

Universidad Miguel Hernández de Elche

**MASTER UNIVERSITARIO EN
ROBÓTICA**



**“Simulación proceso industrial utilizando la herramienta
para Matlab ARTE”**

**Trabajo de Fin de Máster
2017-2018**

**Autor: Vicente Cabanes Ribelles
Tutor: Arturo Gil Aparicio**

AGRADECIMIENTOS

En primer lugar, agradecer a todas las personas que han hecho posible mi formación y en especial poder cursar el máster en robótica en esta universidad.

Agradecer a toda mi familia su esfuerzo e implicación durante el transcurso del curso y en especial a mi madre fallecida durante el transcurso del mismo y a quien dedico todos mis logros y esfuerzo diario

También agradecer a todo el profesorado que ha formado parte de mi formación ya que en todos los casos han sabido transmitir los conceptos necesarios e involucrarme en el proceso de aprendizaje hecho que ha facilitado la tarea del aprendizaje.

En especial a D. Oscar Reinoso director del máster por su accesibilidad, toda la colaboración ofrecida y su dedicación a los alumnos en todo momento.



RESUMEN:

La robótica y la automática es una ciencia relativamente nueva i en continua evolución. Toda nueva tecnología tiene su nacimiento en el estudio, practica y análisis para la determinación de los factores y/o variables que la definen y a la vez permiten desarrollarla y mejorarla.

La simulación es un elemento indispensable en todo proceso automatizado ya que un correcto planteamiento permite obtener una ejecución perfecta de cualquier tipo de proceso industrial automatizado.

Para realizar una simulación completa primero se diseñan los elementos que la forman así como su colocación en el espacio (la celda). Posteriormente se simulan los movimientos tanto de material como de los robots i/o elementos móviles que forman la simulación.

Todo este proceso nos permite obtener planos finales de montaje y programas Off line que se pueden cargar directamente de la simulación al robot y con unos pocos ajustes poner en funcionamiento una celda robotizada.

Cada vez son más las empresas y centros tecnológicos que recurren a las simulaciones digitales para el desarrollo de nuevas técnicas de producción o la obtención de nuevos productos y servicios.

Este trabajo tiene como función principal demostrar los conocimientos adquiridos durante el curso con la obtención de un proyecto técnico de simulación de un proceso industrial totalmente automatizado.

Palabras clave: simulación, diseño, roboguide, fanuc, robots, Matlab, solidworks

ABSTRACT:

Robotics and automation are a relatively new science and one in continuous evolution. All new technology has its origin in the study, practice and analysis to determine the factors and/or variables that define it and at the same time allow it to develop and improve it.

The simulation is an indispensable element in any automated process since a correct approach allows to obtain a perfect execution of any type of automated industrial process.

To carry out a complete simulation, first the elements that form it are designed, as well as their placement in the space (the cell). After that, the movements the material, the robots and the mobile elements are simulated.

All this process allows us to obtain final assembly drawings and the famous "OLP" (Offline program) that can be loaded directly from the simulation to the robot and with a few adjustments to put into operation a robotic cell.

More and more companies and technology centres are turning to digital simulations to develop new production techniques or obtain new products and services.

The main function of this work is to demonstrate the knowledge acquired during the course by obtaining a technical Project to simulate a fully automated industrial process.

ÍNDICE GENERAL

BLOQUE 01 INTRODUCCIÓN Y DESARROLLO INICIAL

1- AGRADECIMIENTOS	Página 2
2- RESUMEN	Página 3
3- ABSTRACT	Página 4
4- ÍNDICE DE FIGURAS	Página 7
5- INTRODUCCIÓN	Página 10
6- ESTADO DEL ARTE	Página 12
7- DISEÑO ESTACIÓN PRODUCCIÓN	Página 13
7.1 MATERIAL DE ENTRADA	Página 13
7.2 HERRAMIENTAS DE TRABAJO	Página 14
7.3 MESA DE ENSAMBLAJE	Página 15
7.4 SALIDA PRODUCTO ACABADO	Página 16
7.5 CELDA PRODUCCIÓN	Página 16
7.6 ELEMENTOS ADICIONALES	Página 17
8- SIMULACIÓN MATLAB	Página 18
8.1 INICIAR LIBRERÍA	Página 19
8.2 CARGAR COMPONENTES	Página 19
8.3 POSICIONAR COMPONENTES	Página 21
8.4 CARGAR HERRAMIENTAS	Página 22
8.5 RECURSOS ADICIONALES	Página 22
8.6 CARGAR PIEZAS TRABAJO	Página 23
8.7 SIMULACIÓN MOVIMIENTOS	Página 24
8.7.1 MOVIMIENTOS PIEZAS	Página 24
8.7.2 MOVIMIENTOS ROBOT	Página 24
8.8 CICLO TRABAJO	Página 26
8.9 SIMULACIÓN OBTENIDA	Página 27

BLOQUE 02 SIMULACIÓN ROBOGUIDE

1- INTRODUCCIÓN PROGRAMA	Página 29
2- CREAR ROBOT 01	Página 32
3- CREAR ROBOT 02	Página 37
4- CARGAR ENTORNO SIMULACIÓN	Página 39
5- INSERTAR HERRAMIENTAS DE TRABAJO	Página 41
6- SIMULACIÓN MATERIAL ENTRADA	Página 44
7- ESTRUCTURA SIMULACIÓN	Página 47
8- DESARROLLO SIMULACIÓN	Página 48
8.1 PROCESO R01 – 01	Página 50
8.2 PROCESO R01 – 02	Página 52
8.3 PROCESO R01 – 03	Página 54
8.4 PROCESO R01 – 04	Página 56
8.5 PROCESO R02 - 05	Página 58
8.6 PROCESO R02 – 06	Página 61
8.7 PROCESO R02 – 07	Página 62
8.8 AJUSTAR CICLO	Página 63
9- PROTOTIPADO ESTACIÓN	Página 64

BLOQUE 03 BIBLIOGRAFIA Y ANEXOS

1- BIBLIOGRAFIA	Página 66
2- PROGRAMA OFFLINE MATLAB	Página 67
3- PROGRAMA OFFLINE ROBOT 01 ROBOGUIDE	Página 70
4- PROGRAMA OFFLINE ROBOT 02 ROBOGUIDE	Página 74
5- HOJA CARACTERÍSTICAS TÉCNICAS ROBOT	Página 77
6- PLANOS ELEMENTOS	Página 78

INDICE FIGURAS

Figura 01	Listón entrada	Página 13
Figura 02	Taco entrada	Página 13
Figura 03	Sistema entrada listones	Página 13
Figura 04	Sistema entrada tacos	Página 13
Figura 05	Pistola clavos	Página 14
Figura 06	Pistola grapas	Página 14
Figura 07	Accionamiento	Página 14
Figura 08	Herramienta robot01	Página 14
Figura 09	Herramienta robot 02	Página 14
Figura 10	Mesa ensamblaje vacía	Página 15
Figura 11	Mesa ensamblaje llena	Página 15
Figura 12	Cinta salida producto final	Página 16
Figura 13	Vallado seguridad	Página 16
Figura 14	Operario	Página 17
Figura 15	Cableado estación	Página 17
Figura 16	Ensamblaje final Solidworks	Página17
Figura 17	Librería robots	Página 19
Figura 18	Archivo .parameters robot	Página19
Figura 19	Código Matlab cargar robot	Página 20
Figura 20	Matrices rotación	Página 21
Figura 21	Ejemplos rotaciones múltiples	Página 21
Figura 22	Alzado herramienta 01	Página 22
Figura 23	Perfil herramienta 01	Página 22
Figura 24	Códigos colores RGB	Página 22
Figura 25	Vista planta entorno Matlab	Página 27
Figura 26	Vista perfil entorno Matlab	Página 27
Figura 27	Estructura teach pendant Fanuc	Página 29
Figura 28	Estructura entorno trabajo Fanuc	Página 30
Figura 29	Distribución ejes robot Fanuc	Página 30
Figura 30	Límite ejes robot Fanuc	Página 31
Figura 31	Robot Fanuc R-2000iC	Página 31
Figura 32	Icono programa Roboguide	Página 32
Figura 33 y 34	Selección nombre celda Roboguide	Página 32
Figura 35	Selección método creación Roboguide	Página 33
Figura 36	Selección software robot Roboguide	Página 33
Figura 37	Selección tecnología robot Roboguide	Página 34
Figura 38	Selección modelo robot Roboguide	Página 34
Figura 39	Selección ejes adicionales robot Roboguide	Página 35

Figura 40	Selección opciones extra Roboguide	Página 35
Figura 41	Resumen estación a cargar Roboguide	Página 36
Figura 42	Selección tipo flange a cargar Roboguide	Página 36
Figura 43	Robot 01 cargado Roboguide	Página 37
Figura 44	Creación robot 02 Roboguide	Página 37
Figura 45	Robot 01 y 02 cargados Roboguide	Página 38
Figura 46	Posicionamiento ambos robots	Página 38
Figura 47	Cargar elementos entorno Roboguide	Página 39
Figuras 48 a 52	Entorno simulación cargado Roboguide	Página 40
Figura 53	Insertar herramienta Roboguide	Página 41
Figuras 54 a 56	Diferentes herramientas robot 01	Página 42
Figuras 57 a 59	Diferentes herramientas robot 02	Página 43
Figura 60	Menú “cell browser” Roboguide	Página 44
Figura 61	Parámetros movimientos Roboguide	Página 45
Figuras 62 a 69	Movimientos entrada material Roboguide	Página 46
Figura 70	Estructura proceso simulación Roboguide	Página 47
Figura 71	Crear programa nuevo “teach pendant”	Página 48
Figuras 72 y 73	Nombre programa nuevo “teach pendant”	Página 48
Figuras 74 y 75	Añadir líneas al programa	Página 48
Figuras 76 a 79	Selección herramientas y frame a utilizar	Página 49
Figuras 80 y 81	Crear primer punto simulación	Página 49
Figura 82	Código creado	Página 50
Figuras 83 y 84	Volteo listones	Página 53
Figura 85	Colocación primeros elementos	Página 55
Figuras 86 y 87	Proceso final robot 01	Página 56
Figuras 88 a 93	Volteo mesa ensamblaje	Página 57
Figura 84	Entrada material robot 02	Página 58
Figuras 95 a 97	Ensamblaje capa final palé	Página 60
Figuras 98 y 99	Proceso fijación grapas	Página 61
Figuras 100 y 101	Movimientos salida producto final	Página 62
Figura 102	Impresión FDM	Página 64

BLOQUE 01 INTRODUCCIÓN Y DESARROLLO INICIAL

- Introducción
- Estado del arte
- Diseño estación producción
 - 1 MATERIAL ENTRADA
 - 2 HERRAMIENTAS DE TRABAJO
 - 3 MESA ENSAMBLAJE
 - 4 SALIDA PRODUCTO ACABADO
 - 5 CELDA PRODUCCIÓN
 - 6 ELEMENTOS ADICIONALES
- Simulación Matlab
 - 01 INICIAR LIBRERÍA
 - 02 CARGAR COMPONENTES
 - 03 POSICIONAR COMPONENTES
 - 04 CARGAR HERRAMIENTAS
 - 05 RECURSOS ADICIONALES
 - 06 CARGAR PIEZAS TRABAJO
 - 07 SIMULACIÓN DE MOVIMIENTOS
 - MOVIMIENTOS PIEZAS
 - MOVIMIENTOS ROBOTS
 - 08 CICLO TRABAJO
 - SIMULACIÓN OBTENIDA

INTRODUCCIÓN

La simulación es un paso esencial en el diseño e implementación de nuevos prototipos y algoritmos en el campo de la robótica. Las herramientas de simulación que ofrece el mercado son variadas y cada una presenta distintas funcionalidades.

Gracias a la evolución de la potencia de los computadores durante las últimas décadas, la simulación se ha convertido en una herramienta indispensable para el diseño de cualquier proceso productivo.

Una simulación tiene como finalidad conocer, analizar y evaluar el funcionamiento de un sistema sin necesidad de la construcción o de la experimentación con el sistema real, ya que esto podría resultar extremadamente caro o peligroso.

La simulación es crucial para el estudio del movimiento, prototipado rápido, diseño de controladores y validación en un ambiente virtual antes de la ejecución sobre un robot real. Aunque el número de herramientas para la simulación de robots ha aumentado en los últimos años, es muy difícil seleccionar la mejor herramienta para un proyecto determinado.

Recientes avances en las metodologías de simulación y la gran disponibilidad de software que actualmente existe en el mercado, han hecho que la técnica de simulación sea una de las herramientas más ampliamente usadas en el análisis de sistemas. Además de las razones antes mencionadas, un estudio de simulación es muy importante para la ingeniería de sistemas porque presenta las siguientes ventajas en el diseño de estos:

- A través de un estudio de simulación, se puede estudiar el efecto de cambios internos y externos del sistema, al hacer alteraciones en el modelo del sistema y observando los efectos de esas alteraciones en el comportamiento del sistema.
- Una observación detallada del sistema que se está simulando puede conducir a un mejor entendimiento del sistema y por consiguiente a sugerir estrategias que mejoren la operación y eficiencia del sistema.
- La simulación de sistemas complejos puede ayudar a entender mejor la operación del sistema, a detectar las variables más importantes que interactúan en el sistema y a entender mejor las interrelaciones entre estas variables.
- La técnica de simulación puede ser utilizada para experimentar con nuevas situaciones, sobre las cuales tiene poca o ninguna información. A través de esta experimentación se puede anticipar mejor a posibles resultados no previstos.

A diferencia de las ventajas mencionadas, la técnica de simulación presenta el problema de requerir equipo computacional y recursos humanos costosos. Además, generalmente se requiere bastante tiempo para que un modelo de simulación sea desarrollado y perfeccionado. Finalmente en algunas ocasiones la simulación puede llegar a ser imprecisa, sin poder medir el grado de error, aunque esto dependerá de las simplificaciones realizadas a la hora de diseñar el modelo del sistema.

Teniendo en cuenta lo expuesto anteriormente es fácil concluir que la simulación también es una herramienta muy útil en robótica y que está comenzando a extenderse. La construcción de un robot es cara y complicada, en ocasiones son difíciles de programar y su fase de desarrollo muy probablemente puede dañar considerablemente el robot, por esto se utiliza un robot virtual que opera en un mundo virtual. En este tipo de aplicaciones los errores cometidos en un entorno virtual durante la fase de desarrollo no provocan graves consecuencias.

A nivel de estudio la simulación es la herramienta perfecta para llevar a cabo proyectos sin la necesidad de grandes inversiones y/o emplear grandes recursos. Una buena simulación permite evitar futuros errores así como planificar la estructura del proyecto de forma independiente.

En éste proyecto vamos a utilizar las diferentes estructuras de simulación valorando las características de cada uno de ellos así como los posibles errores o mejoras.

Específicamente Solidworks se utilizará para el diseño y prototipado de los elementos que forman la simulación y posteriormente Matlab y Roboguide para evaluar los movimientos de los robots en la estación de trabajo y la obtención de programas offline que nos permitan llevar a cabo el proceso real a través de los resultados obtenidos en la simulación.

ESTADO DEL ARTE

Los palés de madera están formados por un conjunto de elementos de madera o derivados de la misma y agrupados en forma de plataforma. Se fabrican con la mínima altura posible con la finalidad de poder ser transportados más fácilmente con carretillas elevadoras, transpaletas y otros mecanismos elevadores a la vez que soportan mercancías.

Las principales aplicaciones de los palés de madera son; poder apilar mercancías ocupando poco volumen y con gran resistencia, facilitar el transporte de mercancías, su manipulación y fácil almacenamiento. En los últimos tiempos también se están utilizando pales reciclados para crear muebles y otros accesorios del hogar.

La madera aserrada es sin duda el material más usado para la construcción de palés de madera seguido por los tableros derivados de la madera, si bien existe la posibilidad de encontrarlos hechos de metal, plástico e incluso cartón.

En la norma UNE-EN 12249 "Madera aserrada utilizada en paletas. Desviaciones admisibles y dimensiones recomendadas." se encuentran las especificaciones de las dimensiones tanto normalizadas como habituales de la madera aserrada usada para las paletas:

- Grosor: desde 15 hasta 100 mm
- Anchura: desde 75 hasta 200 mm
- Longitudes recomendadas: 600, 800, 1.000 y 1.200 mm

Las partes de los palés de madera se suelen unir de forma mecánica, usando tanto clavos como grapas.

Construir paletas es un trabajo difícil y peligroso. Los robots se han utilizado en la fabricación durante muchos años. ¿Por qué no utilizarlos para armar pales, aumentando la productividad, la flexibilidad y las ganancias?

DISEÑO ESTACIÓN PRODUCCIÓN



1- MATERIAL ENTRADA

La primera parte del proceso de una simulación pasa siempre por el diseño de los elementos que la forman. Un buen diseño permite facilitar los procesos siguientes a la vez que permite visualizar el proceso productivo de forma intuitiva.

Para empezar el diseño primero hay que seleccionar qué proceso productivo queremos, es decir cuál va a ser el producto que vamos a producir, bien sea producto terminado o una parte del proceso de producción del mismo.

En este caso se selecciona la producción de pallets de madera ya que es un proceso muy utilizado a día de hoy en la industria actual y a la vez relativamente sencillo.

Para poner en marcha el proceso de diseño hay que tener en cuenta los elementos que van a formar parte del producto final así como las `plataformas necesarias para dar acceso al robot a los elementos de forma automatizada.

En este caso el producto final es el pallet y los elementos que lo forman son los listones y tacos que van a ser ensamblados.

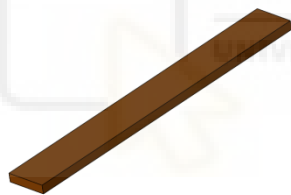


Figura 1 Listón entrada

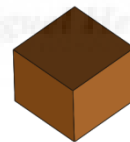


Figura 2 Taco entrada

Por otra parte, el diseño de las plataformas necesarias para automatizar el proceso van a ser las cintas y transportadores que almacenarán y a la vez distribuirán los elementos para dar acceso al robot cuando sea necesario en el proceso de producción.

Se diseña una plataforma de acceso de material para los listones y otra diferente para los tacos y se ensamblan los componentes a modo de almacén para dar autonomía a la celda de producción.

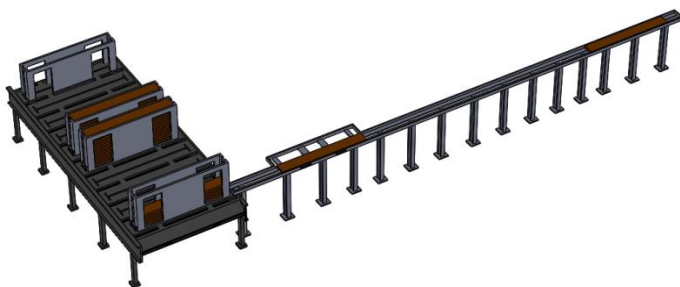


Figura 3 Sistema entrada listones

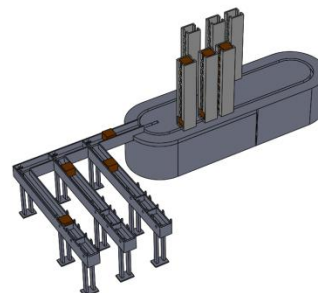


Figura 4 Sistema entrada tacos

2- HERRAMIENTAS DE TRABAJO

Una vez diseñados los elementos de entrada de material es momento ahora de diseñar las herramientas del robot ya que sin ellas el proceso productivo no será posible.

La idea es tener dos herramientas por cada robot, la primera de ellas para poder hacer “pick and place” (coger y dejar piezas) y la segunda para fijar o montar el pallet que utilizará clavos y grapas depende la fase de montaje en la que se encuentre.



Figura 5 Pistola clavos



Figura 6 Pistola grapas



Figura 7 Accionamiento

Una manera clásica de poder usar dos herramientas por cada robot se trata de diseñar las dos herramientas sobre la misma subestructura así podemos utilizar ambas sin la necesidad de recurrir a cambiadores de herramientas externos al robot con la complejidad que esto representa.

