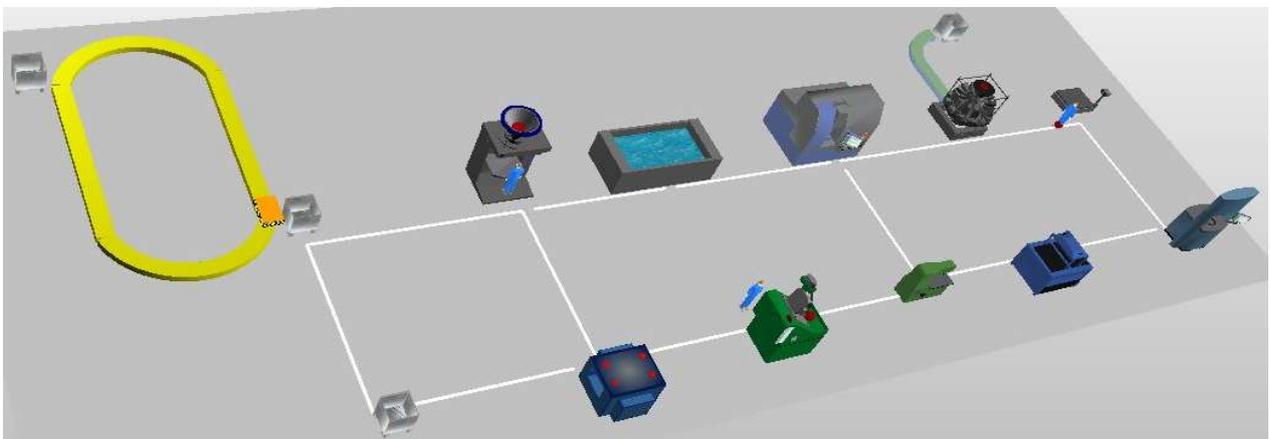


Figura 4.94: visualización del instante 61 modelo 3D acompañar pieza en U

El `op4_5_6_7` realiza la operación en la `maquina7` y lleva el conjunto a la `maquina8` mediante el path `camino7_8`. Cuando deje el conjunto en la `maquina8`, el `op2_3_8_9` tendrá que ir hacia dicha máquina, dado que la `maquina8` se encuentra en la zona correspondiente a ese operario. En la Figura 4.95 se puede apreciar cómo se encuentran los dos operarios en esta `maquina8`, dado que es el instante en el que el `op4_5_6_7` deja el `conjunto` y el `op2_3_8_9` se dirige a esa máquina para realizar la operación. Además, se puede apreciar como el `op_ent_1_10_sal` se encuentra colocando el conjunto en `cont_salida`.

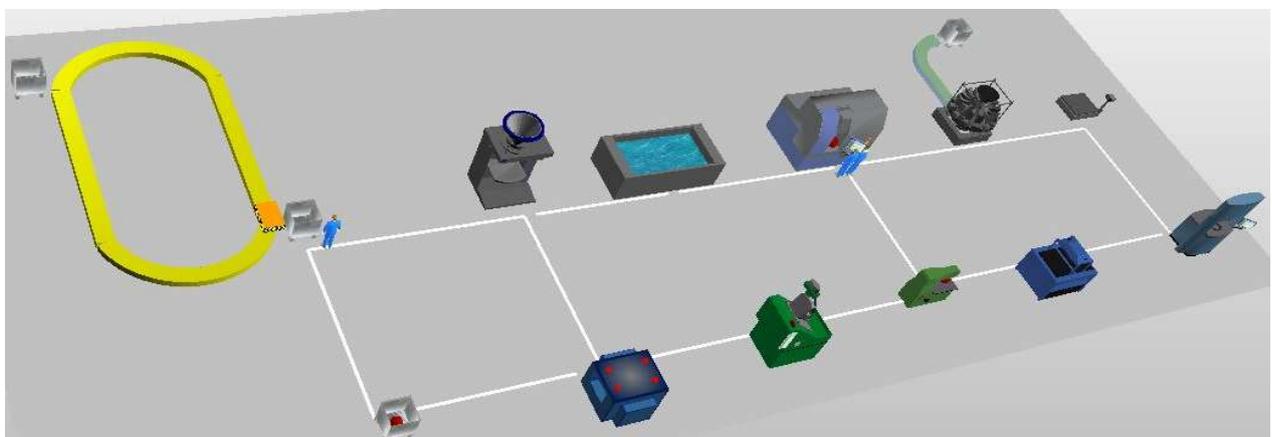


Figura 4.95: visualización del instante 174, comienzo de operación maquina8 modelo 3D acompañar pieza en U

En la Figura 4.96 se visualiza cómo se el `conjunto` se ya ha salido del contenedor `cont_salida` y se encuentra en el `vehiculo` para dirigirse al contenedor `salida_conjunto`. El `op4_5_6_7` ya ha

finalizado las actividades de su zona, dirigiéndose por el path **camino8\_3** para recoger la **pieza\_prin** que se encuentra en la **maquina3** esperando a ser recogida por el dicho operario y comenzando de nuevo la zona3. La **maquina8** se encuentra realizando la operación, estando el **op3\_4\_8\_9** en dicha máquina.

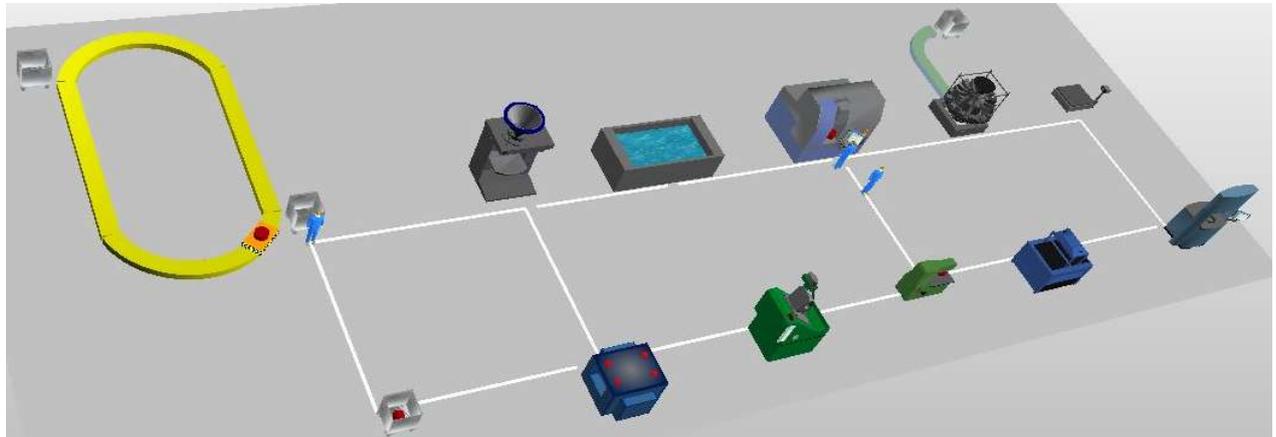


Figura 4.96: visualización instante 176, operación maquina8 y desplazamiento hacia maquina3 modelo 3D acompañar pieza en U

Al dejar la pieza en el contenedor **cont\_salida** se finaliza la zona1, haciendo que se tenga que desplazar por el path **camino\_sal\_ent** para sacar la **pieza\_prin** del **cont\_pieza\_prin**, como se puede apreciar en la Figura 4.97. Además, se puede visualizar cómo comienza la actividad de la zona3, llevando el **op4\_5\_6\_7** la **pieza\_prin** hacia la **maquina4**. En el contenedor **cont\_salida** se puede observar cómo ya se encuentra el **conjunto** que ya se encuentra en **salida\_conjunto**.