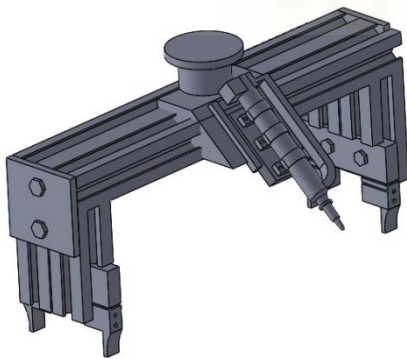


Figura 8 Herramienta robot 01

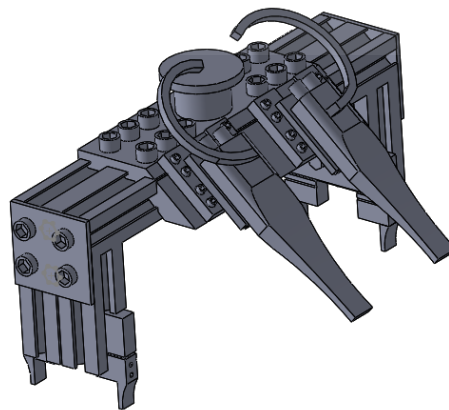


Figura 9 Herramienta robot 02

El diseño de estas herramientas (modelizado) cuenta con una pistola de clavos para la primera herramienta y dos pistolas de grapas para la segunda herramienta al mismo tiempo que el sistema de fijación neumático de las piezas.

3- MESA ENSAMBLAJE

La mesa de ensamble (“fixture”) es el elemento clave de esta simulación ya que es en este elemento donde se lleva a cabo la operación de ensamble de cada pale por parte de ambos robots.

Este elemento es común para los dos robots trabajando a dos caras para poder realizar dos procesos de ensamble al mismo tiempo y así obtener un tiempo de ciclo ajustado a la vez que competitivo en el mercado ya que la cantidad de unidades que somos capaces de producir por hora y por día marcarán la competitividad de nuestro producto en el mercado ya que repercuten directamente en el precio final del producto.

Para este proceso se elige una mesa de trabajo inclinada para facilitar la operación de place de los elementos sobre ella y a la vez rotativa para poder compartir el proceso de producción entre los dos robots.

El resultado obtenido es el siguiente:

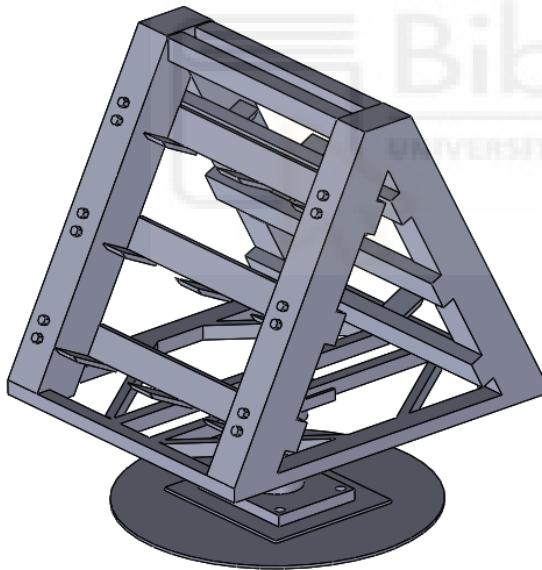


Figura 10 Mesa ensamble vacía

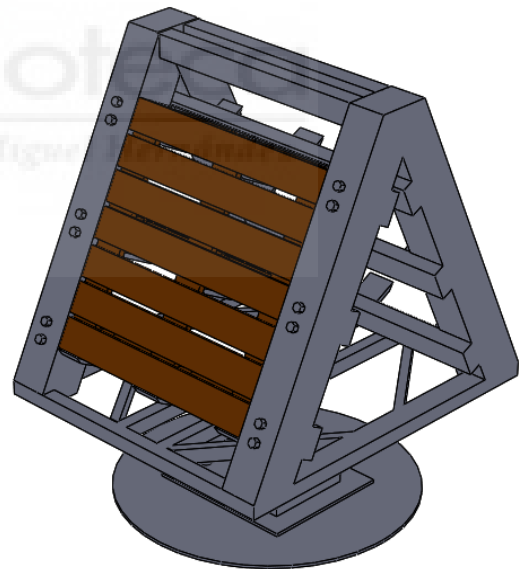


Figura 11 Mesa ensamble llena

Como se puede observar consta de dos caras de trabajo donde cada una de ellas llevará a cabo una parte del ensamble del pallet final ajustando el tiempo de trabajo de cada una de las dos caras para poder conseguir así un ciclo más o menos continuado y sin esperas por parte de ninguno de los dos robots.

Aunque no quedan reflejadas en el diseño inicial este fixture contará con un sistema de clampas neumáticas para fijar los componentes durante el proceso de ensamble y facilitar la tarea de los robots de pick and place sobre el fixture.

4- SALIDA PRODUCTO ACABADO

Una vez terminado el proceso de ensamblaje de los pallets el robot será el encargado de depositar los palés terminados apilados sobre una cinta transportadora que será la encargada de sacar los palés de la celda de producción de tal forma que un operario pueda acceder a ellos y almacenarlos para su posterior venta y/o utilización.

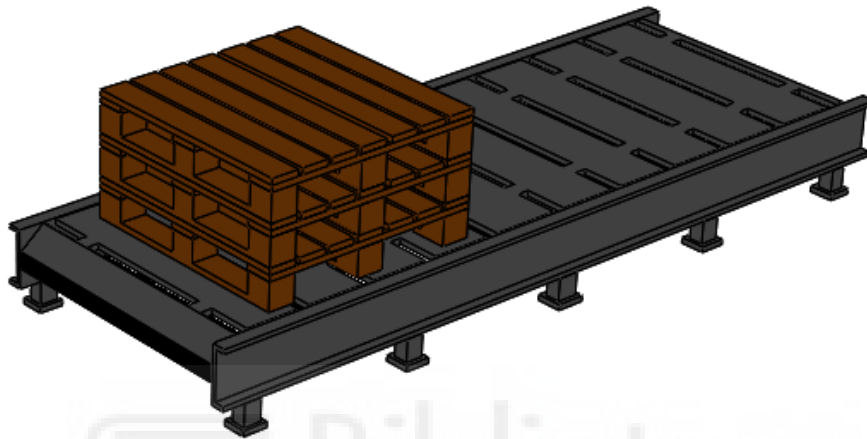


Figura 12 Cinta salida producto final

5- CELDA PRODUCCIÓN

Se diseñan barreras de protección (“fence”) que van a cerrar el entorno de trabajo de los robots por seguridad para los operarios así como las puertas de acceso.

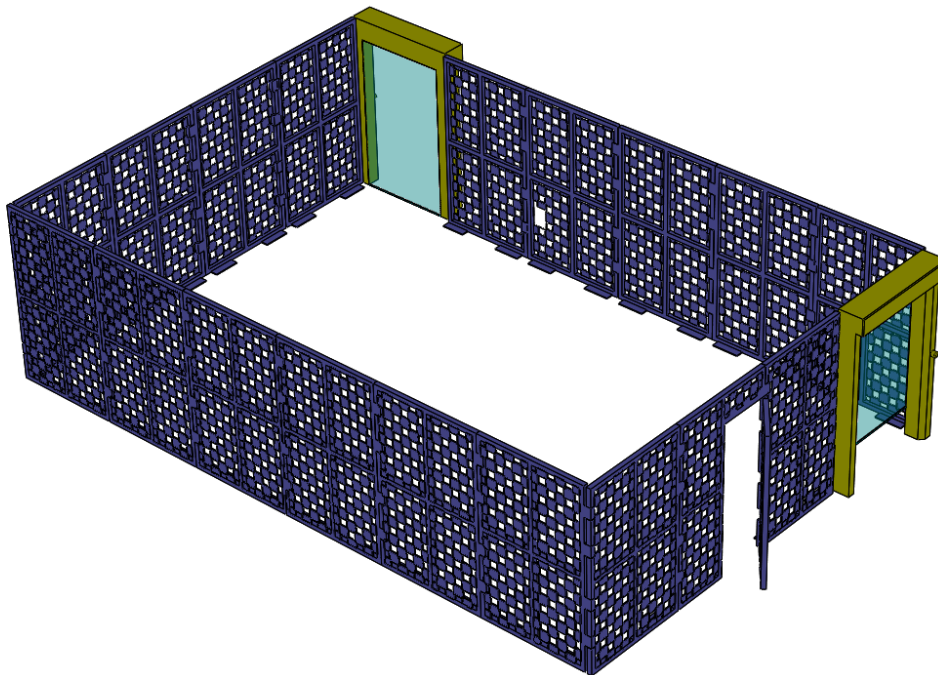


Figura 13 Vallado seguridad

6- ELEMENTOS ADICIONALES

Se diseñan también elementos adicionales como pueden ser los armarios controladores de los robots, operarios y todo el sistema de cableado de los diferentes elementos que forman la estación de trabajo del robot

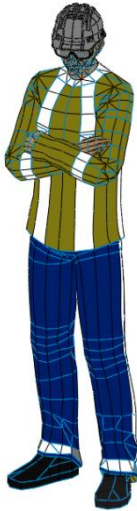


Figura 14 Operario

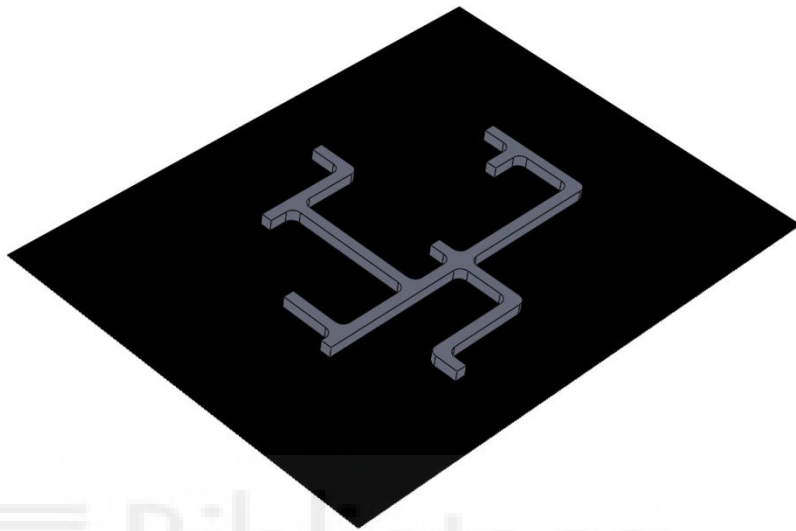


Figura 15 Cableado estación

El resultado final todo ensamblado queda de la siguiente forma:

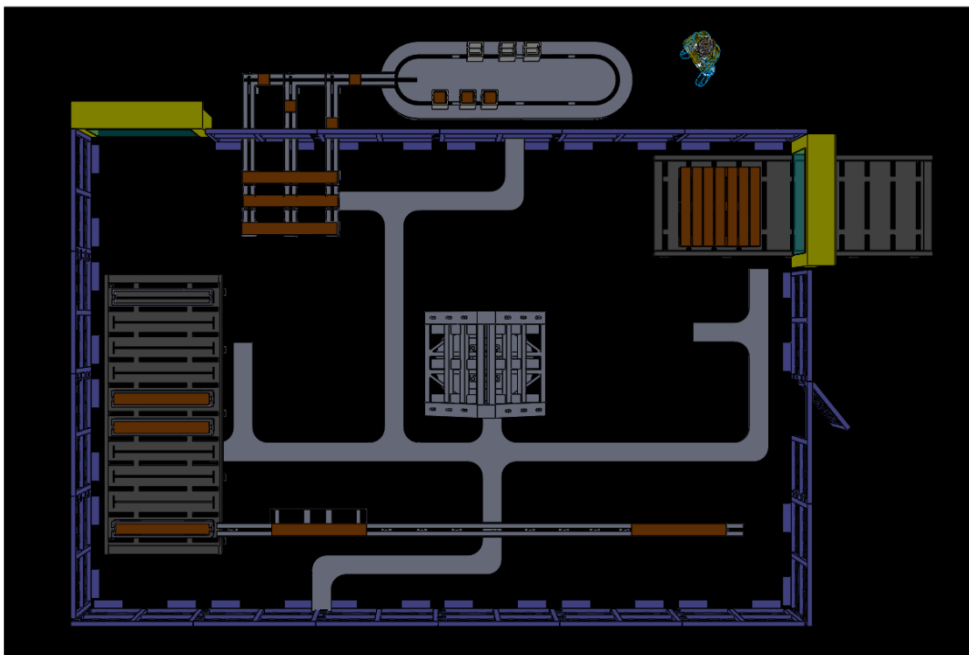


Figura 16 Ensamblaje final Solidworks



SIMULACIÓN MATLAB

INTRODUCCIÓN

Se inicia la simulación usando Matlab y la toolbox diseñado por D. Arturo Gil Aparicio llamada Arte (A robotic toolbox for Education)

ARTE es una toolbox de Matlab centrada en manipuladores robóticos, incluidos mecanismos paralelos.

Las principales características de ARTE son:

- Puede simular cualquier robot industrial dentro de un entorno Matlab.
- Representar a visualizar el sistema de Denavit-Hartenberg en el robot.
- Están disponibles gráficos 3D de una gran cantidad de robots industriales.
- Puede trazar y observar la posición, la velocidad y la aceleración de las coordenadas conjuntas del robot cuando se realiza un movimiento.
- Los robots pueden programarse y simularse el lenguaje ABB RAPID.
- Se puede llevar a cabo una simulación paso a paso de los programas dentro del editor y depurador de Matlab.
- Se incluye un intérprete de Matlab a RAPID que permite programar el robot real utilizando su código de Matlab.
- Los nuevos modelos de robots se pueden incluir fácilmente.
- Se proporcionan sesiones prácticas.

Se establece el procedimiento a seguir cargando primero todos los elementos que forman la estación de simulación, cargando las herramientas de los robots y posteriormente iniciando los movimientos de material de entrada y los primeros pick and place.

Para poder entender el funcionamiento de la herramienta de Arte se utilizan las metodologías impartidas a lo largo de curso en la asignatura de cinemática de robots y dinámica de robots impartida por D: Arturo Gil Aparicio así como los videos del canal de la asignatura y otros materiales proporcionados por el mismo.

A partir del material obtenido se establece una estructura de trabajo para poder cargar piezas y crear el entorno necesario para poder dar lugar a la simulación de producción de pales Hay que tener en cuenta que al ser una herramienta puramente parametrizada no cuenta con las facilidades en cuanto al entorno de trabajo de otras aplicaciones para la simulación hecho que va a ralentizar el proceso ya que es más dificultoso cargar todo el entorno de forma correcta.

1- INICIAR LIBRERÍA

En primer lugar se debe tener instalada la aplicación de Matlab y descargar previamente la librería en formato .rar o .zip desde la página web oficial de la toolbox.

Una vez descargado el material necesario arrancamos la aplicación de Matlab y seleccionamos la carpeta descargada en el apartado de “current folder” del lateral izquierdo (lo mejor es tenerla en el escritorio) para hacer click derecho con el ratón y seleccionar la opción “ADD TO PATH selected Folders and Subfolders”.

Una vez hecho esto hay que ejecutar el comando en la ventana llamada “command window” que ejecuta y carga la librería llamado “init_lib”.

Se puede cargar un robot ahora escribiendo los comandos necesarios en el command window o bien creando un programa que contenga dichas líneas de código y que realizará el mismo proceso al ejecutarlo.

El comando necesario para poder cargar un robot en el entorno 3d de Matlab es el siguiente: “robot=load_robot” seleccionando en la carpeta de la librería llamada robots la marca deseada y el modelo deseado (siempre cargamos el .parameter).

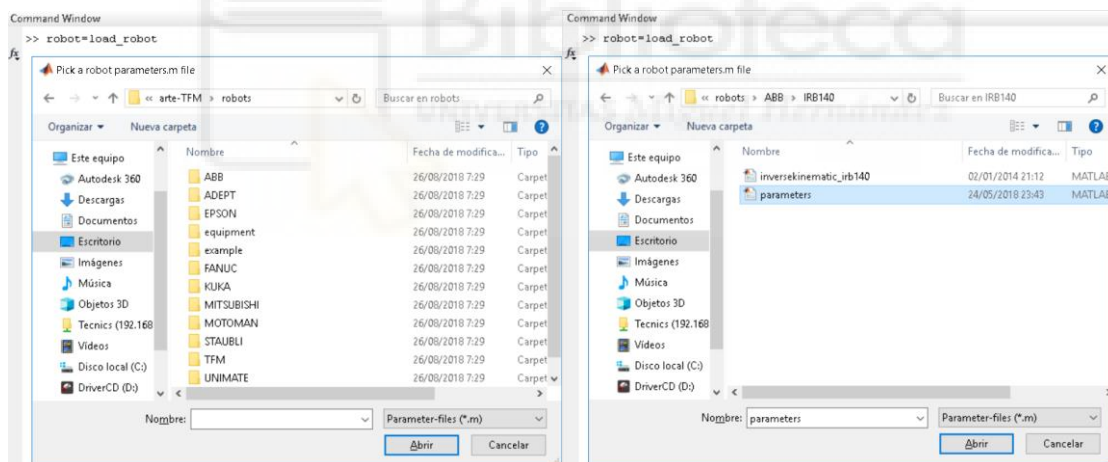


Figura 17 Librería robots

Figura 18 Archivo .parameters del robot

2- CARGAR COMPONENTES

Una vez cargado el robot procedemos a entender cuál es el procedimiento a seguir para poder cargar otros elementos en el mismo entorno de trabajo del robot.

Hay que tener las piezas o componentes a cargar en formato .stl obtenido a través de cualquier programa de diseño 3d y siempre con el mismo nombre (link0_base).

Además del formato .stl en algunos programas como puede ser solidworks se puede seleccionar opciones extra al crear el fichero que nos facilitarán el procesamiento de la toolbox Arte, concretamente seleccionando tipo ASCII, resolución rápida y no convertir a espacio positivo el fichero obtenido.

Una vez los ficheros preparados para cargar se debe crear una carpeta para cada uno de ellos que tiene que contener siempre los mismos elementos en su interior.

1. El archivo .stl obtenido anteriormente con nombre link0_base.
2. Un archivo .parameters de cualquier otro componente e incluso de un robot el cuál debe tener la matriz de “Denavit Hunttemberg” con valores a cero.

Esta carpeta (llamada pieza01) la podemos colocar dentro de la ruta Arte/robots/equipment o crear la carpeta dentro de la ruta Arte/robots.

Una vez las carpetas creadas con los archivos correspondientes en el interior de éstas es momento de ejecutar la función que crea el elemento o link en lenguaje Matlab.

Esta función llamará a la carpeta creada anteriormente de la siguiente forma:

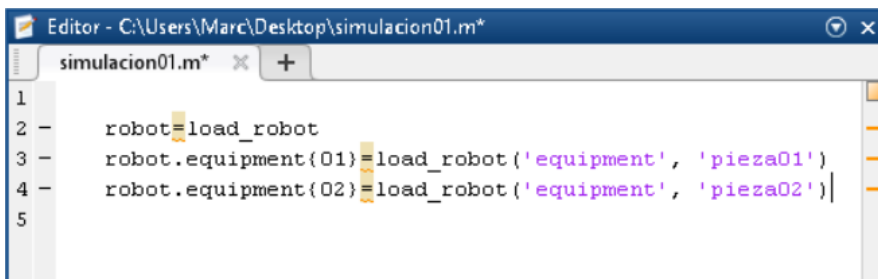
```
“transform_to_own('equipment','pieza01,500)”
```

Al variar el valor que acompaña a dicha función establecemos diferente nivel de zoom o escala del elemento que vamos a crear teniendo en cuenta que siempre vamos a utilizar el mismo número (1000) para mantener la misma escala en todos los elementos cargados ya que la escala la establecemos en el diseño inicial en solidworks.

Una vez ejecutada la función se crea dentro de la carpeta un elemento llamado link 0 que contiene la información necesaria para ser cargado en Matlab utilizando la siguiente función:

```
robot.equipment{01}=load_robot('equipment', 'pieza01')  
robot.equipment{02}=load_robot('equipment', 'pieza02')
```

Para cargar varias de ellas sólo se debe variar el número que acompaña a la función igual que en el ejemplo anterior y utilizar siempre el mismo método utilizado para la primera de ellas. El programa creado contiene las funciones necesarias tanto para cargar el robot inicial de la simulación, así como cada uno de los elementos que se carguen en el entorno de trabajo.



```
Editor - C:\Users\Marc\Desktop\simulacion01.m*  
simulacion01.m* x +  
1  
2 - robot=load_robot  
3 - robot.equipment{01}=load_robot('equipment', 'pieza01')  
4 - robot.equipment{02}=load_robot('equipment', 'pieza02')  
5
```

Figura 19 Código Matlab cargar robot

3- POSICIONAR COMPONENTES

Según lo explicado en la introducción este proceso es costoso ya que para modificar la posición y/o orientación de cada uno de los elementos cargados debemos editar la matriz de rotación y traslación que contiene dentro del archivo .parameters cada una de las piezas cargadas.

La matriz de rotación – traslación es una matriz formada por 4x4 elementos. La última componente de las tres primeras filas expresa la posición en coordenadas cartesianas [x y z] de la pieza respecto al origen de coordenadas del espacio 3d creado por Matlab. Editando estas componentes por lo tanto variamos la posición de la pieza en el entorno al cargarla.

Para las orientaciones se utilizan las matrices de rotación establecidas para cada uno de los ejes aplicando el valor del ángulo en las posiciones donde aparezcan senos o cosenos y calculando el valor obtenido para cada uno de los elementos que la forman.

ROTACIONES EJE Z	ROTACIONES EJE Y	ROTACIONES EJE X
$\begin{bmatrix} \cos & -\text{sen} & 0 & 0 \\ \text{sen} & \cos & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} \cos & 0 & \text{sen} & 0 \\ 0 & 1 & 0 & 0 \\ -\text{sen} & 0 & \cos & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos & -\text{sen} & 0 \\ 0 & \text{sen} & \cos & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$

Figura 20 Matrices de rotación

Al combinar varias rotaciones éstas matrices se operan entre ellas para obtener el resultado final a colocar en el archivo .parameters. Éstos son algunos ejemplos de las rotaciones que se han realizado:

%Ejemplo rotación multiple +90 grados eje z + 90 eje y

$$\begin{bmatrix} 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

%Ejemplo rotación multiple -90 grados eje z + 90 eje y

$$\begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix} * \begin{bmatrix} 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix}$$

%Ejemplo rotación multiple 90 eje z -90 grados eje x

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix} * \begin{bmatrix} 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix}$$

Figura 21 Ejemplos rotaciones múltiples

4- CARGAR HERRAMIENTAS

Para cargar la herramienta del robot basta con crear una carpeta igual que con los elementos extras con el link0_base en formato .stl y un archivo .parameters con su Denavit con valores cero, pero el origen de coordenadas de la herramienta a cargar debe estar colocado en el flange del robot, es decir en la cara de la herramienta que va a quedar en contacto con el robot físicamente

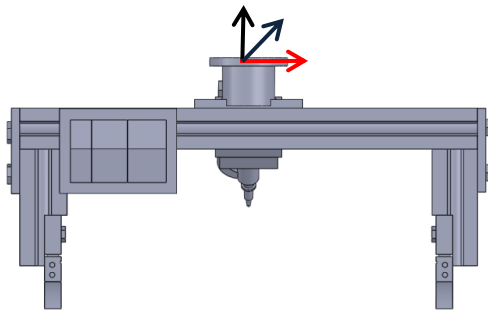


Figura 22 Alzado herramienta 01

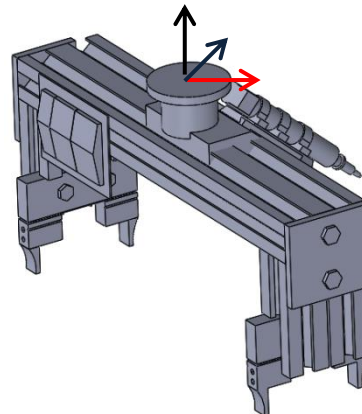


Figura 23 Perfil herramienta 01

Una vez con la herramienta con el origen de coordenadas en su posición y creada en lenguaje Matlab (usando la función “transform_to_own”) podemos cargarla sobre el robot fácilmente usando la función correspondiente:

```
robot.tool= load_robot('equipment','herramienta01');
```

Posteriormente definimos el punto TCP de la herramienta en el archivo .parameters correspondiente a ésta y en el apartado donde aparece la matriz de 4x4 llamada robot.TCP . Para este tipo de herramienta el TCP queda ubicado en el eje z por lo que sólo debemos darle valor a la última componente de la fila 3 que representa el valor de la coordenada z del TCP siempre en respecto al flange del robot (el último eje).

5- RECURSOS ADICIONALES

Como recurso adicional a la simulación se utiliza el color de los elementos cargados pudiéndose variar el color de cualquier elemento en toda la gama RGB siguiendo el código de colores e introduciendo los valores en el .parameters de cada elemento.

```
robot.graphical.color = [250 255 0]./255;
```

Nombre HTML	Código HEX R G B	Código Decimal R G B	Nombre HTML	Código HEX R G B	Código Decimal R G B	Nombre HTML	Código HEX R G B	Código Decimal R G B
Colores Rojos			Colores Verdes			Colores Marrones		
IndianRed	CD 5C 5C	205 92 92	GreenYellow	AD FF 2F	173 255 47	Cornsilk	FF F8 DC	255 248 220
LightCoral	F0 80 80	240 128 128	Chartreuse	7F FF 00	127 255 0	BlanchedAlmond	FF EB CD	255 235 205
Salmon	FA 80 72	250 128 114	LawnGreen	7C FC 00	124 252 0	Bisque	FF E4 C4	255 228 196
DarkSalmon	E9 96 7A	233 150 122	Lime	00 FF 00	0 255 0	NavajoWhite	FF DE AD	255 222 173
LightSalmon	FF A0 7A	255 160 122	LimeGreen	32 CD 32	50 205 50	Wheat	F5 DE B3	245 222 179
Crimson	DC 14 3C	220 20 60	PaleGreen	98 FB 98	152 251 152	BurlyWood	DE B8 87	222 184 135
Red	FF 00 00	255 0 0	LightGreen	90 EE 90	144 238 144	Tan	D2 B4 8C	210 180 140
FireBrick	B2 22 22	178 34 34	MediumSpringGreen	00 FA 9A	0 250 154	RosyBrown	BC 8F 8F	188 143 143

Figura 24 Códigos colores RGB

6- CARGAR PIEZAS DE TRABAJO

Para cargar las piezas de trabajo realizamos el mismo procedimiento que para el entorno creando una carpeta con el archivo .stl guardado anteriormente con el nombre link0_base y el archivo .parameters con el Denavit con valores cero. Posteriormente ejecutamos la función transform_to_own y para finalizar el proceso cargamos la pieza utilizando la función determinada para ello:

```
robot.piece=load_robot('equipment','pieza01');
```

Para poder realizar la correcta simulación necesitamos cargar más piezas, pero Arte está definido para utilizar sólo esta función así que se necesita crear una nueva función para poder cargar más de una pieza.

Se modifica el código de ARTE para poder cargar diferentes piezas de la siguiente forma:

```
robot.piece{1}=load_robot('equipment','cylinders/cylinder_tiny');
```

```
robot.piece{2}=load_robot('equipment','cylinders/cylinder_tiny2');
```

```
robot.piece{3}=load_robot('equipment','cylinders/cylinder_tiny3');
```

Luego, la función que representa el robot gráficamente hay que modificarla también ya que esta función es la que permite representar las piezas en el escenario una vez cargadas. La función “drawrobot3d.m”

```
if isfield(robot, 'piece')
    if isfield(robot.piece, 'graphical')
        for i=1:length(robot.piece),
            if robot.piece.graphical.has_graphics
                if isfield(robot, 'tool')
                    if robot.tool.piece_gripped==1
                        %update the last known position and orientation of the robot
                        %Trel is computed when the simulation_grip_piece function is
                        %executed. Trel is the relative position and orientation of
                        %the tool and the piece that assures that the piece is
                        %picked at a constant and visually effective orientation.
                        draw_link(robot.piece, 1, T07*(robot.tool.Trel));
                    else
                        draw_link(robot.piece, 1, robot.piece.T0);
                    end
                end
            end
        end
    else
        draw_link(robot.piece, 1, robot.piece.T0);
    end
    if robot.piece.graphical.draw_axes
        draw_axes(robot.piece.T0, sprintf('X_{piece%d}',0), sprintf('Y_{piece%d}',0), sprintf('Z_{piece%d}',0));
    end
end
end
end
```

En el código anterior se debe indicar un bucle for para pintar todas las piezas que hallas guardado anteriormente en piece{1}, piece{2}... etc.

7- SIMULACIÓN DE MOVIMIENTOS

7.1 MOVIMIENTOS PIEZAS

Para realizar la simulación de movimientos de piezas se utiliza la matriz de posición y orientación del .parameters pero esta vez actualizando uno de sus valores a través de un bucle for que variará la cota en cada repetición.

La estructura a utilizar es la siguiente:

```
for i=1:12
robot.piece.T0(2,4)=robot.piece.T0(2,4)-i*0.021
drawrobot3d(robot, q)
end
```

Este bucle actualiza el elemento de la fila dos y la columna cuatro.

La distancia en cada recorrido es el valor 0.021.

El número de veces que actualizará este valor es el i=1:12 (doce veces).

Este mismo método lo podemos utilizar para realizar movimientos en cada uno de los ejes deseados y podemos controlar la velocidad de cada uno de los movimientos solamente añadiendo la función `pause(0.002)`; dentro del mismo bucle y jugando con los valores del pause.

```
%MOVIMIENTO LISTON 01
for i=1:12,
robot.piece.T0 (2,4)=robot.piece.T0 (2,4)-i*0.021;
drawrobot3d(robot, q)
end
```

MOVIMIENTOS EJE Y

```
for i=1:6,
robot.piece.T0 (1,4)=robot.piece.T0 (1,4)+i*0.007
drawrobot3d(robot, q)
end
```

MOVIMIENTOS EJE X

7.2 MOVIMIENTOS ROBOT

Para crear los movimientos de la simulación hay que utilizar el controlador del robot para mover el robot al punto deseado llamando a la función "teach" y utilizando el movimiento por ejes (j1, j2, j3, j4, j5, j6) o lineal (X+,X-,Y+,Y-,Z+,Z-). Una vez posicionado el robot en la configuración deseada presionamos "save current target" y se creará el primer target. Este proceso se repite posición a posición hasta crear la trayectoria deseada. Una vez las posiciones guardadas hay que seleccionar la opción de "save TPs to file" y guardamos un archivo con los targets creados que contienen las posiciones de cada uno de los puntos.

Cada uno de los targets creados se expresan en lenguaje rapid y almacena todos los datos necesarios para el robot siguiendo la siguiente estructura:

```
RT_tp1=[[TRUE],[[0,0,0],[1,0,0,0]],[5,[9,0,9],[1,0,0,0],0.01,0.01,0.01]];
```

1. TRUE: Corresponde al campo "robhold" e indica si el robot sujeta la herramienta o no.
2. El primer vector [0,0,0] corresponde a la posición del punto origen de la herramienta respecto al extremo del robot (flange) en mm.
3. El segundo vector [1,0,0,0] corresponde a la orientación de la herramienta respecto al extremo del robot (flange) en cuaternos.
4. El resto de datos expresan la masa y los momentos de inercia de la herramienta del robot.

4.1 El primer valor 5 es el peso de la herramienta en kilogramos.

4.2 El primer vector [9,0,9] indica la posición del centro de gravedad.

4.3 El segundo vector [1,0,0,0] indica la orientación del sistema de coordenadas sobre el que se aplican los momentos de inercia

4.4 Los tres restantes valores 0.01,0.01,0.01 representan los momentos de inercia mencionados anteriormente en Kg/m².

Una vez creados los movimientos del robot utilizamos la función que nos permite juntar las piezas a las herramientas con el fin de simular el proceso de cogida de pieza (pick)

La función "simulation_grip_piece" es la que permite mover el objeto que deseamos hacer pick conjuntamente con el tcp de la herramienta a fin de simular el movimiento conjunto de ambos elementos.

Utilizando los recursos anteriores conseguimos realizar todas las rutinas del robot01 y es el momento de realizar las rutinas del robot 02.

De la misma forma se crean los movimientos de alimentación de listones y los targets con las posiciones articulares de pick and place de los elementos necesarios para terminar el ensamblaje del pale y depositar el producto final sobre la cinta de salida.

8- CICLO TRABAJO

Una vez creados los dos ciclos de trabajo de ambos robots es el momento de realizar el ciclo de trabajo de la celda de producción reproduciendo los movimientos de ambos robots conjuntamente para poder establecer los tiempos de producción de cada uno de los dos robots ya que en todas las estaciones de trabajo formadas por varios robots el tiempo de producción lo define el robot más lento de la celda.

El estudio de la viabilidad del ciclo depende directa y proporcionalmente de los tiempos productivos obtenidos en el diseño del mismo ya que para poder rentabilizar el costo de una celda de producción la productividad debe ser lo más alta posible.

Cuando los diferentes robots que componen una celda de producción tienen tiempos de trabajo diferentes el ciclo siempre tendrá componentes esperando con el costo que implica tener maquinas o herramientas paradas en una nueva celda productiva que debe cumplir los planes de rentabilidad determinados anteriormente.

Para resolver las diferencias de tiempos de producción hay que estudiar proceso por proceso y dividir los procesos obteniéndose tiempos relativamente similares entre los diferentes integrantes de la celda de producción

El estudio de esta parte de la simulación se realiza a partir de segundo ciclo de trabajo del robot 01 ya que el primer ciclo (ciclo de llenado) es realizado por el robot 01 y una vez realizada su labor puede dar inicio el ciclo de trabajo del robot02.

Al iniciar el ciclo el robot 02 el robot 01 vuelve a iniciar su ciclo, pero la simulación en Matlab se realiza línea a línea y aunque se pueden realizar saltos en la programación y repeticiones resulta extremadamente complicado poder ejecutar dos códigos de forma paralela para dar lugar a la reproducción de los movimientos de ambos robots simultáneamente.

Es en este momento cuando se decide la viabilidad de la modificación de esta herramienta para poder dar resultado a la simulación completa o trasladar todos los datos a otro simulador para intentar reproducir la simulación simultánea de movimientos de ambos robots.

Determinar una solución al problema pasa por decidir que herramienta de simulación podíamos utilizar para finalizar el ciclo y poder dar por concluida la simulación completa del ciclo productivo

9- SIMULACIÓN OBTENIDA MATLAB

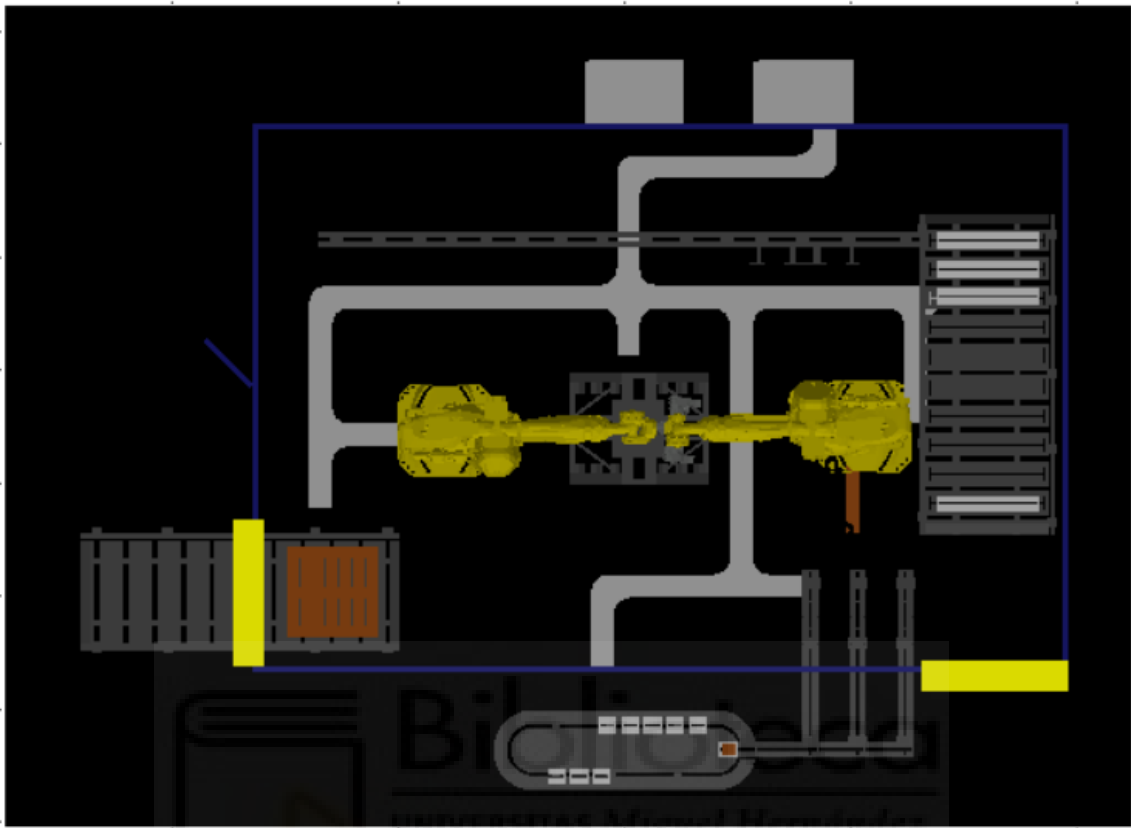


Figura 25 Vista planta entrono cargado en Matlab

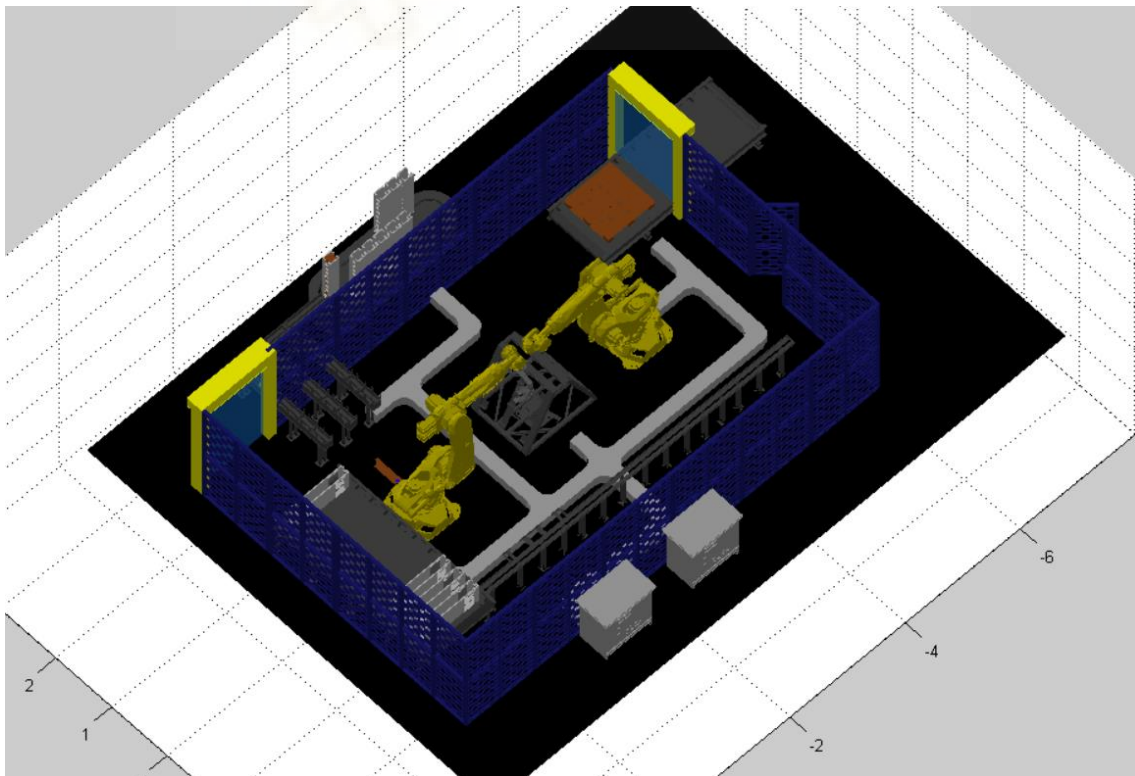


Figura 26 Vista perfil entrono cargado Matlab

BLOQUE 02 SIMULACIÓN ROBOGUIDE

- 1 INTRODUCCIÓN PROGRAMA
- 2 CREAR ROBOT 1
- 3 CREAR ROBOT 2
- 4 CARGAR ENTORNO SIMULACIÓN
- 5 INSERTAR HERRAMIENTAS TRABAJO
- 6 SIMULACIÓN MATERIAL ENTRADA
- 7 ESTRUCTURA SIMULACIÓN
- 8 DESARROLLO SIMULACIÓN
 - i) PROCESO r1 – 01 (PICK AND PLACE BASE PALET)
 - ii) PROCESO r1 – 02 (UNIÓN BASE PALLET CLAVOS)
 - iii) PROCESO r1 – 03 (PICK AND PLACE BASE PALLET – FIXTURE)
 - iv) PROCESO r1 – 04 (PICK AND PLACE CAPA INTERMEDIA – FIXTURE)
 - v) PROCESO r2 – 05 (PICK AND PLACE CAPA FINAL – FIXTURE)
 - vi) PROCESO r2 – 06 (UNIÓN CAPA INTERMEDIA Y FINAL GRAPAS)
 - vii) PROCESO r2 – 07 (PICK AND PLACE PRODUCTO FINAL)
 - viii) AJUSTAR CICLO
- 9 PROTOTIPADO ESTACIÓN

SIMULACIÓN ROBOGUIDE



1- INTRODUCCION PROGRAMA ROBOGUIDE

FANUC ROBOGUIDE es un simulador de robots que simula tanto el movimiento del propio robot como los comandos del programa. De este modo, reduce en gran medida el tiempo que se tarda en crear nuevas configuraciones de movimiento. Para garantizar un impacto mínimo en la producción, se pueden diseñar, probar y modificar las células offline. Para ahorrar tiempo de modelado 3D, es posible importar los modelos de las piezas de un PC como datos de CAD. La amplia librería del software de simulación de robots permite asimismo al usuario seleccionar y modificar las piezas y las dimensiones de acuerdo con sus requisitos. Gracias a su diseño intuitivo y su gran facilidad de uso, ROBOGUIDE no requiere prácticamente formación. También está disponible con herramientas especializadas para aplicaciones concretas.