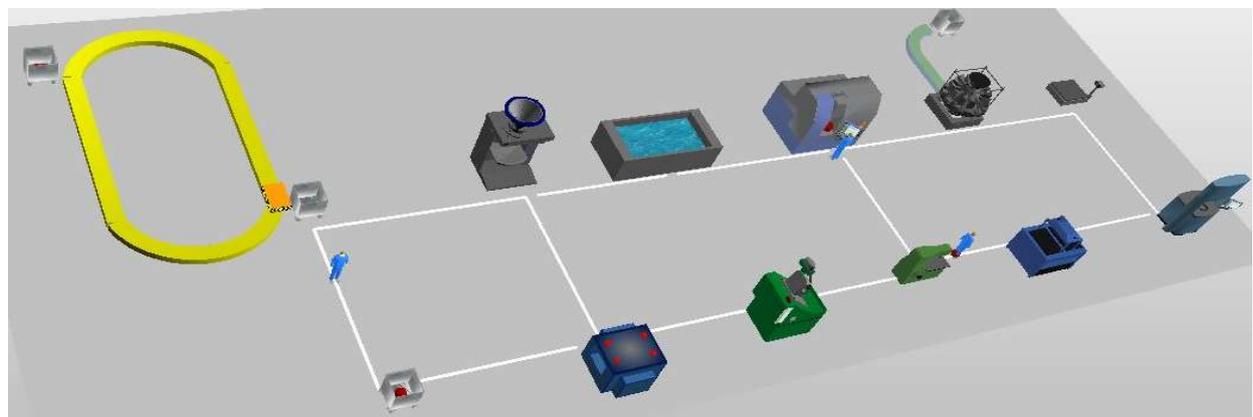


Figura 4.97: visualización del instante 1097.6, comienzo de la zona1 y 3 modelo 3D acompañar pieza en U

### 4.3. COMPARACIÓN DE RESULTADOS

Una vez se han explicado cómo se comportan ambos modelos, se procederá a visualizar los modelos en distintos instantes (en el 200, 4.000, 10.000 y 20.000) para visualizar las diferencias entre el modelo que posee técnica Lean y el que no la posee. En la Tabla 4.4 se puede apreciar la diferencia de piezas que se generan en los distintos modelos; esto se debe a los excesivos tiempos que se emplean en los trayectos en forma lineal.

Tiempo	Lineal		Célula en U	
	Piezas entrada	Piezas salida	Piezas entrada	Piezas salida
2000	8	5	8	5
4000	13	10	13	10
10000	27	23	29	26
20000	51	47	52	55

Tabla 4.4: comparación del número de piezas en modelo poka yoke a diferentes tiempos

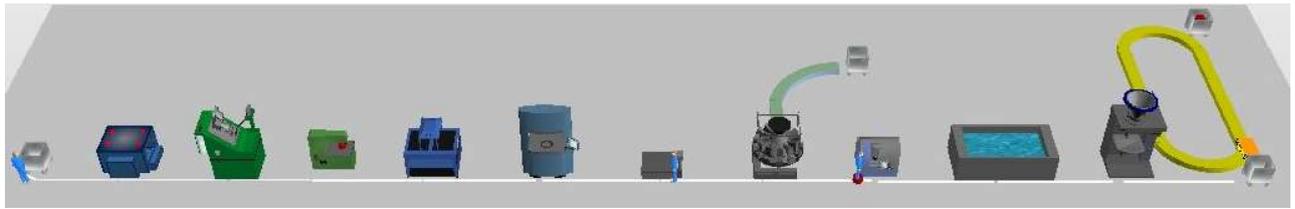


Figura 4.98: visualización del instante 2000, modelo 3D acompañar pieza lineal

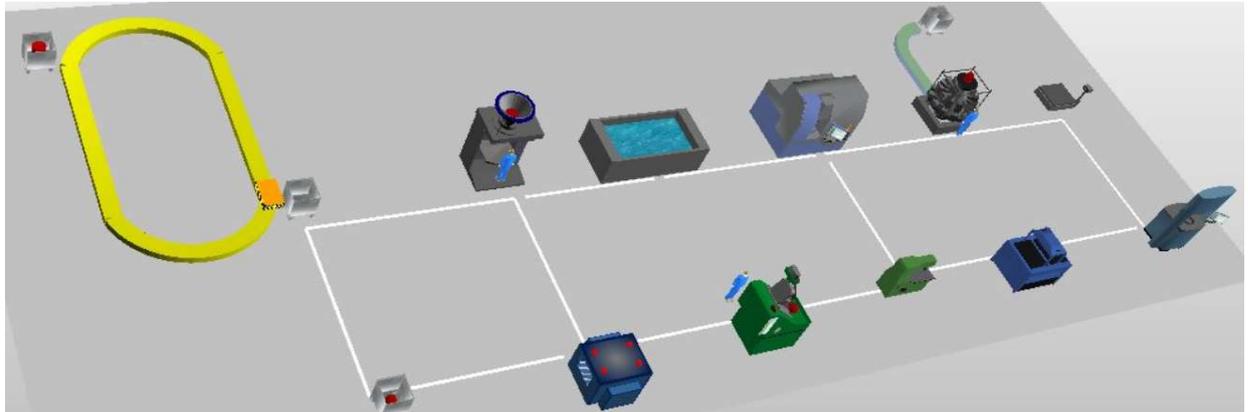


Figura 4.99: visualización del instante 2000, modelo 3D acompañar pieza en U

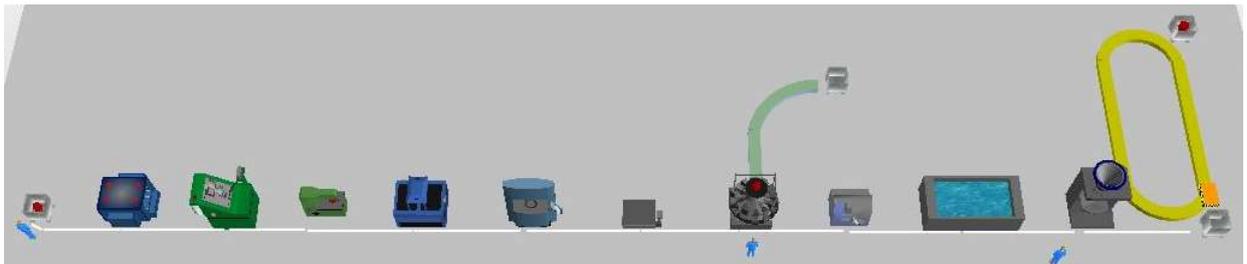


Figura 4.100: visualización del instante 4000, modelo 3D acompañar pieza lineal

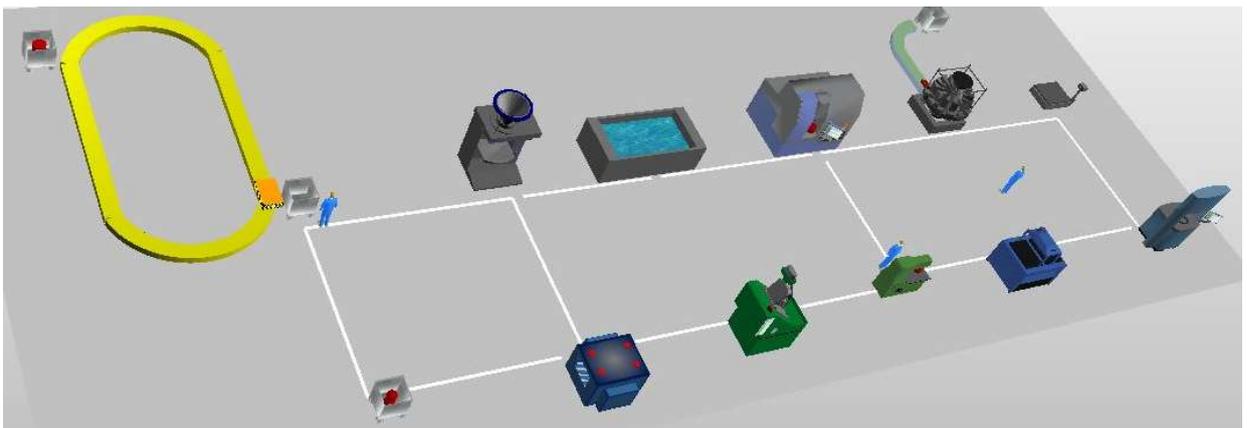


Figura 4.101: visualización del instante 4000, modelo 3D acompañar pieza en U

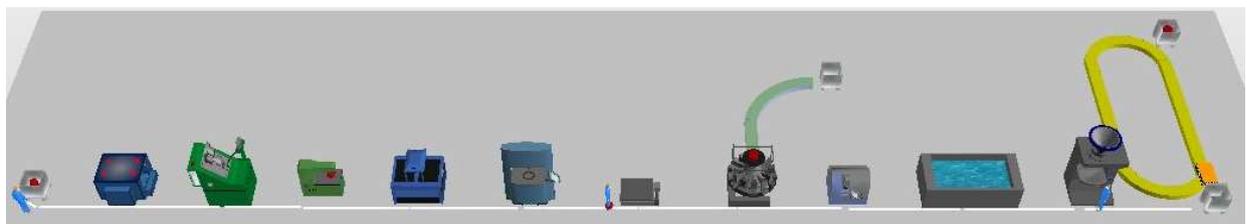


Figura 4.102: visualización del instante 10000, modelo 3D acompañar pieza lineal



## 5.1. MODELO SIN HEIJUNKA NI SMED

Para comenzar, se visualizará el modelo sin Heijunka ni Smed, Figura 4.106 (será el mismo que el modelo con Heijunka y Smed, dado que estas mejoras no tienen una representación física, sino que son conceptuales). En éste se pueden visualizar los elementos a los que se van a referir a lo largo de la explicación de la visualización. Una vez que se conoce cómo es la representación del modelo, se procederá a visualizar su funcionamiento. Primero, se desarrollará cómo es la visualización de la zona común a todas las piezas y, posteriormente, se explicará el funcionamiento de la máquina **unión**.

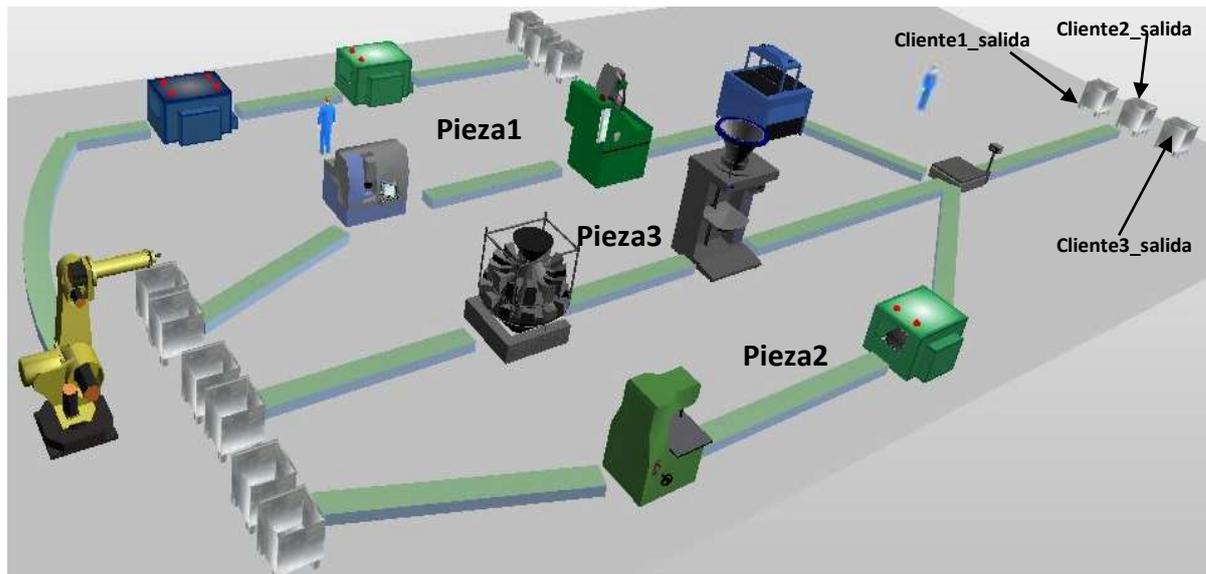


Figura 4.106: visualización del modelo 3D técnica heijunka y smed

Como se ha comentado, a la hora de describir el modelo, las piezas se procesan en grandes lotes, lo que puede apreciarse en la Figura 4.107. Cuando se termina de procesar las **piezas1**, se comenzará a procesar las **piezas2**; como se produce un cambio del **tipo** de piezas, habrá que realizar una acción en la **maq1** simulando el cambio de tipo, como se puede apreciar en la Figura 4.108. Debido a que no se ha implementado técnica Smed, esta acción llevará una gran cantidad de tiempo, como se puede apreciar en la salida de la primera **pieza2**, Figura 4.109. Se procederá de igual forma cuando lleguen las piezas a la **maq2**, lo que puede verse en la Figura 4.110.

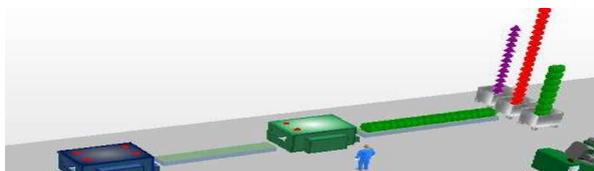


Figura 4.107: lotes de piezas en trans1 en el modelo 3D sin heijunka ni smed

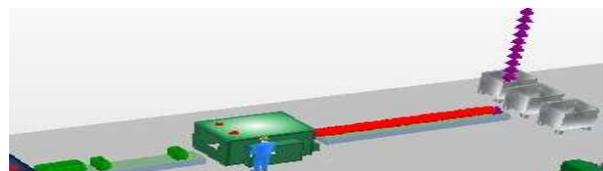


Figura 4.108: acción en maq1 debida al cambio del tipo de pieza en el instante 324, modelo 3D sin heijunka ni smed

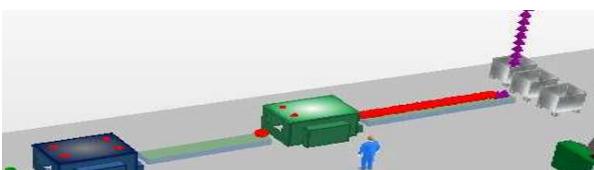


Figura 4.109: primer pieza2 salida de la maq1 instante 392, modelo 3D sin heijunka ni smed

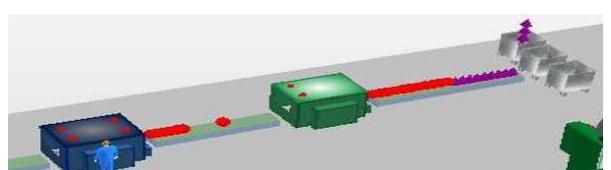


Figura 4.110: acción en maq2 debida al cambio del tipo de pieza en el instante 400, modelo 3D sin heijunka ni smed

A continuación, se explicará cómo es el proceso en la zona específica y la **unión**. Como se ha comentado al describir el proceso, las piezas entran a cada zona asociada al tipo de pieza en lotes de 3 piezas, como se puede apreciar en la Figura 4.111.