INTRODUCCION TEACH PENDANT

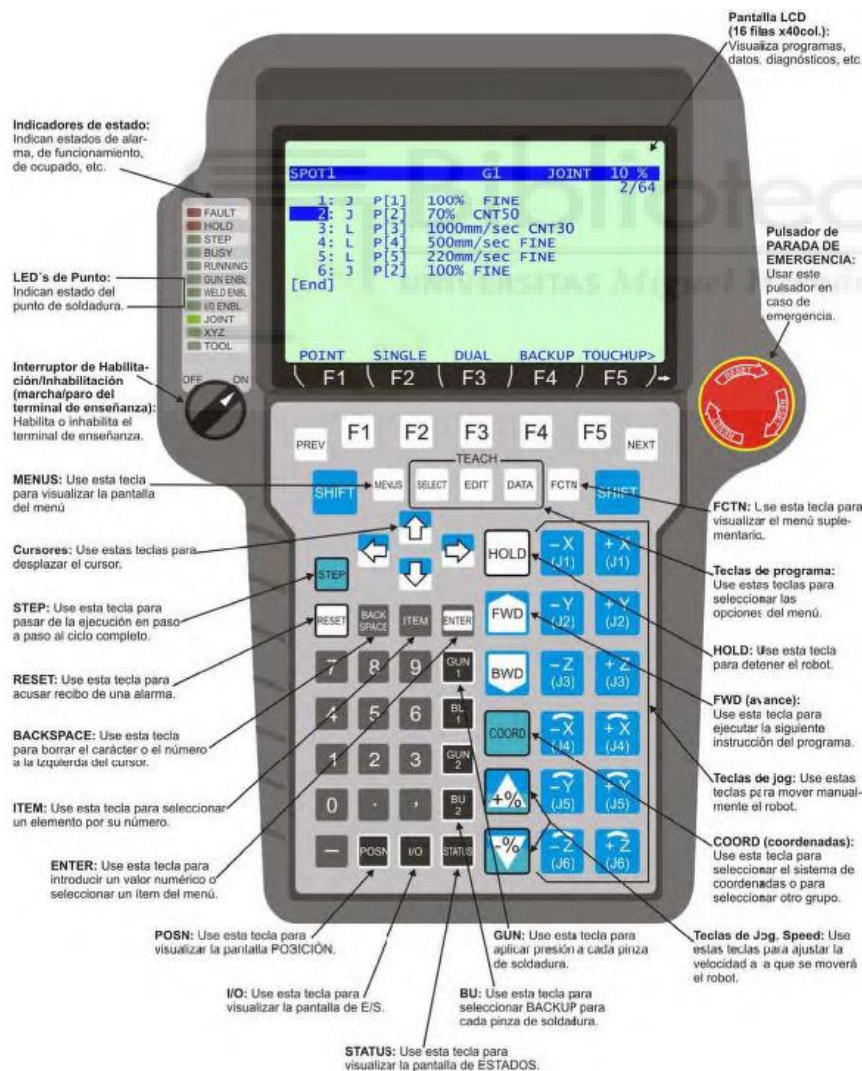


Figura 27 Estructura "teach pendant" Fanuc

INTRODUCCION ENTORNO PROGRAMACIÓN

Número de línea ejecutándose
Si indica 0 significa que el programa está abortado.

Programa ejecutándose

Programa o menú que se está editando

Cursor

Número de línea

Indicación de fin de programa

Modo operativo de la instalación
T1
T2
AUTO

Forward Backward Disable
El paso a paso está deshabilitado tanto hacia adelante como hacia atrás.

Estado del programa ejecutándose
ABORTED
PAUSED
RUNNIG

Velocidad de movimiento del robot

Posición actual del cursor /Total de líneas disponibles

Sistema de coordenadas de movimiento en manual
JOINT
JGFRM
USER
TOOL
(R1/RTCP)

Más funciones disponibles con NEXT

En esta línea se invita al operador a teclear datos

Línea de mensajes de error

Línea de funciones

Menú de función
Indica la descripción de la función.
Si aparece entre corchetes [] indica que posee varias funciones.

1	FOLGE1	LINE 3	T1	PAUSED	FBD
2	FOLGE1		JOINT	10 %	
3	1: J	P[1]	100%	FINE	
4	2: J	@P[2]	100%	FINE	
5	3: L	P[3]	80mm/sec	FINE	
6	4: L	P[4]	80mm/sec	CNT/100	
7	5: L	P[5]	80mm/sec	FINE	
8	6: J	P[2]	100%	CNT100	
9	[End]				
10					
11	POINT				TOUCHUP>
12					
13					
14					
15					
16					

Figura 28 Estructura entorno trabajo Fanuc

INTRODUCCIÓN ROBOT

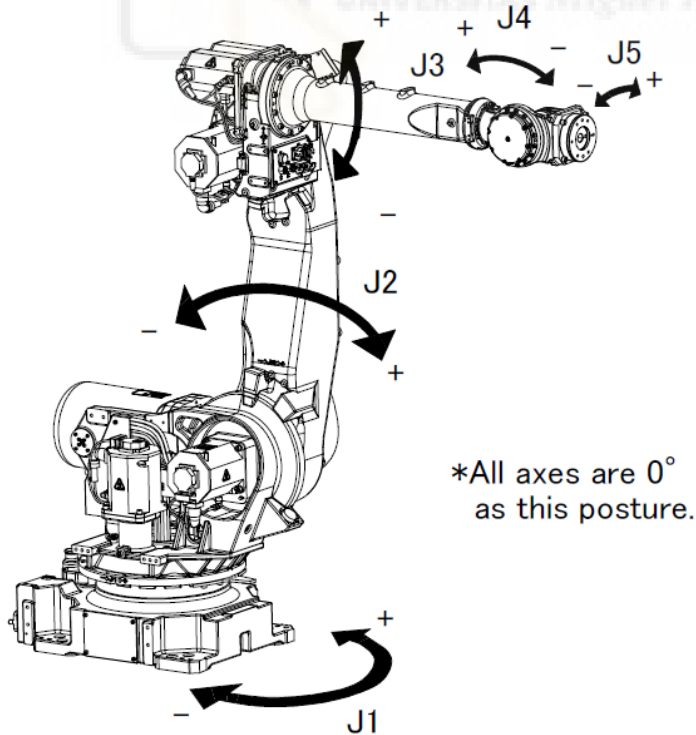
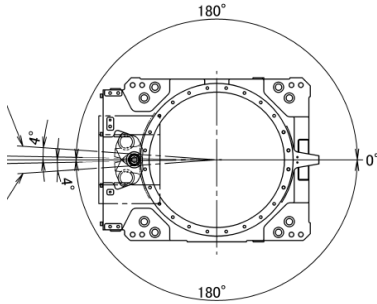


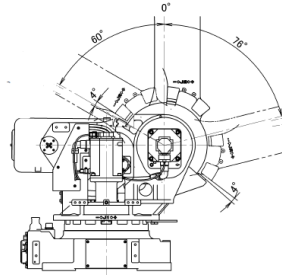
Figura 29 Distribución ejes robot Fanuc

LIMITE MOVIMIENTO EJES

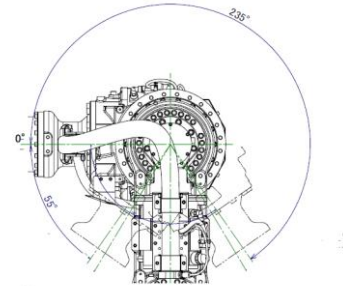
EJE 1



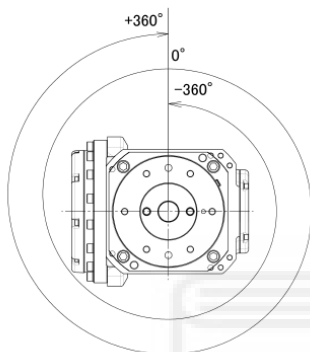
EJE 2



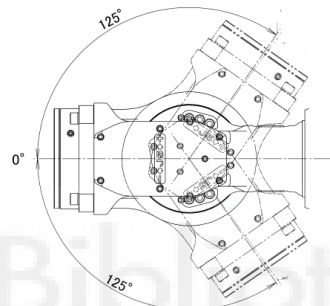
EJE 3



EJE 4



EJE 5



EJE 6

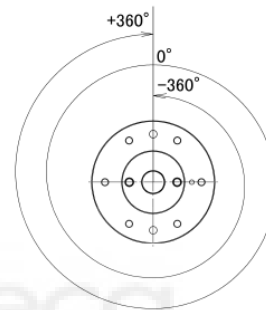


Figura 30 Limite ejes robot Fanuc

SELECCIÓN ROBOTS EMPLEADOS

ROBOT R2000iC-165F

La flexibilidad con cargas pesadas que aporta este modelo polivalente de seis ejes de la gama R-2000 define nuevos estándares en prácticamente todas las industrias. Equipado con una muñeca estilizada y con capacidad para lograr tiempos de ciclo excelentes, se trata de un robot ágil de soldadura por puntos para automoción y otras aplicaciones semejantes. Además, se ha diseñado para maximizar el uso del espacio y funcionar en estrecha proximidad con otros robots.

EJES 6

ALCANCE 2655 mm

CAPACIDAD DE CARGA 165 kg



Figura 31 Robot Fanuc R2000iC

INICIO SIMULACIÓN ROBOGUIDE

2- CREAR ROBOT 1

El Primer paso será iniciar el programa i buscar el icono de nuevo documento como aparece a continuación:

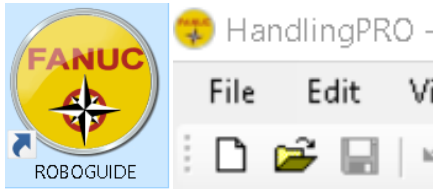


Figura 32 Icono programa Roboguide

Step1 - Al seleccionar documento nuevo aparece la pantalla donde escribimos el nombre del documento a crear y seleccionamos siguiente para seguir configurando toda nuestra celda robotizada.

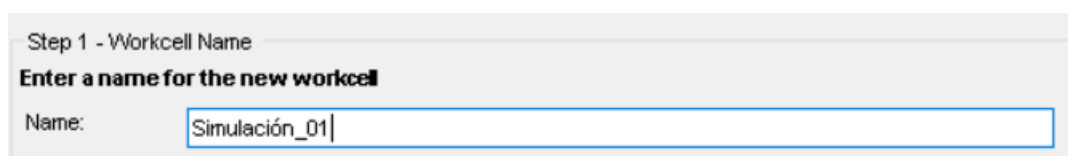
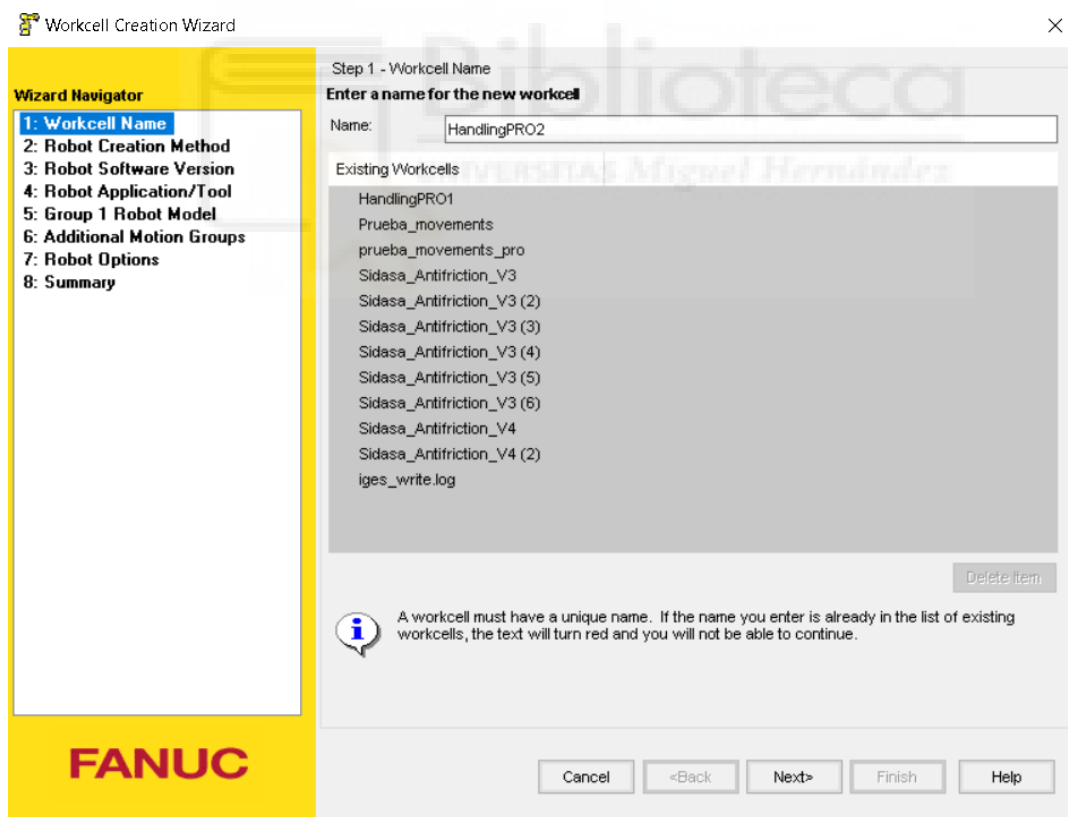


Figura 33 y 34 Selección nombre celda Roboguide

Step-2 Seleccionar el método de creación de nuestro robot, eligiendo entre utilizar la configuración por defecto, la configuración utilizada anteriormente o crear un robot desde un backup realizado anteriormente.

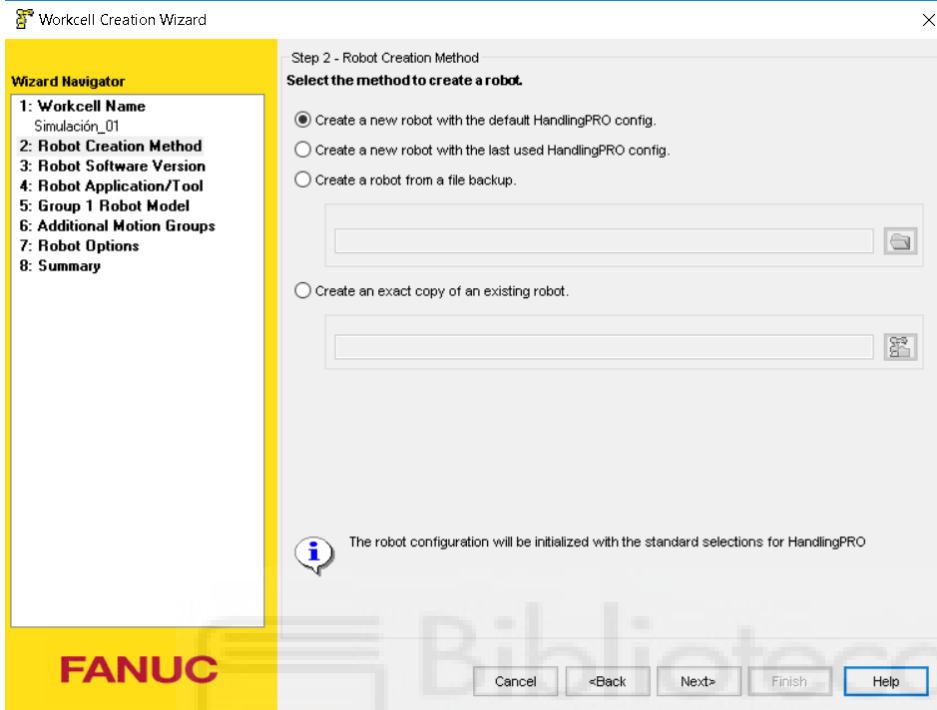


Figura 35 Selección método creación Roboguide

Step-3 Seleccionar al software del robot. Normalmente seleccionamos la última versión disponible. Utilizar versiones anteriores modificaría las características que podemos utilizar con el robot así como la versión del teach pendant (el mando)

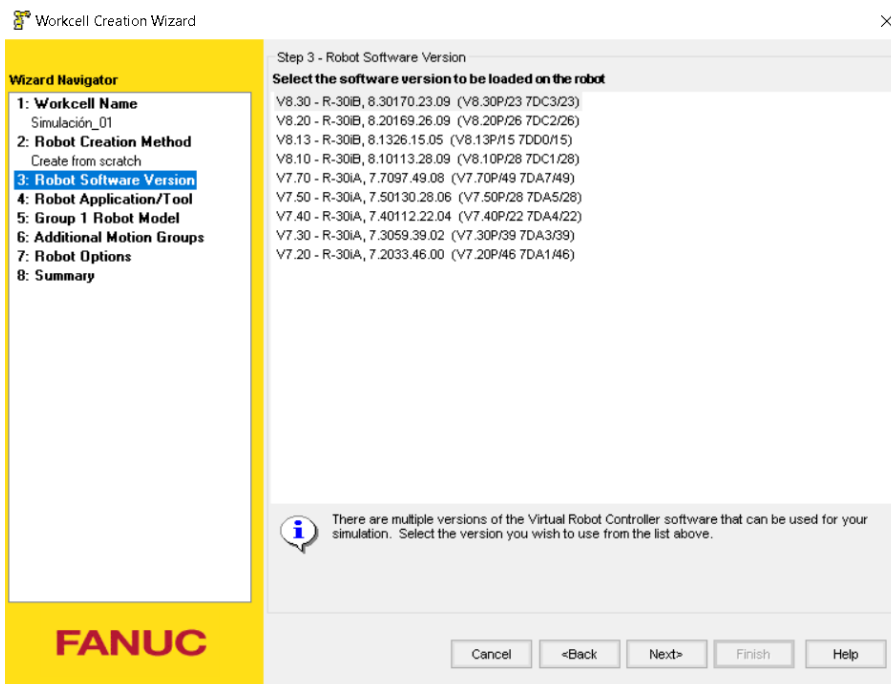


Figura 36 Selección software robot Roboguide

Step-4 Seleccionar la tecnología del robot. El tipo de tecnología a utilizar se relaciona directamente con la aplicación a la que esté destinado el robot. En este caso seleccionamos HandlingTool porque básicamente nuestro robot estará destinado a operaciones de pick and place

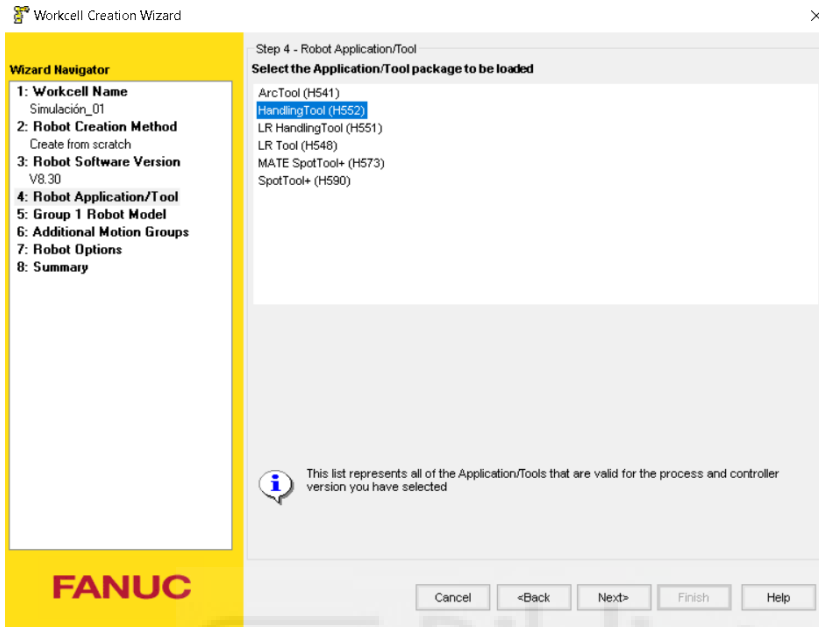


Figura 37 Selección tecnología robot Roboguide

Step 5- Seleccionar el modelo exacto de robot a utilizar. Esta selección no es fácil ya que disponemos de una gran serie de robots a nuestra disposición. Los dos parámetros más significativos a la hora de la selección de un robot pueden ser el tamaño del mismo, así como la capacidad de carga, aunque también se tendrá en cuenta la precisión y rapidez a la hora de efectuar tareas. En este caso seleccionamos el R-2000iC/165F ya que su tamaño, capacidad de carga y eficacia lo hace ideal para este tipo de tareas.