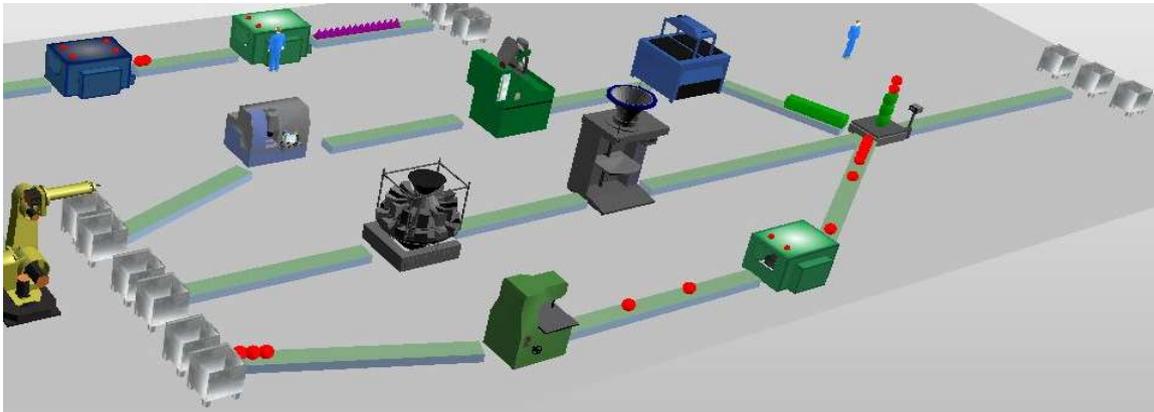


Figura 4.111: entrada en zona específica a cada pieza en lote de 3 piezas, modelo 3D sin heijunka ni smed

La máquina **union** realizará el ensamblaje de distinto número de piezas dependiendo del cliente al que tenga que fabricar los conjuntos; el primero será el **cliente1**. Por tanto, las necesidades serán 3 **piezas1**, 2 **piezas2** y 1 **pieza3**, como se puede apreciar en la Figura 4.112. Cuando acabe **union** de realizar la operación, el conjunto que se forma será el que se visualice en la Figura 4.113. Se realizarán todos los lotes asociados al **cliente1**, es decir, 10, como se puede apreciar en la Figura 4.114. Además, se puede apreciar cómo **union** ya ha cogido las piezas necesarias para el **cliente2**.

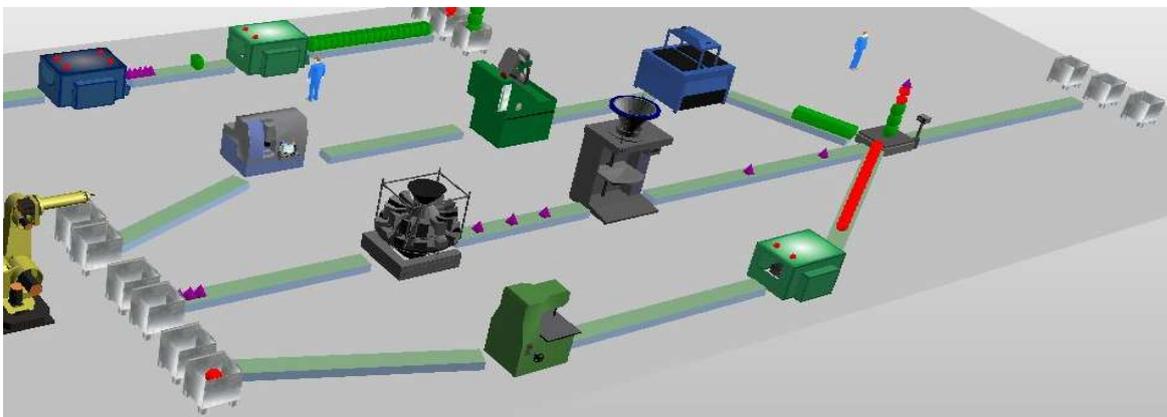


Figura 4.112: piezas necesarias para el ensamblaje cliente1, modelo 3D sin heijunka ni smed

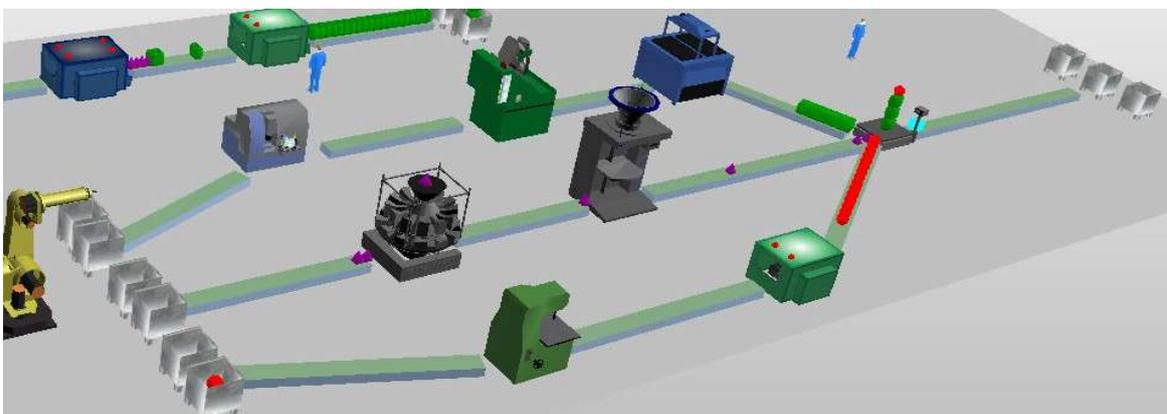


Figura 4.113: visualización del primer conjunto cliente1 instante 985 modelo 3D sin heijunka ni smed

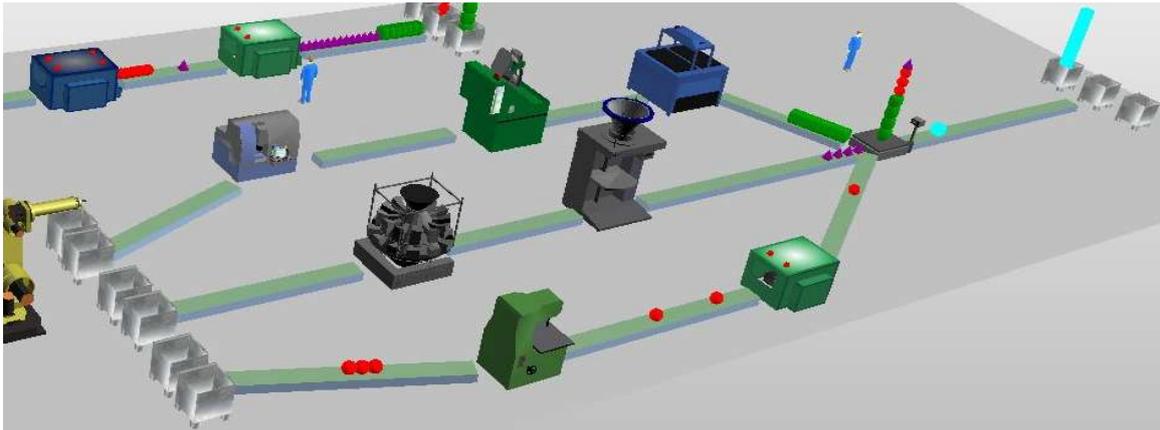


Figura 4.114: visualización del último conjunto para el cliente1 en el instante 1705, modelo 3D sin heijunka ni smed

En el instante en el que se comience a producir para el **cliente2** se tendrá que realizar una operación asociada al cambio de tipo en la máquina **union**, como se puede apreciar en la Figura 4.115. Cuando se acabe la operación, se podrá visualizar como se genera el **lote\_cliente2**, Figura 4.116. Además, se puede visualizar cómo las piezas **lote\_cliente1** ya no se encuentran en **cliente1\_salida** dado que ya se encuentran los 10 conjuntos. Se realizarán 6 conjuntos, como se puede apreciar en la Figura 4.117.

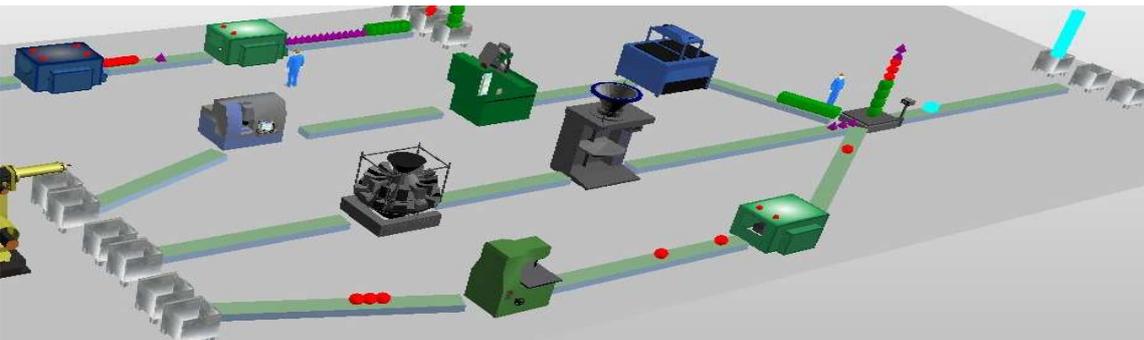


Figura 4.115: acción asociada al cambio de cliente1 a cliente2 en unión, modelo 3D sin heijunka ni smed

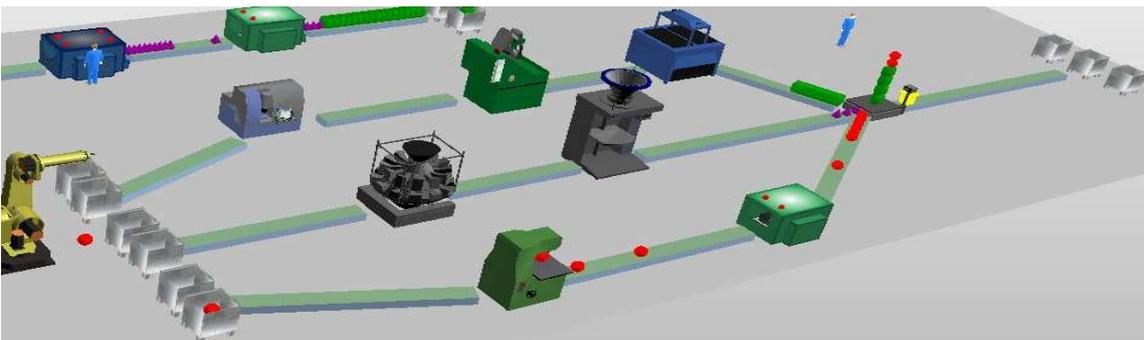


Figura 4.116: visualización del primer conjunto cliente2 en el instante 1776, modelo 3D sin heijunka ni smed

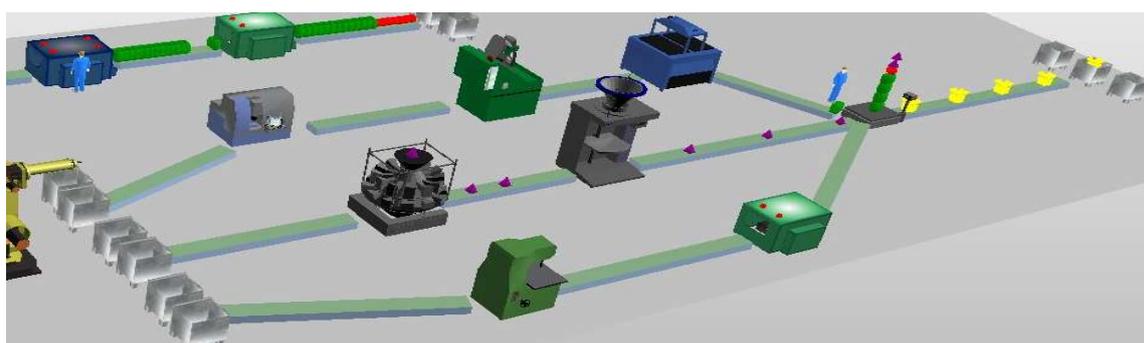


Figura 4.117: visualización del último conjunto cliente2 y piezas cliente3, modelo 3D sin heijunka ni smed

Como cuando se acabaron de producir piezas para el **cliente1**, se tendrá que hacer operaciones asociadas al cambio de cliente en la máquina **union**. La primera pieza se podrá visualizar en la Figura 4.118. Cuando se acaben de fabricar los 4 **lote\_cliente3**, se volverá a comenzar a fabricar para el **cliente1**, como se puede apreciar en la Figura 4.119.

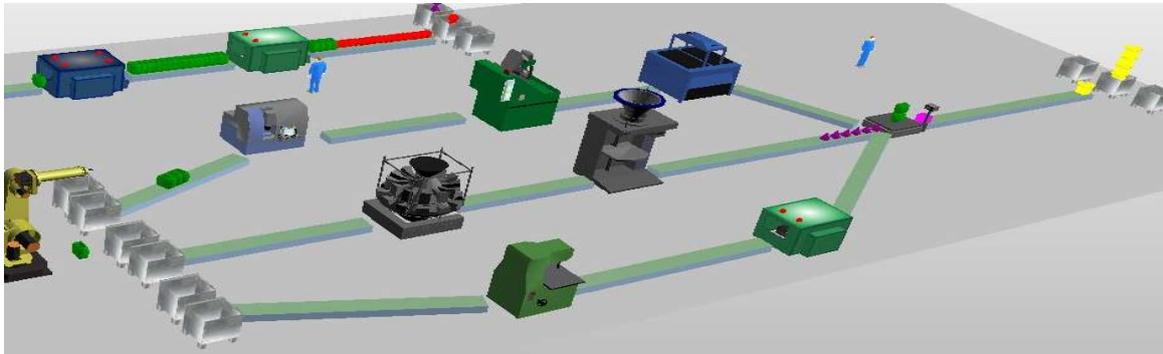


Figura 4.118: visualización del último conjunto cliente2 y primer conjunto cliente3, modelo 3D sin heijunka ni smed

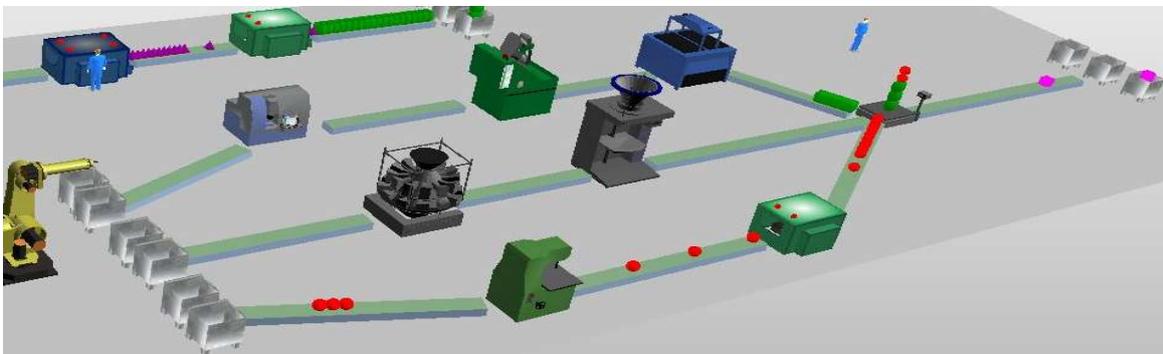


Figura 4.119: visualización del último conjunto cliente3 y comienzo conjunto cliente1, modelo 3D sin heijunka ni smed

Una vez que se ha explicado cómo es el funcionamiento de este modelo, se va a indicar en la Tabla 4.5 los tiempos en los que se producen la primera y la última pieza de cada uno de los lotes de cada **cliente**.

	Cliente1	Cliente2	Cliente3
Primera pieza	985	1776	2640
Última pieza	1705	2570	3205

Tabla 4.5: tiempos en los que se producen piezas para cada cliente, modelo sin heijunka ni smed

## 5.2. MODELO CON HEIJUNKA Y SMED

Una vez que se conoce cómo es el modelo sin Heijunka ni Smed, se procederá a visualizar cómo se comporta el modelo aplicando técnicas Heijunka y Smed. Para comenzar se procederá a visualizar los lotes que entran en la cinta **trans1**, como se puede apreciar en la Figura 4.120, se poseen lotes más reducidos que en el modelo sin Heijunka ni Smed. Como en el caso anterior, hay que realizar tareas previas en las máquinas debido al cambio de tipo, en la Figura 4.121 se puede apreciar el debido al cambio de **pieza1** a **pieza2**. En la Figura 4.122 se observa el momento en el que sale la primera **pieza2** después del cambio; como se puede comprobar, los tiempos asociados al cambio de tipo son muy inferiores comparados con el modelo sin Heijunka ni Smed.