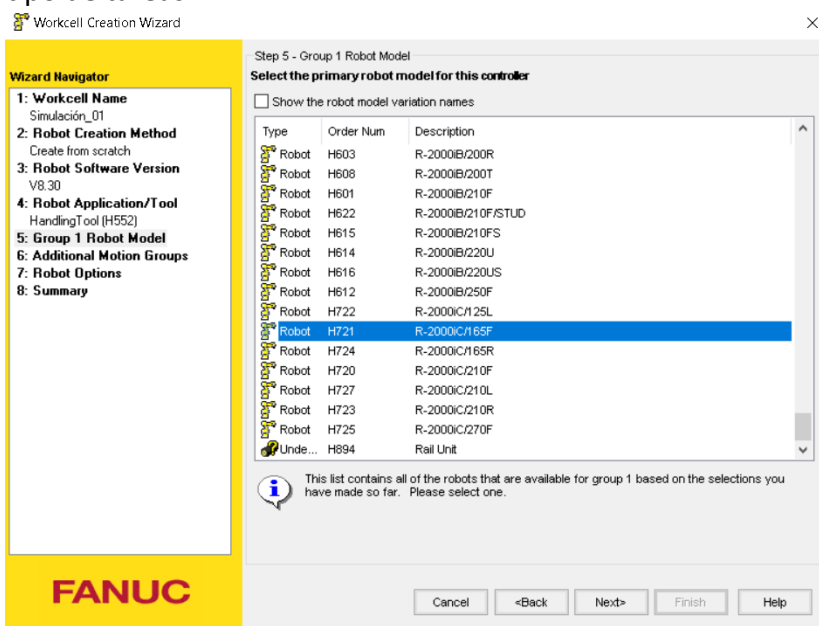


Figura 38 Selección modelo robot Roboguide

Step-6 Ahora podemos seleccionar ejes adicionales que nos pueden ayudar a la simulación como veremos más adelante. Normalmente no se utilizan pero se pueden crear fácilmente seleccionando el tipo motor a utilizar y posicionándolo. En este caso no creamos ningún eje externo al robot ya que no lo vamos a utilizar.

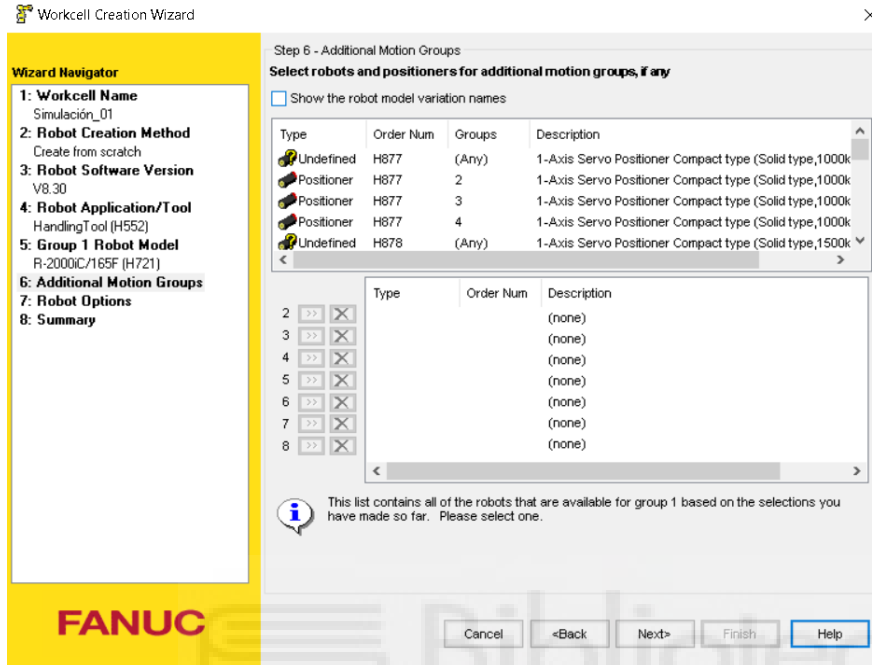


Figura 39 Selección ejes adicionales robot Roboguide

Step-7 Cargar opciones extra al software inicial. Roboguide dispone de muchos extras que nos permiten controlar y/o calcular parámetros adicionales a la programación común. En este caso no necesitamos de ninguna opción extra por lo que cargamos sólo las opciones definidas por defecto.

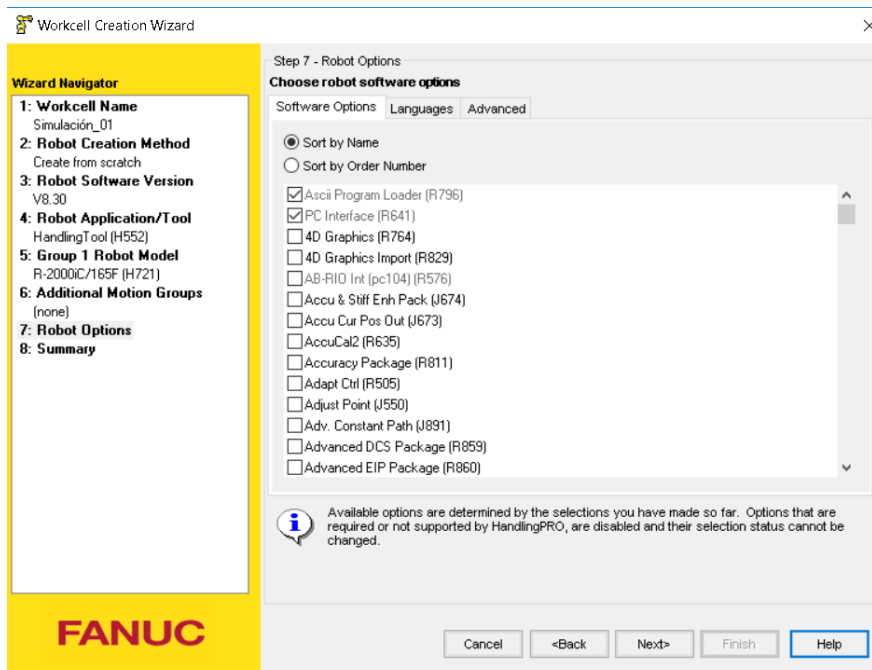


Figura 40 Selección opciones extras Roboguide

Step-8 Este paso es sólo un resumen de todas las opciones seleccionadas por lo que sólo hay que seleccionar la opción Finalizar para continuar con la creación de nuestra celda de simulación.

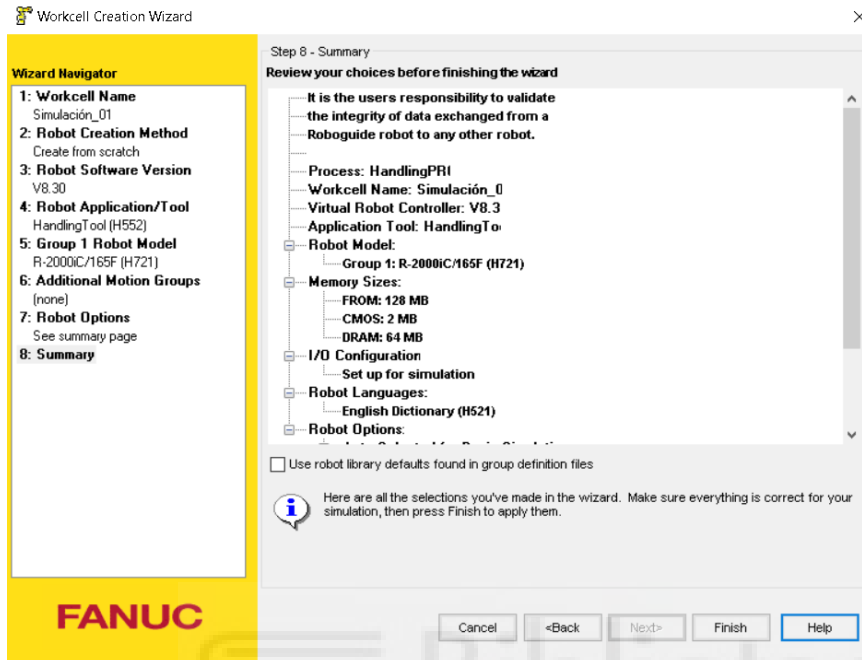


Figura 41 Resumen estación a cargar Roboguide

Después de cargar todas las configuraciones seleccionadas sólo es necesario seleccionar el tipo de Flange del robot que es la parte final del robot (el disco de anclaje de la herramienta a utilizar). Seleccionamos el Standard presionando el número 1 y continua con la carga del robot.

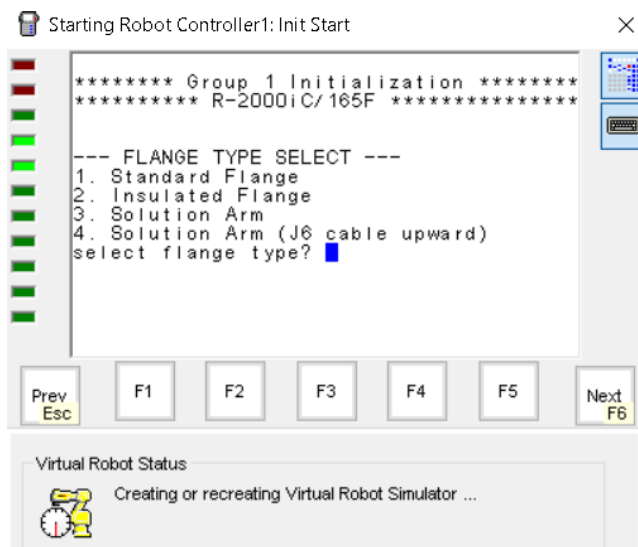


Figura 42 Selección tipo flange a cargar Roboguide

Finalmente obtenemos el robot seleccionado representado en un entorno 3d que será el espacio donde vamos a continuar creando nuestra celda de simulación.

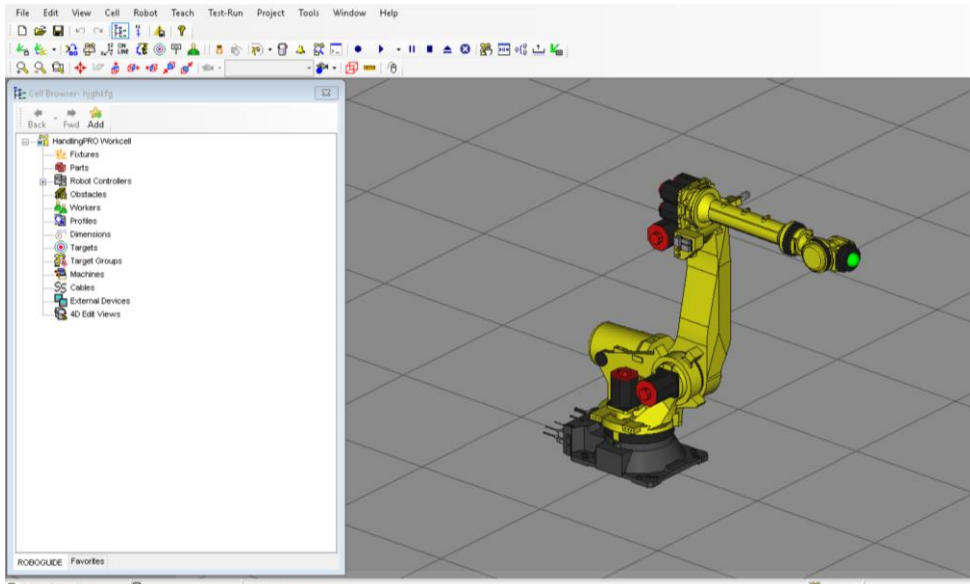


Figura 43 Robot 01 cargado Roboguide

3- CREAR ROBOT 2

Existen varias formas de realizar este proceso, una de ellas es crear una copia exacta del mismo robot seleccionando el controlador del robot creado anteriormente y con click derecho Add Robot y Add robot clone dentro del controlador del robot 1 en el menú llamado “cell browser” .

Automáticamente se creará un robot exactamente igual que el creado anteriormente con ubicación en la cota definida en el dialogo emergente.

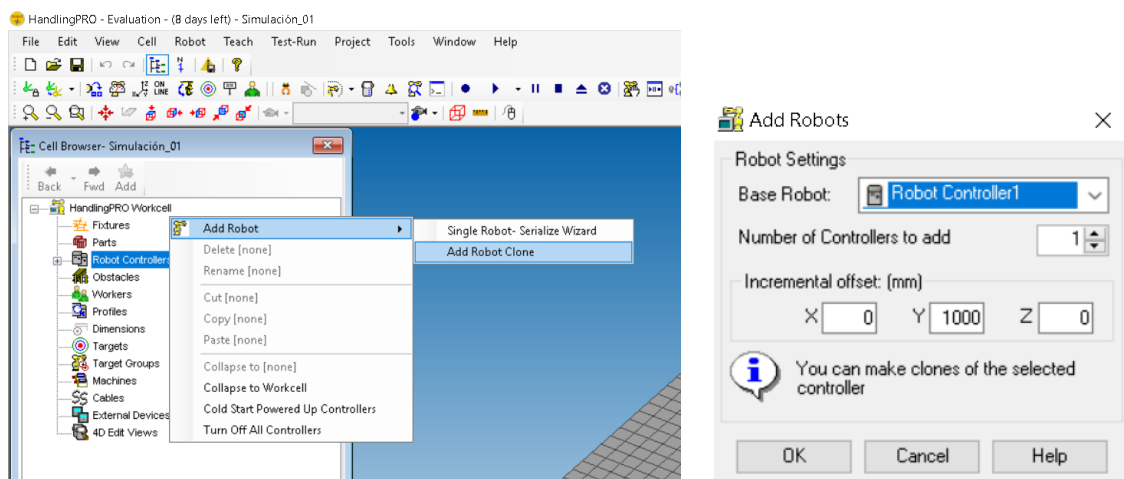


Figura 44 Creación robot 02 Roboguide

Una vez creado el segundo robot hay que situarlo en su zona de trabajo con la orientación deseada. Haciendo doble click sobre el robot que deseamos emplazar podemos editar la orientación y posición final de éste en cualquier momento.

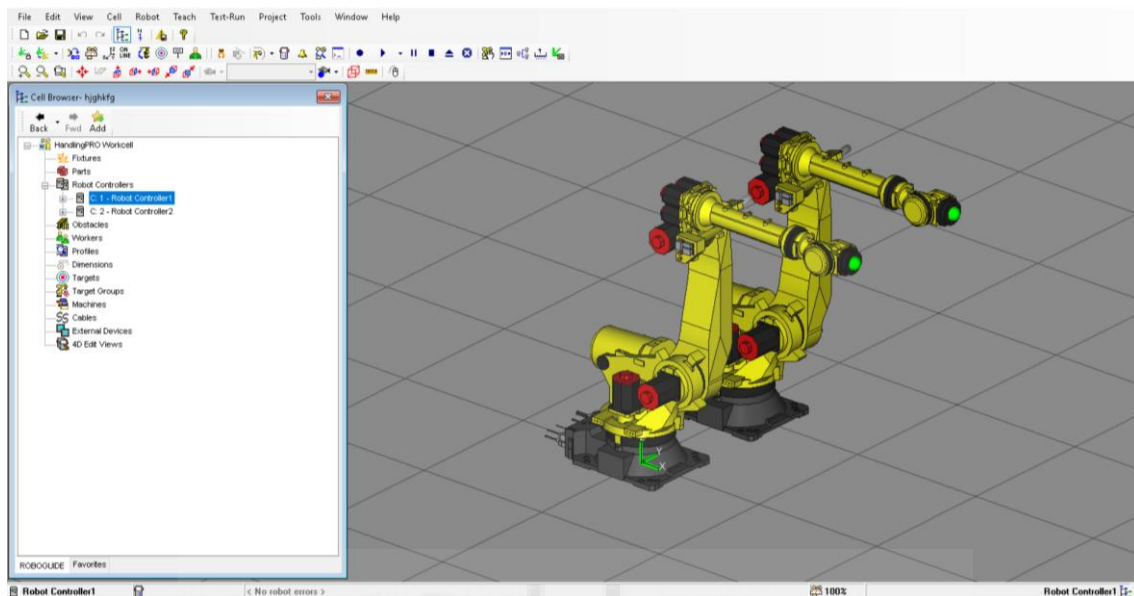


Figura 45 Robot 01 y robot 02 cargados Roboguide

Editamos los valores de X e Y así como la orientación en el eje Z (eje R) asignando 180 grados para colocar un robot enfrente del otro.

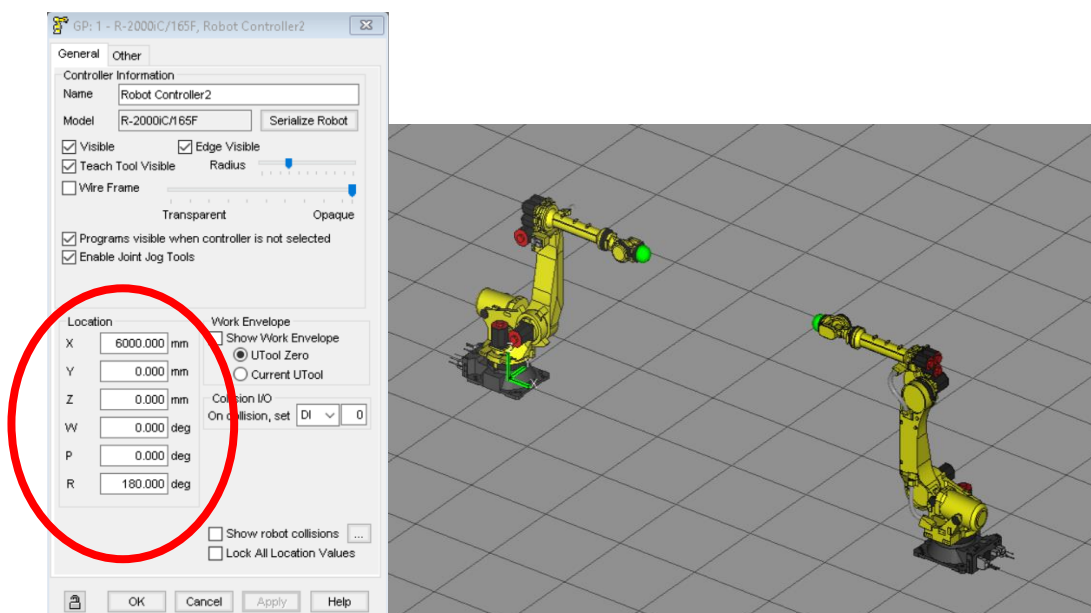


Figura 46 Posicionamiento ambos robots Roboguide

4 - CARGAR ENTORNO SIMULACIÓN

Una vez tenemos los dos robots cargados es el momento de crear todo el entorno de simulación donde vamos a poder desplazar el robot y crear programas de simulación. Para cargar elementos del entorno utilizamos nuestro menú “cell browser” seleccionando el apartado de Fixtures i con el click derecho, add fixture, single cad file.

En este caso vamos a empezar cargando todas las barreras que van a formar nuestra celda cerrada de producción. Los modelos a cargar deben tener extensión .stl o .iges

Una vez cargada la pieza aparece en una cota definida por defecto y con color por defecto (gris). Haciendo doble click sobre el elemento cargado dentro del listado del “cell browser” podemos editar tanto colores como orientaciones y/o posiciones finales del elemento tal y como se muestra en las capturas siguientes:

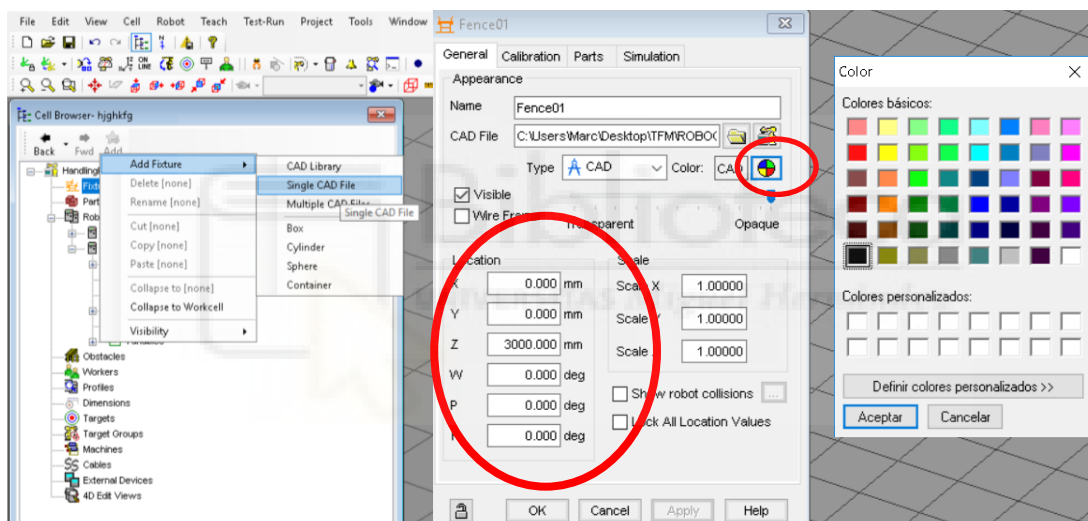


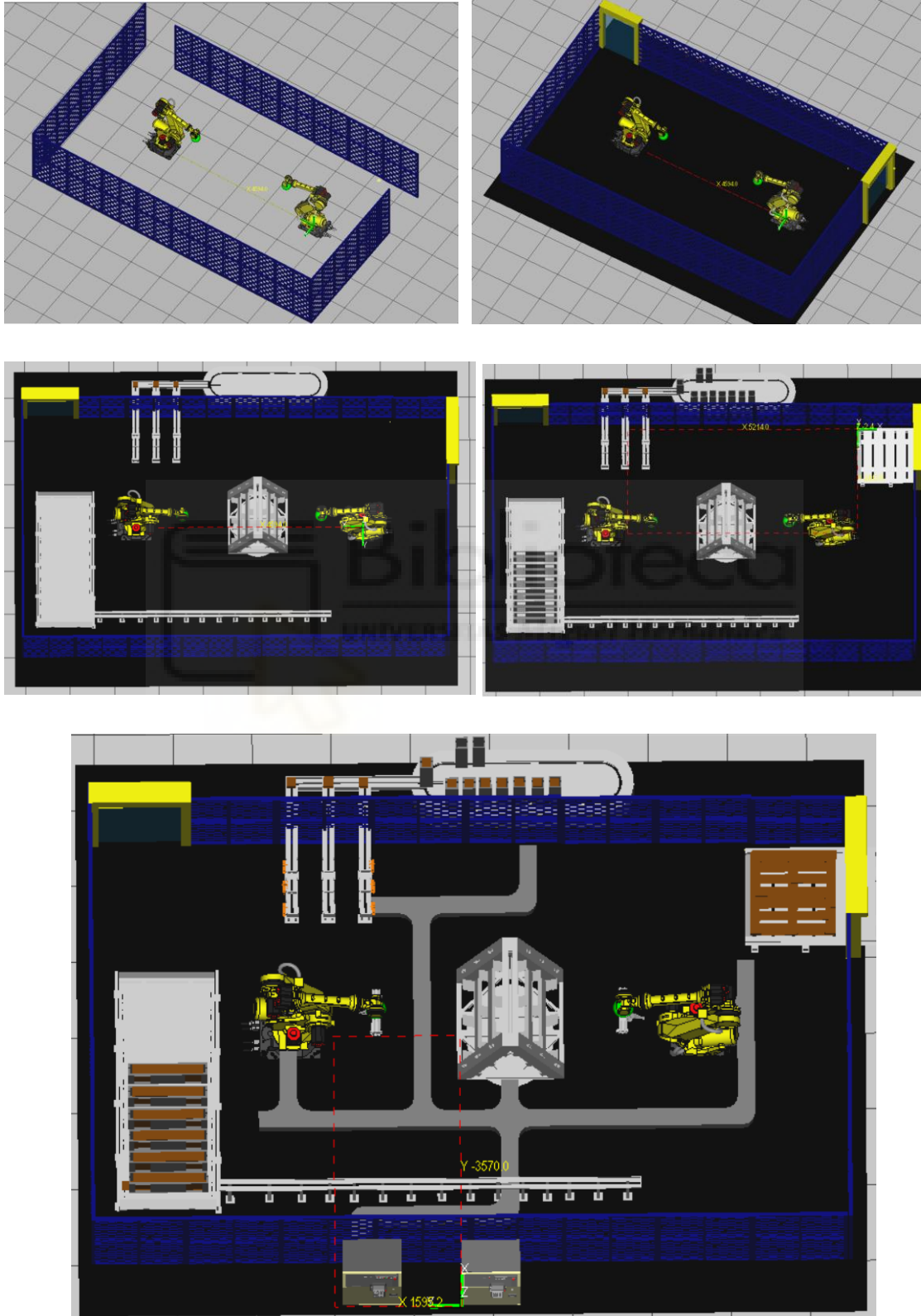
Figura 47 Cargar elementos entorno Roboguide

Si todas nuestras barreras o vallas forman parte de un mismo archivo solamente con este proceso ya tendríamos cargada nuestra celda cerrada.

Si no tenemos un archivo que contenga todas las vallas juntas podemos repetir este proceso una a una o bien hacer copiar y pegar de la primera dentro del menú de las Fixtures en el “cell browser” y editar los valores que sean necesarios.

Otra manera muy útil para la colocación de los elementos es seleccionarlos haciendo click sobre el elemento físico en pantalla y moviendo el eje de coordenadas que aparece de color verde debajo de cada uno de los elementos cargados. Solamente arrastrando el eje que se desea mover la pieza cambia su cota definida.

De esta manera podemos cargar todo el entorno de trabajo del robot obteniendo los siguientes resultados:



Figuras 48 a 52 Entorno simulación cargado Roboguide

5- INSERTAR HERRAMIENTAS TRABAJO

Una vez cargado todo el entorno de trabajo de la simulación es momento ahora de cargar las herramientas de trabajo de los robots, para ello debemos seleccionar el robot dentro del “cell browser” y abrir el apartado de tooling y seleccionar el número de herramienta deseado.

Haciendo click derecho sobre el número de tool deseado (en este caso el 2) seleccionamos la opción Add link y Cad file para buscar el archivo .STL de nuestra herramienta.

Si al cargar nuestra herramienta la posición o la orientación no es la deseada editamos los valores haciendo doble click sobre el link añadido en el “cell browser”

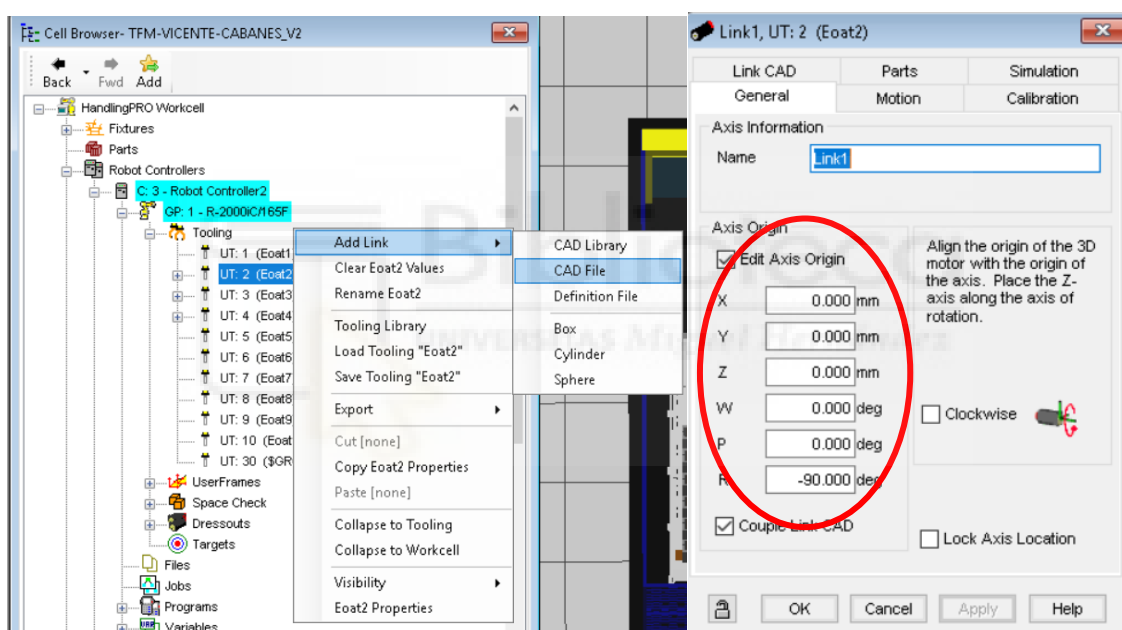


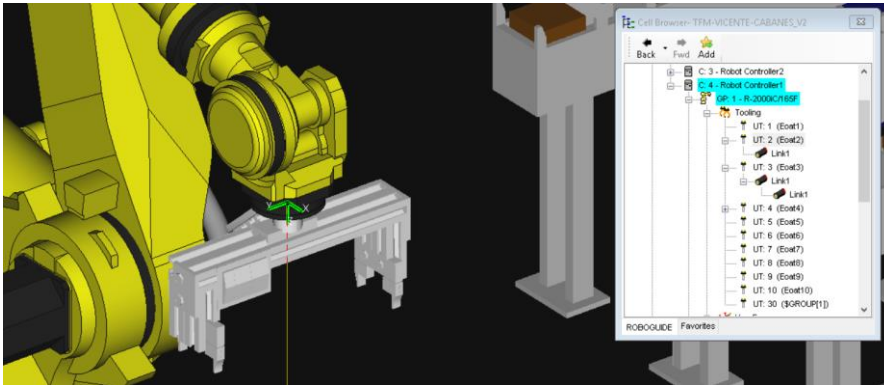
Figura 53 Insertar herramientas Roboguide

Una vez tenemos el tool montado sobre el robot en la posición y orientación deseada hay que repetir el proceso para el otro robot obteniendo los dos robots con su herramienta colocada y cargada correctamente.

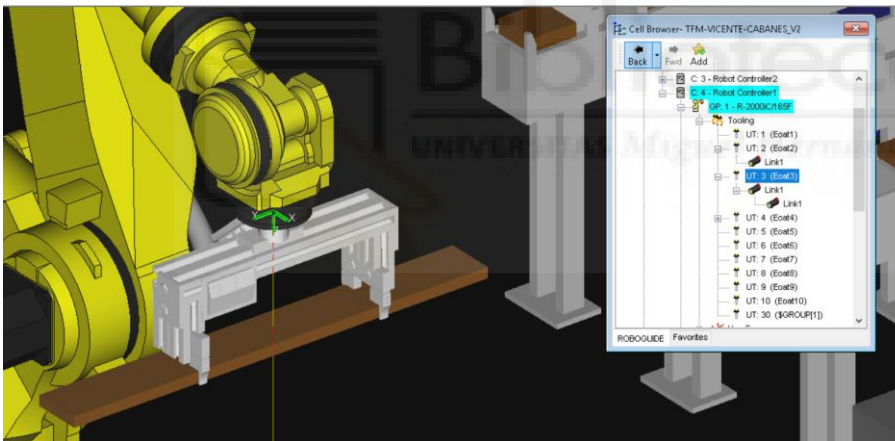
Una vez este proceso queda terminado pasamos a crear herramientas auxiliares con las piezas en posición de pick. Para realizar este proceso primero duplicamos las herramientas utilizando los números 3 y 4 del tooling de cada robot haciendo copiar y pegar sobre el link que representa nuestra herramienta.

Ahora que tenemos el tool 2, 3 y 4 cargados con el mismo link pasamos a añadir las piezas a la herramienta numero 3 (un listón). Para realizar este proceso basta con añadir un link sobre el link ya existente y editar el color y la posición de éste. Al cambiar de tool vemos como aparece la pieza en la posición exacta de pick.

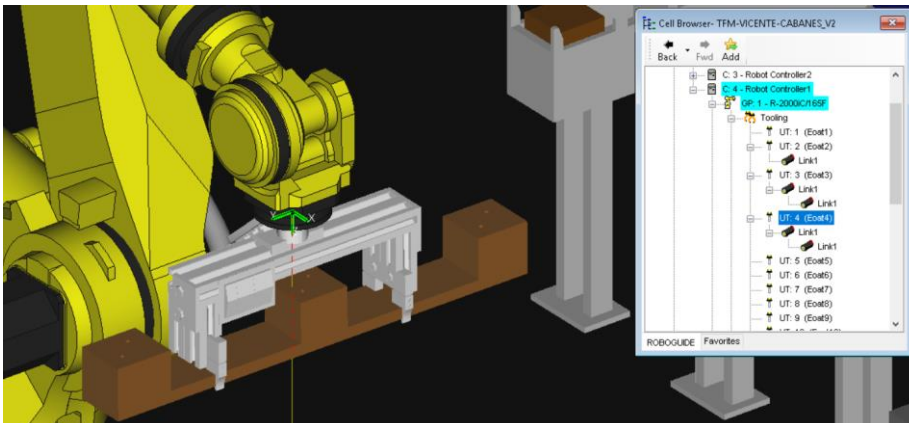
ROBOT 1 TOOL 2



ROBOT 1 TOOL 3



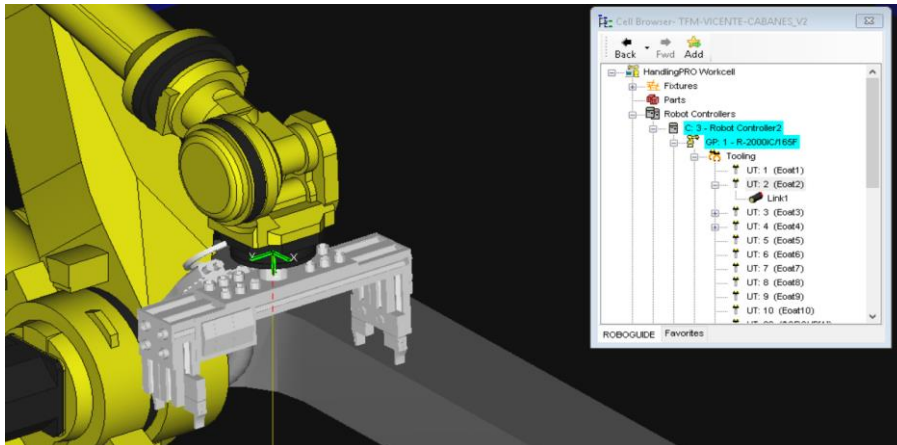
ROBOT 1 TOOL 4



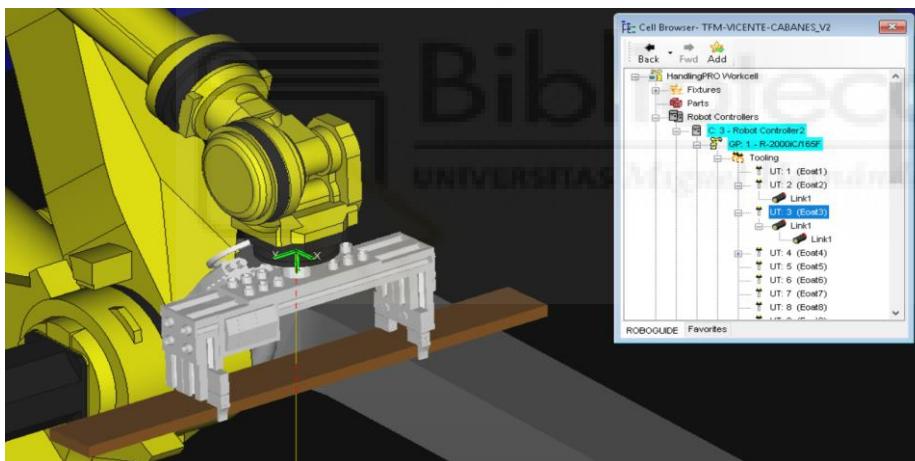
Figuras 54 a 56 Diferentes herramientas robot 01 Roboguide

Haremos exactamente lo mismo para el robot 2 aunque la herramienta sea distinta, pero en este caso el tool 4 será nuestro pale terminado y lo utilizaremos para realizar la simulación final de salida del pale terminado.

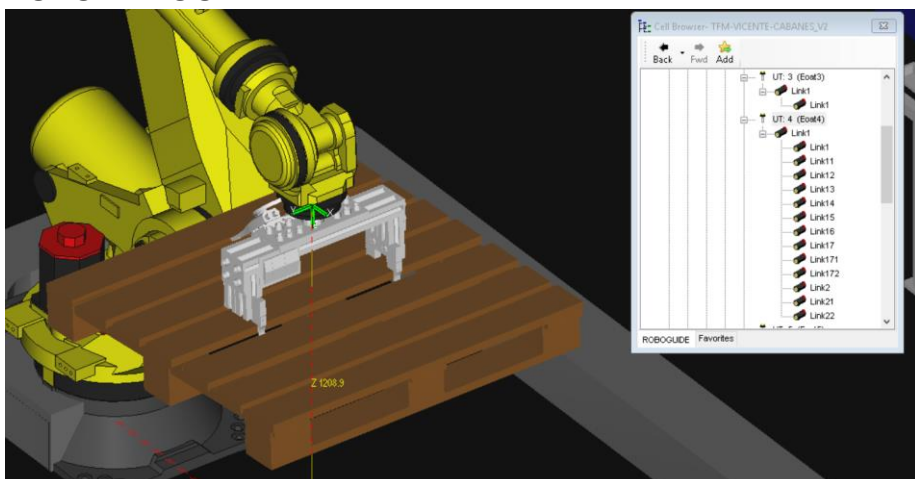
ROBOT 2 TOOL 2



ROBOT 2 TOOL 3



ROBOT 2 TOOL 4



Figuras 57 a 59 Diferentes herramientas robot 02 Roboguide

6- SIMULACIÓN MATERIAL ENTRADA

El primer paso para iniciar la simulación es configurar los movimientos de las piezas de entrada a la celda de producción.

Existen dos formas diferentes para poder reproducir los movimientos; se pueden crear ejes externos de tal forma que podemos controlar los movimientos de las piezas como un eje más del robot o bien asignar cada movimiento a una señal digital controlada también por el propio robot.

Seleccionamos la segunda opción ya que visualmente es más fácil entender el funcionamiento de la simulación asignando señales digitales con sus respectivos nombres pudiendo activar y desactivar las señales cuando sea necesario.

Para ello vamos a eliminar la cinta de listones introducida anteriormente en la parte 3 como un fixture para añadir la misma como una máquina. Dentro del menú “cell browser” buscamos “machines” y con el click derecho seleccionamos add file, cad file. Una vez introducida la colocamos editando los valores de la **pestaña LinkCad** en la misma posición donde se encontraba la anterior y ya tenemos lista la primera cinta de transporte.