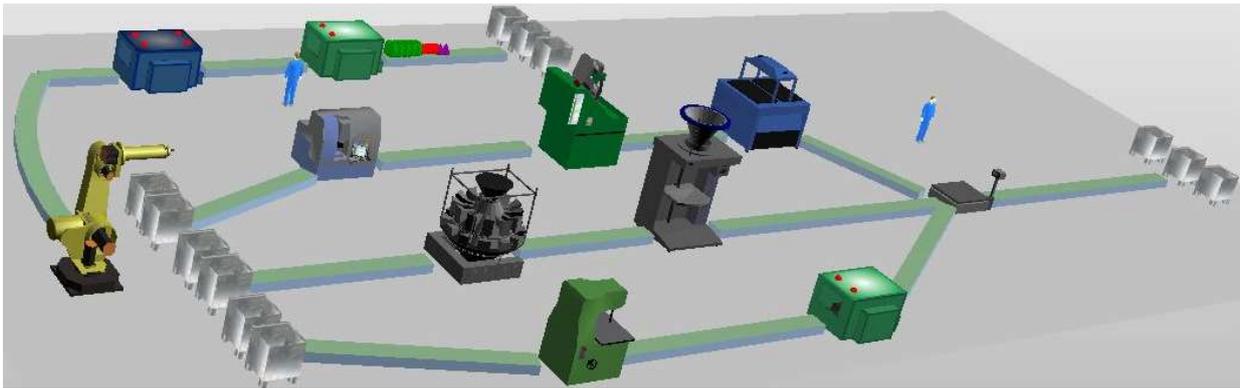


Figura 4.120: visualización de los lotes que entran en la cinta trans1, modelo 3D con heijunka y smed

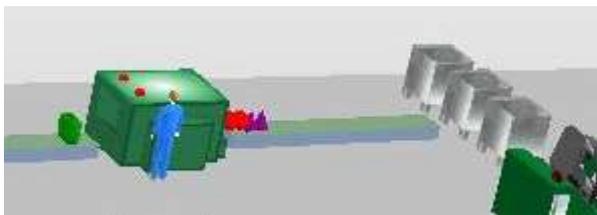


Figura 4.121: acción asociada al cambio tipo en maq1 instante 189, modelo 3D con heijunka y smed

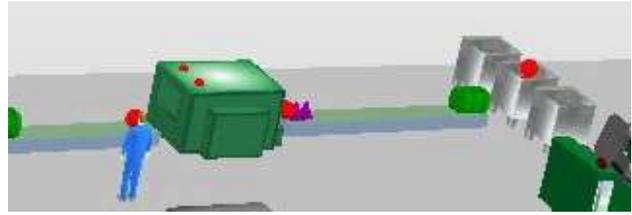


Figura 4.122: primera pieza2 después cambio de tipo instante 203.5, modelo 3D con heijunka y smed

A continuación, se procederá a visualizar cómo procede la máquina **unión**. En este caso, se producirán conjuntos en lotes más pequeños que en el modelo anterior. El primer lote que se va a realizar serán 5 conjuntos para el **cliente1**, que vendrá dado por las piezas que se pueden ver en la máquina **union** en la Figura 4.123. En la Figura 4.124 se puede apreciar cómo se han generado ya 4 piezas para el **cliente1**, siendo el conjunto que se está produciendo en ese momento el último conjunto del lote. Cabe destacar que, aunque se procesen los conjuntos por lotes, depositándose en los contenedores asociados a cada uno de los clientes, éstos no saldrán hacia los clientes hasta que no se encuentren en éstos las cantidades requeridas por cada cliente.

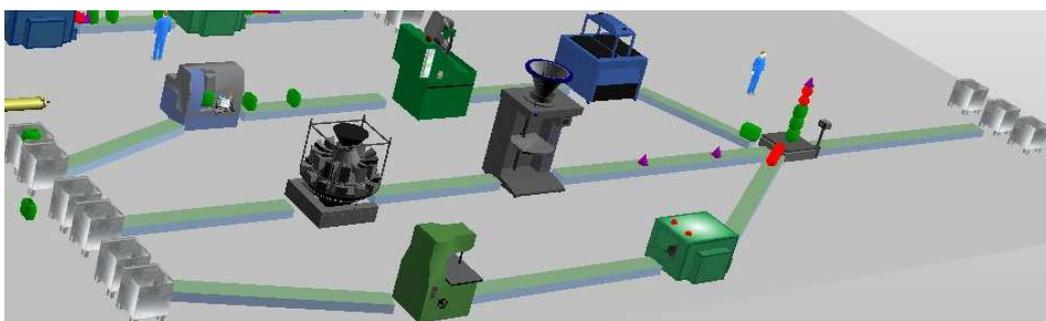


Figura 4.123: visualización del primer conjunto a realizar del cliente1, modelo 3D con heijunka y smed

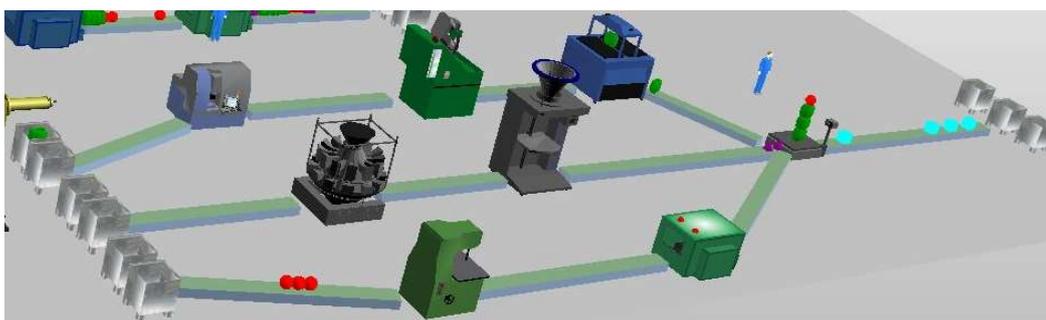


Figura 4.124: visualización de 4 conjuntos del primer lote del cliente1, modelo 3D con heijunka y smed

A continuación, se procesará el lote de 3 conjuntos para el **cliente2**, teniendo que ser necesario realizar una operación previa en la máquina **unión**, como se puede apreciar en la Figura 4.125, pudiéndose observar el primer conjunto que se genera, Figura 4.126. Además, en la Figura 4.127, se puede visualizar cómo se genera el último conjunto para este cliente, pudiéndose apreciar como en el contenedor **salida\_cliente1** continúan los 5 conjuntos para el **cliente1**.

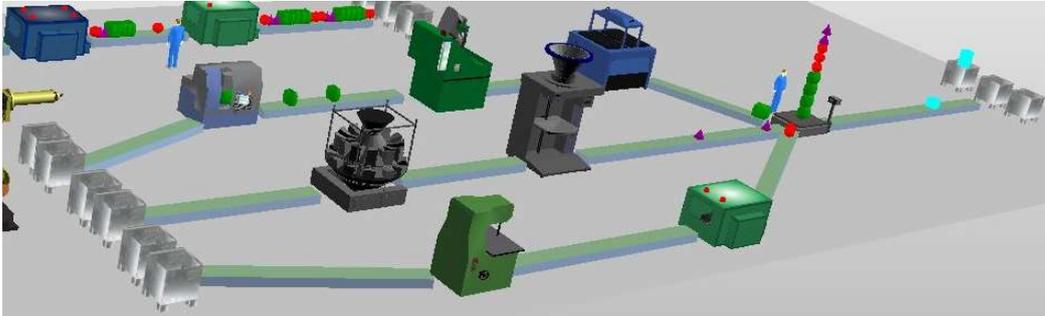


Figura 4.125: acción previa en la máquina unión para la realización del segundo lote instante 820.7, modelo 3D con heijunka y smed

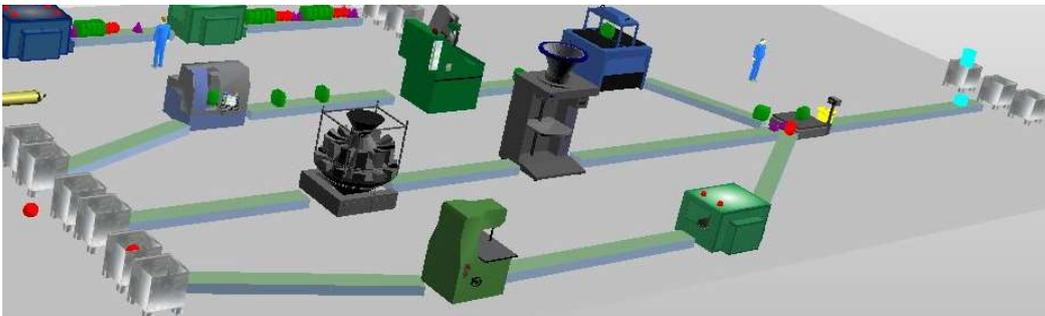


Figura 4.126: primer conjunto del primer lote cliente2 instante 856, modelo 3D con heijunka y smed