Hacemos lo mismo para la otra cinta de tacos eliminarla de fixtures para añadirla como una máquina colocándola en la misma posición. Cuando este proceso esté finalizado el “cell browser” quedará de la siguiente manera:

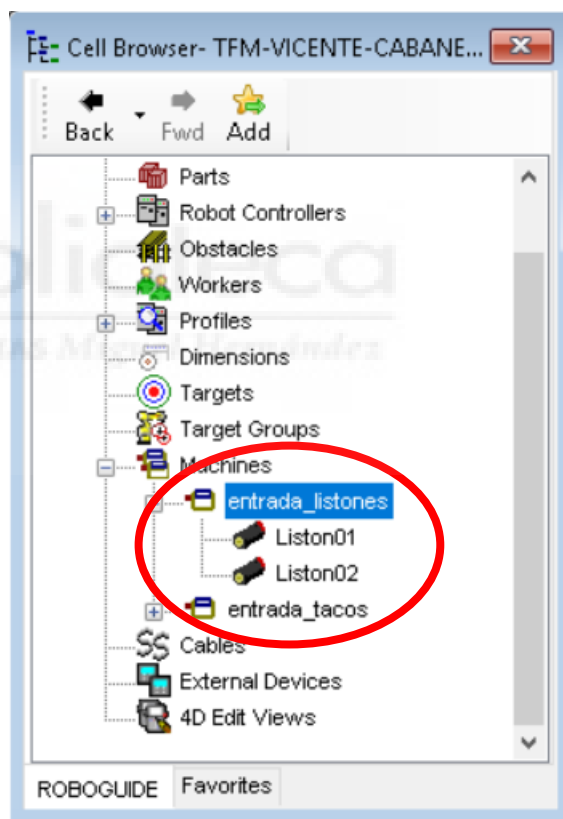


Figura 60 Menú “cell browser” Roboguide

Una vez las dos máquinas configuradas debemos introducir las piezas que van a moverse sobre ellas haciendo click derecho sobre cada una de las máquinas y como siempre add file, cad file y colocaremos las piezas en la posición de partida deseada, es decir dónde va a dar inicio el movimiento editando como siempre los valores del link de posición y orientación como se ha hecho anteriormente al añadir los fixtures.

Una vez las maquinas configuradas es momento ahora de editar los movimientos y probar si funcionan correctamente.

Vamos ahora a introducir donde va a ser la posición final de nuestras piezas en la cinta y que señal digital va a activar dicho movimiento haciendo doble click sobre el link introducido dentro de la maquina creada y seleccionando la pestaña de motion.

Motion control type: device I/O controlled

Axis type: linear

Speed: Time

Output device	IO Tag	Value	Location
Controller 1	DO(1)	OFF	0
Controller 1	DO(1)	ON	1985

Aplicamos todos los cambios y seleccionamos OK para cerrar el cuadro de diálogo.

Tener en cuenta que los movimientos se originan siempre en el eje z de la pieza, si nuestro eje z tiene orientación no deseada podemos cambiarla dentro de la pestaña de General sin cambiar la orientación de la pieza física, solo los ejes.



Figura 61 Parámetros movimientos Roboguide

Hacemos lo mismo para una segunda pieza esta vez controlada por el Controller 2 (DO1) y con los valores de location de la posición de pick del robot 2 así como los tacos de entrada controlados por el robot1 en la cinta de tacos y para las 3 posiciones de ensamblaje de la base del pale (DO2, DO4 y DO6).

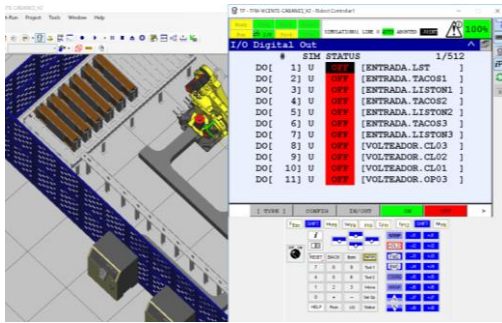
Ahora podemos probar que al activar o desactivar la señal digital configurada en el controlador del robot la pieza cambia su posición a la posición deseada tanto en un robot como en el otro y para las dos cintas (listones y tacos).

Si deseamos más piezas podemos añadirlas y configurarlas exactamente igual que las anteriores estableciendo la configuración deseada y que nos permita la simulación.

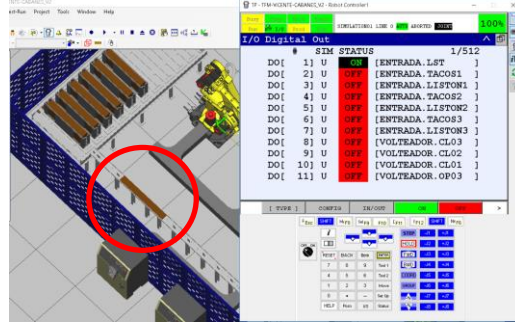
Para probar los efectos debemos habilitar un teach pendant seleccionando el robot y el icono del teach en la barra de herramientas superior. Una vez abierto el controlador debemos seleccionar la tecla Menú, I/O, Digital y aparecen nuestras señales digitales.

ENTRADA LISTONES ROBOT 1

Señal DO (1) OFF

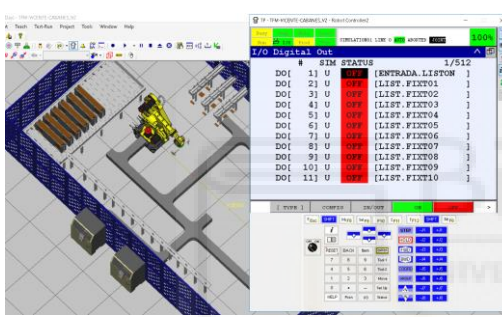


Señal DO (1) ON

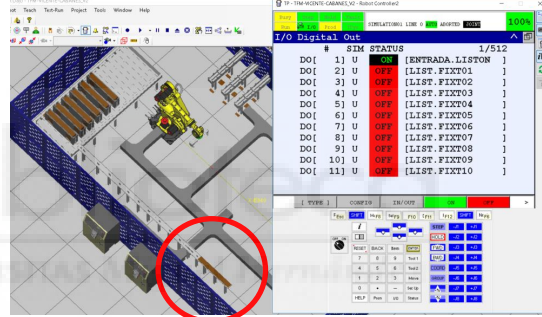


ENTRADA LISTONES ROBOT 2

Señal DO (1) OFF

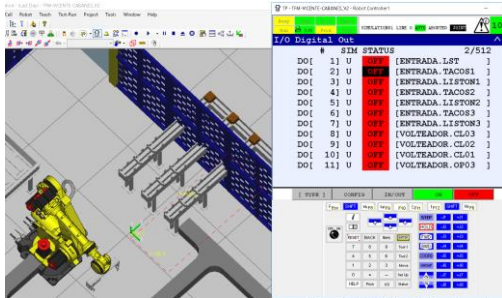


Señal DO (1) ON

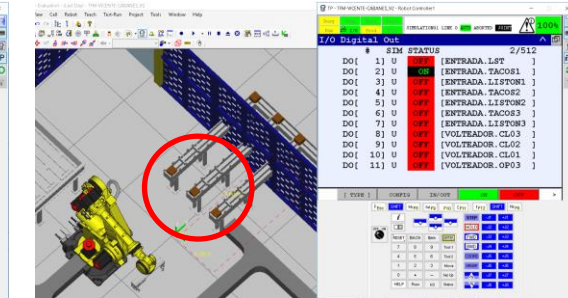


ENTRADA TACOS ROBOT 1

Señal DO (2) OFF

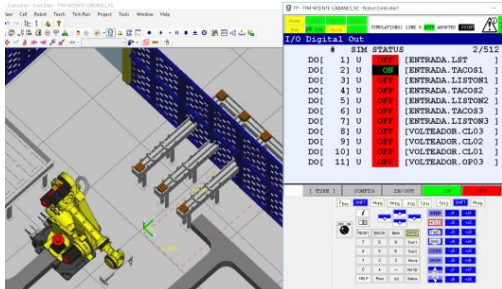


Señal DO (2) ON

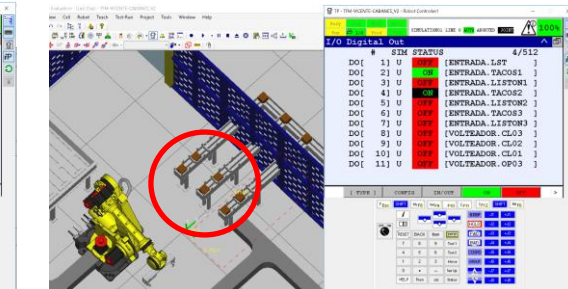


ENTRADA TACOS ROBOT 1

Señal DO (4) OFF



Señal DO (4) ON



Figuras 62 a 69 Movimientos entrada material

7- ESTRUCTURA SIMULACIÓN

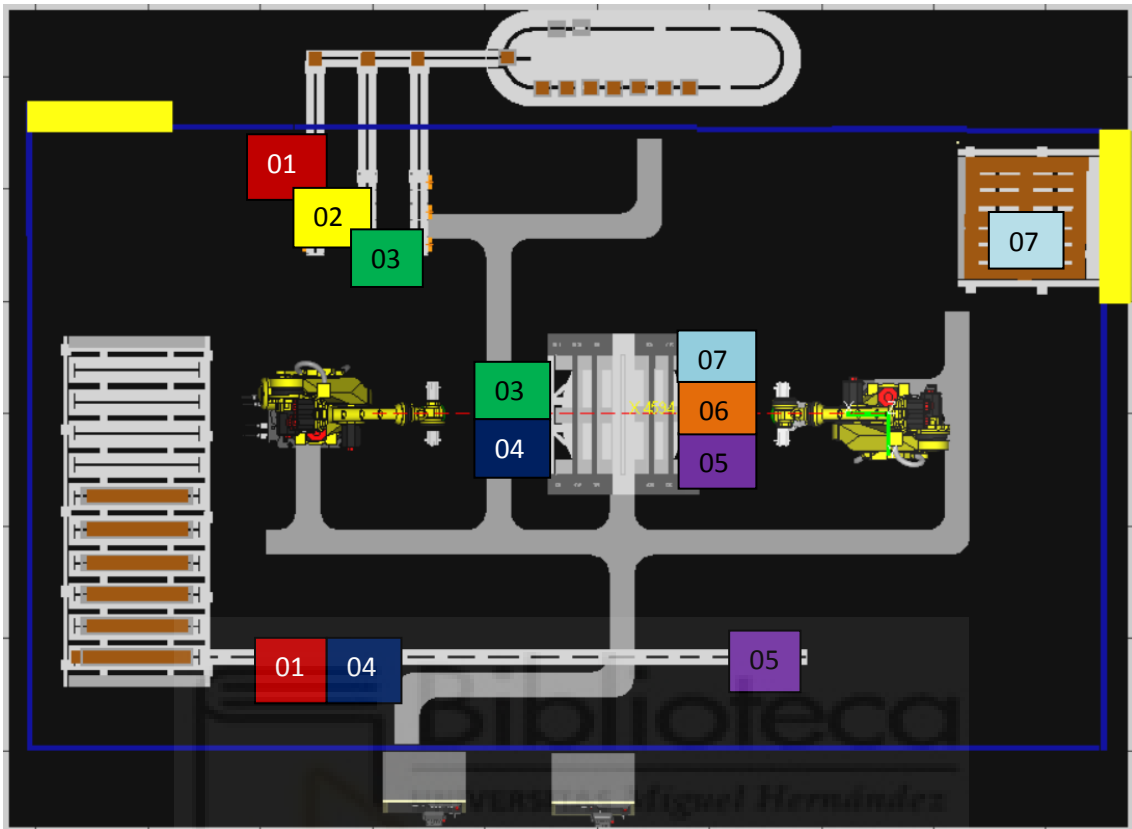


Figura 70 Estructura procesos simulación Roboguide

- PROCESO r1 - 01
Pick and place base pallet
- PROCESO r1 - 02
Unión base pallet clavos (Nail gun)
- PROCESO r1 - 03
Pick and place base pallet - Fixture
- PROCESO r1 - 04
Pick and place capa intermedia - Fixture
- PROCESO r2 - 05
Pick and place capa final - Fixture
- PROCESO r2 - 06
Unión capa intermedia y final grapas
- PROCESO r2 - 07
Pick and place pallet terminado producto final

8- DESARROLLO SIMULACIÓN

Una vez terminado todo el proceso de configuración de herramientas y movimientos de pieza es momento de empezar a programar movimientos del robot.

1- Abrimos el “teach pendant” del primer robot seleccionando el icono de la barra de herramientas superior y creamos un programa nuevo llamado simulation01 de la siguiente forma: Tecla select (F9) y seleccionar la opción Create.

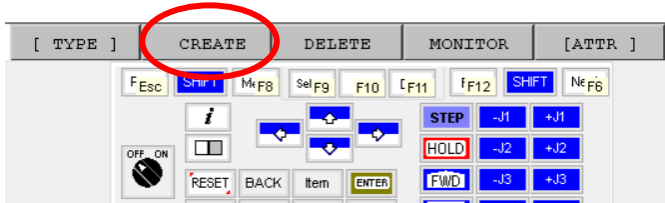


Figura 71 Crear programa nuevo “teach pendant”

2- Escribir un nombre de programa usando Upper Case (mayúsculas) + END. Ahora aparece un programa nuevo sin ninguna línea (debemos añadirlas)

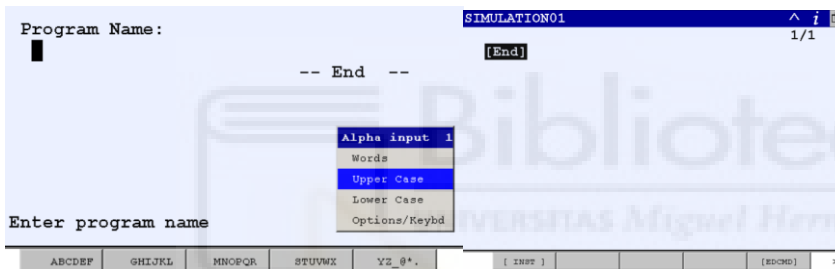


Figura 72 y 73 Nombre programa nuevo “teach pendant”

3- Habilitar el teach cambiando el selector a ON y seleccionar ECMD y la opción Insert. Posteriormente preguntará cuantas líneas deseamos insertar y escribimos 100 + ENTER.

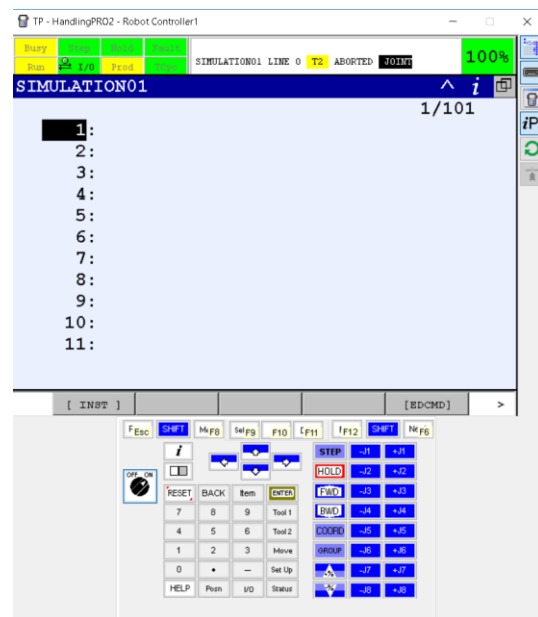
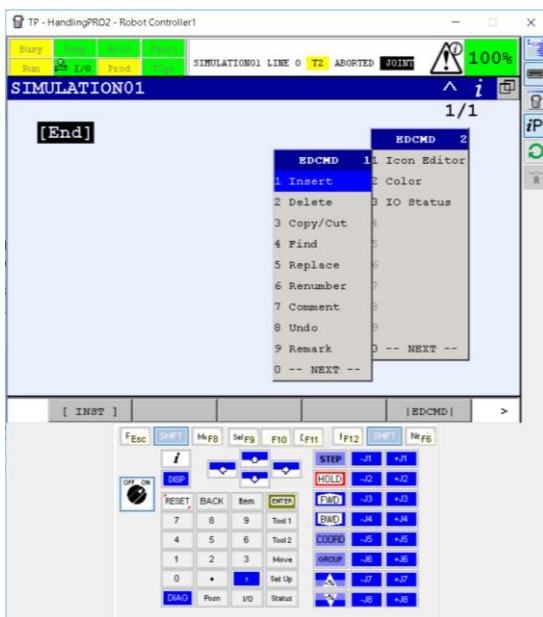


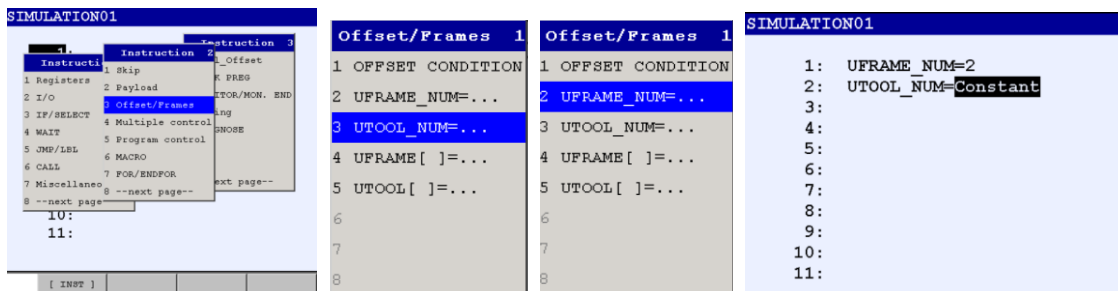
Figura 74 y 75 Añadir líneas al programa

4- Insertar las primeras líneas (siempre será la selección del frame y tool a utilizar) y el primer punto (home o wait pos).

Tecla INST, menú instruction 2, Offset/Frames, UFrame_Num, Constant.

Repetimos el proceso seleccionando ahora UTool_Num

Una vez tenemos la instrucción cambiamos Constant por el número deseado.



Figuras 76 a 79 Selección herramienta y frame a utilizar

5- Crear primera posición moviendo el robot a la posición deseada.

Existen 4 tipos de movimiento en Roboguide:

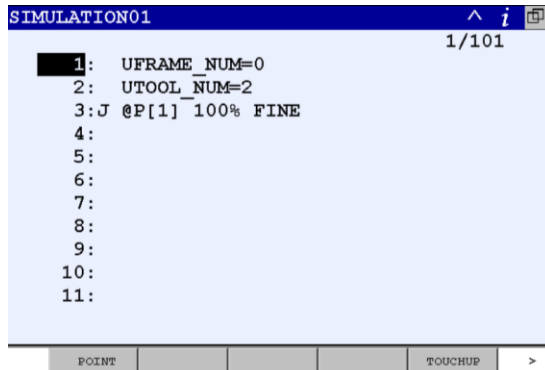
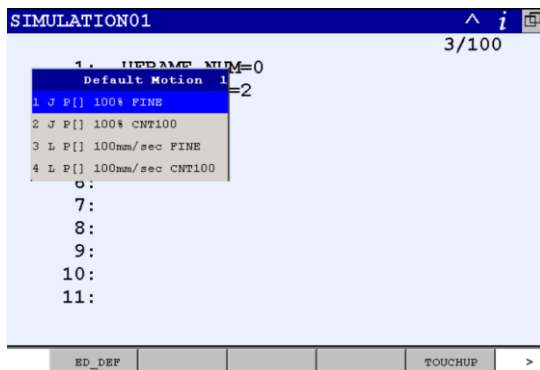
- Joint Fine
 - Joint Continuous
 - Lineal Fine
 - Lineal Continuous
- } Movimientos articulares
- } Movimientos Lineales

Para los movimientos donde el robot va a efectuar una parada como son los movimientos de pick, place o de espera o inicio siempre utilizaremos Fine.

Para los movimientos donde el robot no va a efectuar parada, es decir, movimientos de trayectorias del robot siempre utilizaremos Continuous.

Movemos el robot a la posición deseada utilizando la tecla Shift seleccionada y con el teach pendant habilitado (ON) y utilizamos la tecla POINT para seleccionar Joint Fine y aparecerá nuestro punto creado en el código como el primer punto creado.

Siempre podemos saber las coordenadas del punto seleccionando el número del punto y la tecla Position.



Figuras 80 y 81 Crear primer punto simulación

PROCESO r1 – 01 (PICK AND PLACE BASE PALET)

Una vez establecida la posición de inicio del ciclo en el robot 01 se procede a programar los movimientos para coger la primera pieza (el listón).

Primeramente activamos la señal digital nº1 para hacer que aparezca la pieza que vamos a coger utilizando una instrucción en código.

Presionamos la tecla INST, seleccionamos I/O y DO [] = ... y escribiremos entre los corchetes el número de señal a setear y en los puntos suspensivos On ó Off.