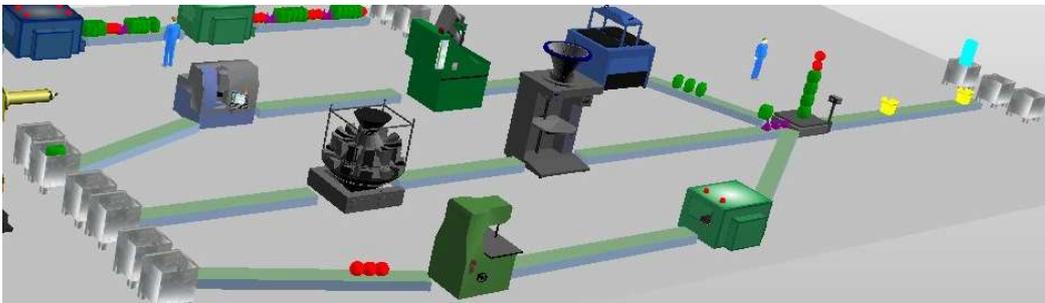


Figura 4.127: último conjunto para el cliente2 en el primer lote, modelo 3D con heijunka y smed

Para finalizar, el primer lote de conjuntos, se realizarán dos conjuntos para el **cliente3**. En la Figura 4.128 se aprecia cómo en la máquina **unión** ya se encuentran las piezas que van a formar el conjunto del **cliente3** y el **operario2** está realizando la actividad asociada al cambio de conjunto. En la Figura 4.129 ya se han realizado los dos conjuntos para el **cliente3**, y se ha generado un conjunto para el **cliente1**, iniciando de nuevo el ciclo que previamente descrito.

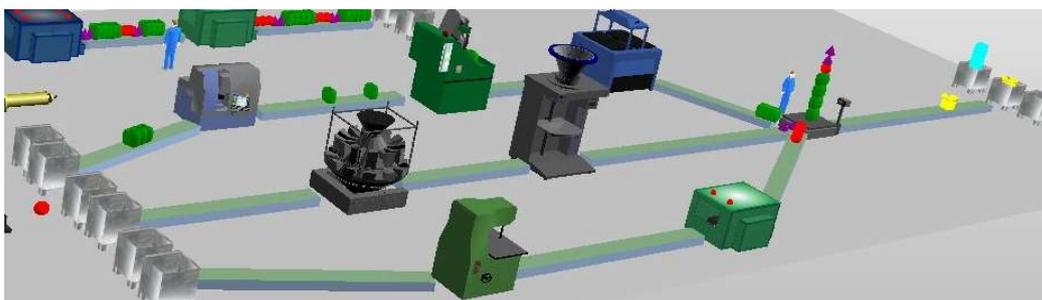


Figura 4.128: primer conjunto del primer lote cliente3, modelo 3D con heijunka y smed

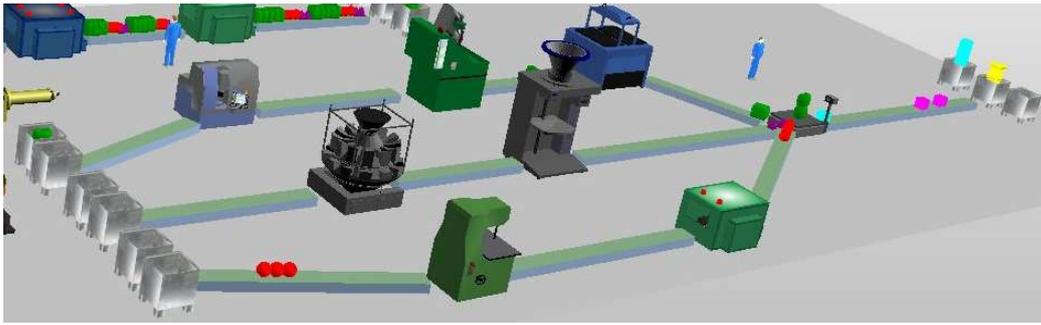


Figura 4.129: último conjunto para el cliente3 en el primer lote, modelo 3D con heijunka y smed

Una vez que se han descrito cómo se lleva a cabo el proceso, se mostrarán los tiempos que tardan en producirse ciertos conjuntos asociados a cada cliente. Comparando estos tiempos con los que se muestran en la Tabla 4.6, se puede apreciar que los lotes para cada uno de los clientes se producen antes en el modelo con Heijunka y Smed que en el caso sin las técnicas Lean.

	Cliente1	Cliente2	Cliente3
Primera pieza	422	855	1283
Pieza intermedia	681	1115	1308
Última pieza	1695	2088	2273

Tabla 4.6: tiempos en los que se producen piezas para cada cliente, modelo con heijunka y smed

## CONCLUSIONES Y LÍNEAS FUTURAS



Para finalizar este Trabajo Fin de Máster, se indicarán una serie de conclusiones en relación al trabajo desarrollado y, además, se propondrán algunas líneas de actuación futuras que permitan completar y ampliar dicho trabajo.

El objetivo principal que se marcaba para este trabajo Fin de Máster era favorecer un aprendizaje más práctico y visual de las técnicas Lean mediante la construcción de modelos 3D que permitiesen a los alumnos detectar fácilmente los problemas que se presentan en un proceso productivo y proponer soluciones Lean. A lo largo de este Trabajo Fin de Máster se han ido explicando una serie de modelos que se asemejan a procesos productivos y en los que no se utilizan técnicas asociadas a la filosofía Lean Manufacturing, pudiéndose visualizar el problema más significativo que se presentaba en cada uno de ellos; por ejemplo, en el modelo sin Kanban el excesivo stock que se genera, o en el proceso productivo lineal el elevado tiempo que se pierde en los desplazamientos.

Una vez detectado el problema, se ha indicado la técnica Lean Manufacturing que lo solucionaría (por ejemplo, generar un proceso productivo en célula en U o trabajar sólo cuando es necesario), y posteriormente se indicará cómo se implantaría la técnica Lean Manufacturing que solucionaría el problema detectado, pudiéndose visualizar cómo se comporta el nuevo modelo. De esta forma, se va a poder entender cómo es el funcionamiento de ciertas técnicas Lean Manufacturing de forma visual y sencilla.

Como se ha comentado en el documento, actualmente es muy importante que los alumnos conozcan los conceptos y las técnicas asociadas a la filosofía Lean Manufacturing, debido al auge que están teniendo en distintos sectores productivos. Con este Trabajo Fin de Máster se han podido conocer algunas de las técnicas Lean de una forma muy visual, alejándose de los conceptos teóricos sin ejemplos prácticos. De esta forma, se pueden visualizar los conceptos teóricos mediante ejemplos, como puede ser el concepto teórico en la técnica Kanban de la necesidad de transporte o fabricación sólo cuando es necesario, trabajando con tarjetas, o cómo se produce la flexibilidad de los productos generados, disminuyendo los tiempos de espera de los clientes al implementar técnicas Heijunka y Smed.

Visualizando los modelos en los que se han implementado técnicas Lean, se puede aprender fácilmente cuáles son sus principios teóricos. Además, se puede ver cómo funcionan las técnicas Lean explicadas, alejándonos de las explicaciones teóricas que no tienen ningún fundamento práctico, y que no permiten a los alumnos implantarlas correctamente en el momento en el que se necesita la aplicación de una de ellas.

Cabe destacar que algunas técnicas, como puede ser las 5S o el mantenimiento preventivo que se han explicado en el apartado Introducción, no se han podido llevar desarrollar en este Trabajo Fin de Máster mediante la visualización de los modelos correspondientes. Esto ha sido debido a que poseen unos conceptos muy teóricos, muy difíciles de conseguir en la simulación; por ejemplo, en la técnica 5S mantener el puesto de trabajo ordenado o sin innecesarios. Por esta razón, se ha decidido centrarse en aquellas técnicas que se podrían entender su funcionamiento mediante modelos, en los que se puedan visualizar las consecuencias de aplicar o no una técnica Lean, así como entender los principios de funcionamiento de dicha técnica. Aunque no se hayan llevado a la práctica estas técnicas, se han descrito brevemente en la Introducción, para que se tenga un acercamiento a ellas.

Como se ha comentado, este Trabajo Fin de Máster tiene un fin educacional, por lo que no se representan procesos productivos reales. Por este motivo, todas las características que se han descrito, como tiempos de ciclos, set-up, ..., no se corresponden con la realidad. Aun así, se ha querido realizar una comparación entre los resultados obtenidos en los modelos con técnicas Lean y sin ellas, pudiéndose apreciar que las técnicas Lean disminuyen considerablemente los niveles de stock, los tiempos asociados a desplazamientos, despilfarros generados por defectos de calidad, tiempos para abastecer a clientes, ..., pero siempre teniendo en cuenta que los resultados no son reales, dado que las características del modelo no se corresponden con procesos productivos reales.

En cuanto al conocimiento de programas de simulación, en el presente Trabajo Fin de Máster se ha podido comprobar cómo se comportan los procesos productivos sin tener que llevarlos a la práctica, y se han planteado las distintas alternativas que pueden mejorar procesos productivos y decidir cuál de ellas se adapta mejor al resultado que queremos obtener. Se ha podido comprobar cómo el programa Witness es una potente herramienta de simulación, en la que se puede llevar a cabo cualquier proceso productivo, tanto con elementos discretos, como los que se han ido describiendo en este Trabajo Fin de Máster, como con elementos continuos.

En el desarrollo de este Trabajo Fin de Máster se han podido visualizar cómo se programa en Witness. Se han programado las reglas de entrada y salida en las diferentes máquinas (dadas por reglas *pull*, *match*, *push*, ...), se ha definido la operativa de los distintos operarios, se han transformado piezas en las máquinas, se han modificado el comportamiento del modelo a través de acciones, etc.

Además de un gran programa de simulación en 2D, se pueden obtener los modelos en 3D, favoreciendo su comprensión; por ejemplo, mostrando cómo se comportan las distintas máquinas. Además, la visualización en 3D puede ser una herramienta muy importante a la hora de “vender” un proceso o querer defender una alternativa sobre otra, dado que visualmente se pueden apreciar las características de cada modelo.

Aunque los modelos se podían haber representado perfectamente en 2D, la simulación en 3D ha mejorado considerablemente el conocimiento de estas técnicas. Dicha representación tridimensional permite comprender mejor cómo se comportan los modelos y entender las técnicas que se han comentado: en el caso del modelo Heijunka, pudiéndose visualizar las piezas que componen las necesidades de cada cliente en la máquina unión; en el modelo Kanban, pudiendo apreciar cómo el operario traslada el *k\_fabricacion* de un buzón a otro, dejando o comenzado de fabricar las máquinas dependiendo del buzón donde se encuentre el kanban; también se puede apreciar cómo se realiza el transporte de piezas a la zona de acabado, únicamente cuando ha llegado alguno de los *k\_transporte* a la zona de fabricación. De esta forma, con esta herramienta se ha podido conseguir una mejor comprensión de las técnicas lean, de forma visual y amena, debido a que con el modelo en 2D las técnicas lean no se aprecian.

En cuanto a trabajos futuros, como se ha indicado a lo largo de este trabajo, la finalidad de éste es educacional. Por tanto, una vez que se han generado los distintos modelos, ya sea con o sin técnica Lean asociada, será necesario presentárselos a los alumnos, ya que ellos serán los beneficiarios de este trabajo. A éstos se les pedirá su opinión acerca de los modelos y cómo ha beneficiado su aprendizaje de las técnicas Lean Manufacturing. Realizando una síntesis de

sus valoraciones, se podrán mejorar los modelos en los aspectos que indiquen los alumnos para que su aprendizaje sea el óptimo.

Aunque los modelos que se han explicado a lo largo de este Trabajo Fin de Máster representen las técnicas Lean que se han querido implementar, siempre se pueden realizar algunas mejoras desde el punto de vista de la simulación en 3D, tratándose de una línea futura que mejore la visualización. Estas mejoras pueden ser la modificación de algunas imágenes en 3D, no sólo con las opciones de cambiar los colores como se realizó en este Trabajo Fin de Máster, sino generando nuevas imágenes para asociarlas a piezas o máquinas.

Una línea de actuación que sería muy interesante consistiría en asemejar los modelos descritos a procesos reales, pudiendo introducir tiempos de ciclos reales, probabilidad de averías de las máquinas, ..., para que los alumnos puedan conocer realmente como afectan las técnicas Lean Manufacturing a los procesos productivos reales y, de este modo, se conciencien de las ventajas de usar la filosofía Lean.

Además, una línea futura muy interesante sería la realización de un modelo que simule un proceso productivo en el que se puedan encontrar varios problemas que se podrían solucionar con distintas técnicas Lean de una forma gradual (algo similar a lo que ocurre en la Escuela Lean de la UVA). De esta forma, se tendrían que identificar cuál es la causa de cada uno de los problemas y sería necesario, en consecuencia, aplicar varias técnicas Lean Manufacturing. Se podría visualizar cómo se realizan mejoras en varios aspectos de un proceso productivo y cuál es el resultado final aplicando las máximas técnicas asociadas a la filosofía Lean Manufacturing. De esta forma, se asemejaría mejor a un proceso productivo, dado que en éstos no se aplica exclusivamente una técnica, sino varias.



## REFERENCIAS



CUATRECASAS, L. 2010. *Lean management: la gestión competitiva por excelencia*. Barcelona. Profit. ISBN: 978-84-96998-15-5

DE BENITO MARTÍN, J.J.; SÁNZ ANGULO, P. 2014. Simulación y evaluación de procesos. Universidad de Valladolid

DEMETER, K.; MATYUSZ, Z. The impact of lean practices on inventory turnover. *International Journal of production Economics*, **133(1)**, 154-163.

GARCÍA-ALCARAZ, J.L.; MALDONADO-MACÍAS, A.A.; CORTES-ROBLES, G. 2014. *Lean Manufacturing in the Developing World*. Springer. ISBN 978-3-319-04950-2

HERNÁNDEZ, J.C.; VIZÁN, A. 2013. *Lean Manufacturing. Conceptos, técnicas e implantación*. Escuela de organización Industrial. Madrid. ISBN 978-84-15061-40-3

PRIETO DIEGO, J. 2016. *Modulo I: Dirección de Operaciones*. Fundación General de la UNED.

ORTEGA MIER, M.; GARCÍA SÁNCHEZ, A. 2010. *Primeros pasos con Witness*. Universidad Politecnica de Madrid. ISBN: 974-2x-2323-23-23

SANZ P. 2015. *Justo a Tiempo. La filosofía Just in Time. Apuntes Dirección de la Producción*. Universidad de Valladolid.

SHINGO, SHIGEO. 2003. *Revolución en la producción: el Sistema SMED*. ISBN: 9788487022029

SUNDAR, R.; BALAJI, A.N.; KUMAR, R.M.S. 2014. A Review on Lean Manufacturing Implementation Techniques. *Procedia Engineering*, **2014 (97)**, 1875-1885.

WOMACK, J.P; JONES, D.T. 2003. *Lean thinking, Como usar el pensamiento Lean para eliminar los despilfarros y crear valor en la empresa*. Barcelona. Centro de libros PAPF. ISBN: 978-84-9875-199-4

WAHAB, A.N.A; MUKHTAR, M.; SULAIMAN, R. 2013. A Conceptual Model of Lean Manufacturing Dimensions. *Procedia Technology*, **2013(11)**, 1292-1298.