En este caso la instrucción quedaría como DO[1]=On

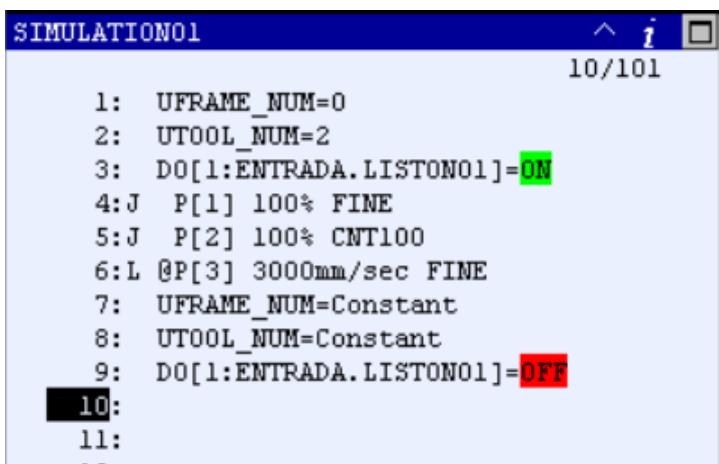
Movemos el robot a una posición de aproximación utilizando el teach pendant como anteriormente activándolo (ON) y presionando la tecla de Shift podemos mover el robot. Una vez en la posición deseada crearemos el siguiente punto en Joint y Continuidad ya que se trata de un movimiento de trayectoria.

Ahora movemos el robot hasta la posición final de pick y creamos otro punto esta vez en Lineal y movimiento Fine ya que el robot se detendrá en la posición de pick.

Podemos probar los movimientos utilizando la tecla forward desde el inicio del código creado bien instrucción a instrucción (tecla Step activada) o bien continuadamente de principio a final (tecla Step desactivada).

Una vez en la posición de pick utilizaremos el **recurso más importante de toda la simulación**. Se trata de hacer un cambio de herramienta a la creada anteriormente con el listón incorporado y al mismo tiempo desactivar la señal digital que controla la pieza suministrada por la cinta. Este paso nos permite visualizar una cogida de pieza de forma simulada pero realista.

Ahora el código quedaría de la siguiente manera:



```
SIMULATION01
10/101
1: UFRAME_NUM=0
2: UTOOL_NUM=2
3: DO[1:ENTRADA.LISTON01]=ON
4: J P[1] 100% FINE
5: J P[2] 100% CNT100
6: L @P[3] 3000mm/sec FINE
7: UFRAME_NUM=Constant
8: UTOOL_NUM=Constant
9: DO[1:ENTRADA.LISTON01]=OFF
10:
11:
12:
```

Figura 82 Código creado

Vamos ahora a mover el robot a una posición de aproximación y crear un punto en Joint y Continuo para realizar la salida desde la posición de pick.

Continuamos moviendo el robot a una posición de aproximación a la posición de place o dejada y creando otro punto igual que el anterior. Ahora activamos la señal digital que hemos configurado para que aparezcan los primeros tacos (DO2) para terminar moviendo el robot a la posición exacta de place y crear otro punto esta vez en Lineal y Fine ya que el robot se detiene para dejar la pieza en esa posición.

Una vez tenemos el robot en posición de drop realizamos el proceso similar al pick realizando un cambio de herramienta a la usada inicialmente y crearemos piezas en la cinta (listones) de la forma ya explicada en el paso 5 (simulación material entrada) para que aparezcan en el mismo momento que cambiamos de herramienta. Asignamos estas piezas a otras 3 nuevas señales digitales (DO3, DO5, DO7).

Así pues cada vez que realizamos el proceso de pick repetiremos los mismos puntos y señales y cada vez que realizamos el proceso de drop variaremos las señales a activar dependiendo la posición en la que vamos a dejar la pieza (3 posiciones distintas).

Una vez editado el código queda de la siguiente forma:

```

UFRAME_NUM=0 ;
UTOOL_NUM=2 ;
J P[5] 100% FINE ; Home position
DO[1]=ON ;
J P[3] 100% CNT100 ;
J P[6] 100% CNT100 ;
J P[1] 100% FINE ; Pick01
UTOOL_NUM=3 ;
DO[1]=OFF ;
L P[2] 3000mm/sec CNT100 ;
J P[4] 100% CNT100 ;
J P[7] 100% CNT100 ;
DO[2]=ON ;
J P[8] 100% CNT100 ;
L P[9] 3000mm/sec FINE ; Drop01
DO[3]=ON ;
UFRAME_NUM=0 ;
UTOOL_NUM=2 ;
L P[10] 3000mm/sec CNT100 ;
J P[11] 100% CNT100 ;
J P[12] 100% CNT100 ;
DO[1]=ON ;
J P[14] 100% CNT100 ;
J P[15] 100% FINE ; Pick02
UTOOL_NUM=3 ;
DO[1]=OFF ;
L P[16] 3000mm/sec CNT100 ;
J P[17] 100% CNT100 ;

DO[4]=ON ;
J P[18] 100% CNT100 ;
L P[19] 3000mm/sec FINE ; Drop02
DO[5]=ON ;
UFRAME_NUM=0 ;
UTOOL_NUM=2 ;
L P[13] 3000mm/sec CNT100 ;
J P[20] 100% CNT100 ;
J P[21] 100% CNT100 ;
DO[1]=ON ;
J P[22] 100% CNT100 ;
J P[23] 100% FINE ; Pick03
UTOOL_NUM=3 ;
DO[1]=OFF ;
L P[24] 3000mm/sec CNT100 ;
J P[25] 100% CNT100 ;
DO[6]=ON ;
J P[26] 100% CNT100 ;
J P[27] 100% CNT100 ;
L P[28] 3000mm/sec FINE ; Drop03
DO[7]=ON ;
UFRAME_NUM=0 ;
UTOOL_NUM=2 ;
J P[29] 100% CNT100 ;

```

PROCESO R1 – 02 (UNIÓN BASE PALLET CLAVOS)

Desde la última posición creada a la salida del pick and place de la base del pale vamos a simular ahora la operación de unión a través de clavos en el pale para fijar los tacos a los listones de la base del pale.

Pasamos a crear todas las trayectorias siempre siguiendo la norma de los puntos de trabajo en Fine y esta vez utilizamos lineales como trabajo normal ya que la herramienta de trabajo debe permanecer lo más perpendicular posible a la superficie de trabajo en toda la operación para facilitar la tarea y/o ahorrar tiempo de ciclo.

```

UFRAME_NUM=0 ;
85: UTOOL_NUM=2 ;
86:J P[29] 100% CNT100 ;
87:J P[30] 100% CNT100 ;
88:L P[31:NAIL01] 3000mm/sec FINE ;
89:L P[32] 3000mm/sec CNT100 ;
90:L P[33] 3000mm/sec CNT100 ;
91:L P[34:NAIL02] 100mm/sec FINE ;
92:J P[35] 100% CNT100 ;
93:J P[37] 100% CNT100 ;
94:L P[38:NAIL03] 3000mm/sec FINE ;
95:L P[39] 3000mm/sec CNT100 ;
96:L P[40] 3000mm/sec CNT100 ;
97:L P[41:NAIL04] 3000mm/sec FINE ;
98:L P[42] 3000mm/sec CNT100 ;
99:L P[36] 3000mm/sec CNT100 ;
100:L P[43:NAIL05] 3000mm/sec FINE ;
101:L P[44] 3000mm/sec CNT100 ;
102:L P[45] 3000mm/sec CNT100 ;
103:L P[46:NAIL06] 3000mm/sec FINE ;
104:L P[47] 3000mm/sec CNT100 ;
105:L P[48] 3000mm/sec CNT100 ;
106:L P[49:NAIL07] 3000mm/sec FINE ;
107:L P[50] 3000mm/sec CNT100 ;
108:L P[51] 3000mm/sec CNT100 ;
109:L P[52:NAIL08] 3000mm/sec FINE ;
110:L P[53] 3000mm/sec CNT100 ;
111:L P[54] 3000mm/sec CNT100 ;
112:L P[55:NAIL09] 3000mm/sec FINE ;
113:L P[56] 3000mm/sec CNT100 ;
114:L P[57] 3000mm/sec CNT100 ;
115:L P[58:NAIL10] 3000mm/sec FINE
116:L P[59] 3000mm/sec CNT100 ;
117:L P[60] 3000mm/sec CNT100 ;
118:L P[61:NAIL11] 3000mm/sec FINE ;
119:L P[62] 3000mm/sec CNT100 ;
120:L P[63] 3000mm/sec CNT100 ;
121:L P[64:NAIL12] 3000mm/sec FINE ;
122:L P[65] 3000mm/sec CNT100 ;
123:L P[66] 3000mm/sec CNT100 ;
124:L P[67:NAIL13] 3000mm/sec FINE ;
125:L P[68] 3000mm/sec CNT100 ;
126:L P[69] 3000mm/sec CNT100 ;
127:L P[70:NAIL14] 3000mm/sec FINE ;
128:L P[71] 3000mm/sec CNT100 ;
129:L P[72] 3000mm/sec CNT100 ;
130:L P[73:NAIL15] 3000mm/sec FINE ;
131:L P[74] 3000mm/sec CNT100 ;
132:L P[75] 3000mm/sec CNT100 ;
133:L P[76:NAIL16] 3000mm/sec FINE ;
134:L P[77] 3000mm/sec CNT100 ;
135:L P[78] 3000mm/sec CNT100 ;
136:L P[79:NAIL17] 3000mm/sec FINE ;
137:L P[80] 3000mm/sec CNT100 ;
138:L P[81] 3000mm/sec CNT100 ;
139:L P[82:NAIL18] 3000mm/sec FINE ;
140:L P[83] 100mm/sec CNT100 ;
141:J P[85:WAIT.POS] 100% FINE ;

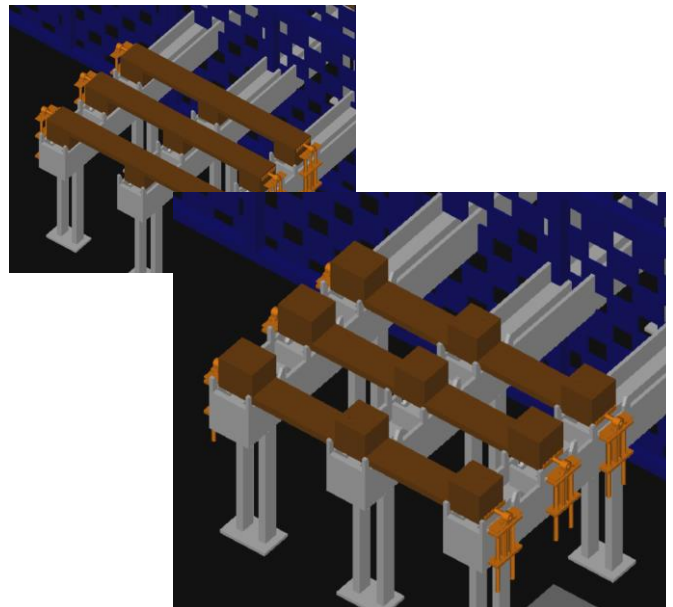
```

En este momento la capa inferior del pallet queda terminada gracias a la unión de los 9 tacos inferiores a 3 listones a través de los dos procesos anteriores. Es ahora cuando vamos a pasar a insertar una pequeña simulación de movimientos de pieza con origen en la mesa de trabajo.

Creamos elementos necesarios para poder simular la apertura de los cilindros de rotación (volteadores) así como el movimiento de elevación de los 3 listones que forman la base, la rotación de éstos y vuelta a la posición de pick ya girados 180 grados y listos para poder colocarlos directamente sobre la fixture central en posición correcta para continuar con el ensamblaje del pallet.

La secuencia de entradas digitales va a ser la siguiente:

143: DO[6]=OFF ;	} Desactivar tacos + listón 01	172: ;	} L01 Girado abajo
144: DO[7]=OFF ;		173: DO[11]=OFF ;	
145: DO[8]=OFF ;		174: DO[14]=OFF ;	
146: DO[11]=ON ;	} Activar mecanismo y subir	175: DO[8]=ON ;	} L01 Girado abajo
147: DO[20]=ON ;		176: DO[17]=ON ;	
148: WAIT .25(sec) ;		177: WAIT .25(sec) ;	
149: DO[4]=OFF ;	} Desactivar tacos + listón 02	178: DO[12]=OFF ;	} L01 Girado abajo
150: DO[5]=OFF ;		179: DO[15]=OFF ;	
151: DO[9]=OFF ;		180: DO[9]=ON ;	
152: DO[12]=ON ;	} Activar mecanismo y subir	181: DO[18]=ON ;	} L01 Girado abajo
153: DO[21]=ON ;		182: WAIT .25(sec) ;	
154: WAIT .25(sec) ;		183: DO[13]=OFF ;	
155: DO[2]=OFF ;	} Desactivar tacos + listón 03	184: DO[16]=OFF ;	} L01 Girado abajo
156: DO[3]=OFF ;		185: DO[10]=ON ;	
157: DO[10]=OFF ;		186: DO[19]=ON ;	
158: DO[13]=ON ;	} Activar mecanismo y subir	187: WAIT .25(sec) ;	
159: DO[22]=ON ;			
160: WAIT .25(sec) ;			
163: DO[20]=OFF ;	} L01 Girado arriba		
164: DO[14]=ON ;			
165: WAIT .25(sec) ;			
166: DO[21]=OFF ;	} L02 Girado arriba		
167: DO[15]=ON ;			
168: WAIT .25(sec) ;			
169: DO[22]=OFF ;	} L03 Girado arriba		
170: DO[16]=ON ;			
171: WAIT .25(sec) ;			



Figuras 83 y 84 Volteo listones

PROCESO r1 – 03 (PICK AND PLACE BASE PALLET – FIXTURE)

Este proceso consiste en la carga sobre la fixture central de los elementos unidos anteriormente para poder continuar con el ensamblaje del pallet final.

Desde la posición creada llamada Wait pos. justo después de terminar el proceso de fijación mediante clavos se ejecuta la rotación de los componentes y una vez terminada continua el proceso de pick and place.

Para este proceso vamos a utilizar la herramienta creada con anterioridad (Tool 4) para el Robot 1 que contiene la pieza ya girada sobre la herramienta.

Así pues creamos los puntos correspondientes utilizando Joints en continuo para las trayectorias y Lineales en Fine para los puntos de trabajo; los puntos donde se detiene el robot para efectuar una cogida o dejada de pieza.

Necesitamos también las piezas que van a ser visualizadas sobre el fixture central en cada uno de los movimientos de drop. Creamos para ello las respectivas piezas con sus señales digitales asociadas (DO 23, DO 24 y DO 25) las cuales cambiarán su estado a On justo en el mismo instante cuando el robot efectúe su cambio de herramienta.

141:J P[85:WAIT.POS] 100% FINE ;
 Señales de rotación piezas (141 – 189)
 189:J P[84] 100% CNT100 ;
 190:L P[86] 3000mm/sec FINE ;
 191: UFRAME_NUM=0 ;
 192: UTOOL_NUM=4 ;
 193: DO[19]=OFF ;
 194:J P[87] 100% CNT100 ;
 195:J P[88] 100% CNT100 ;
 196:J P[89] 100% CNT100 ;
 197:L P[90] 3000mm/sec FINE ;
 198: DO[23]=ON ;
 199: UFRAME_NUM=0 ;
 200: UTOOL_NUM=2 ;
 201:L P[91] 3000mm/sec CNT100 ;
 202:J P[92] 100% CNT100 ;
 203:L P[93] 3000mm/sec FINE ;
 204: UFRAME_NUM=0 ;
 205: UTOOL_NUM=4 ;
 206: DO[18]=OFF ;
 207:L P[94] 3000mm/sec CNT100 ;
 208:J P[95] 100% CNT100 ;
 209:J P[97] 100% CNT100 ;

} Pick listón girado 01

} Drop listón girado 01

} Pick listón girado 02

```
210:J P[96] 100% FINE ;  
211: DO[24]=ON ;  
212: UFRAME_NUM=0 ;  
213: UTOOL_NUM=2 ; } Drop listón girado 02  
214:J P[98] 100% CNT100 ;  
215:J P[100] 100% CNT100 ;  
216:L P[99] 3000mm/sec FINE ;  
217: UFRAME_NUM=0 ;  
218: UTOOL_NUM=4 ; } Pick listón girado 03  
219: DO[17]=OFF ;  
220:J P[101] 100% CNT100 ;  
221:J P[102] 100% CNT100 ;  
222:J P[103] 100% CNT100 ;  
223:L P[104] 3000mm/sec FINE ;  
224: DO[25]=ON ;  
225: UFRAME_NUM=0 ;  
226: UTOOL_NUM=2 ; } Drop listón girado 03  
227:J P[105] 100% CNT100 ;  
228: ;
```

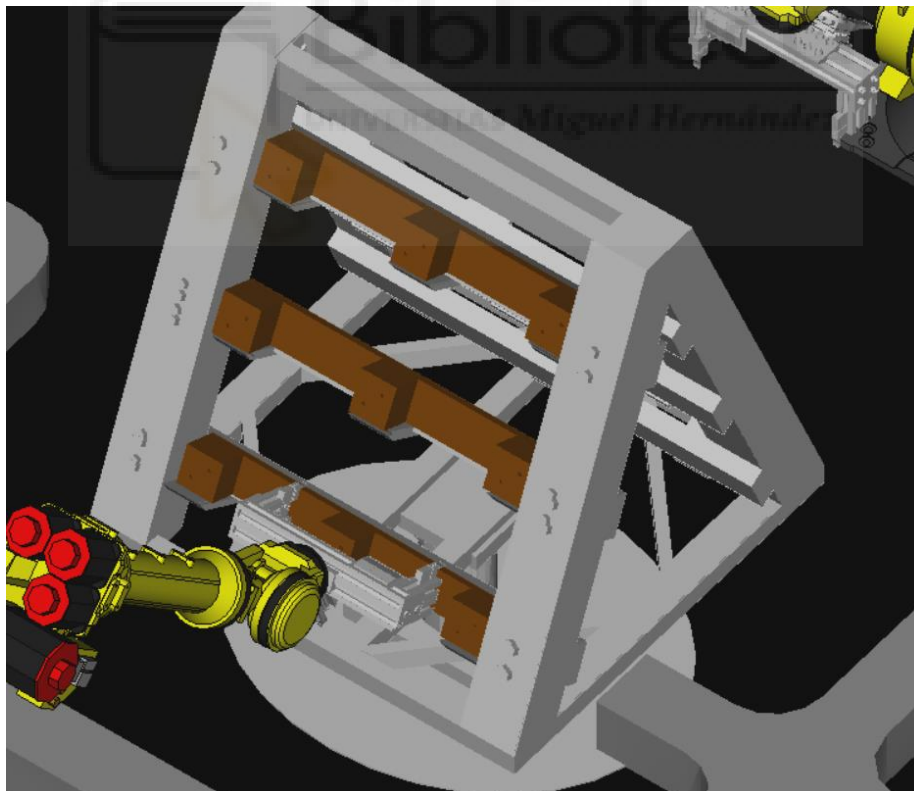


Figura 85 Colocación primeros elementos

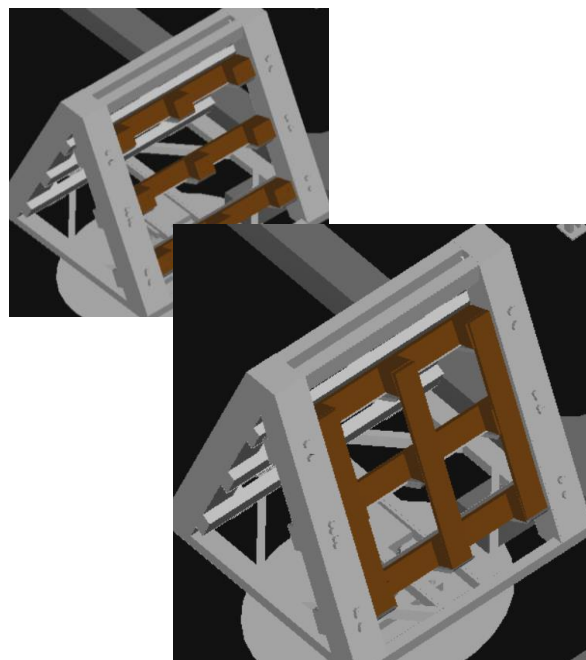
PROCESO r1 – 04 (PICK AND PLACE CAPA INTERMEDIA – FIXTURE)

Éste es el último proceso de robot 01 y consiste en completar la mitad del pallet uniendo 3 listones más perpendiculares a la dirección de los primeros montados sobre la fixture central de tal forma que se completa la estructura básica del pallet.

De la misma forma que el Proceso r1 – 01 activamos la señal digital de entrada de listones (DO 1) y creamos los movimientos para simular el pick de este listón.

Una vez sobre el fixture central necesitamos las respectivas señales digitales que activarán la visualización de las piezas colocadas. Como siempre creamos dichos componentes estableciendo la debida señal digital de actuación (DO31 , DO32 y DO33)

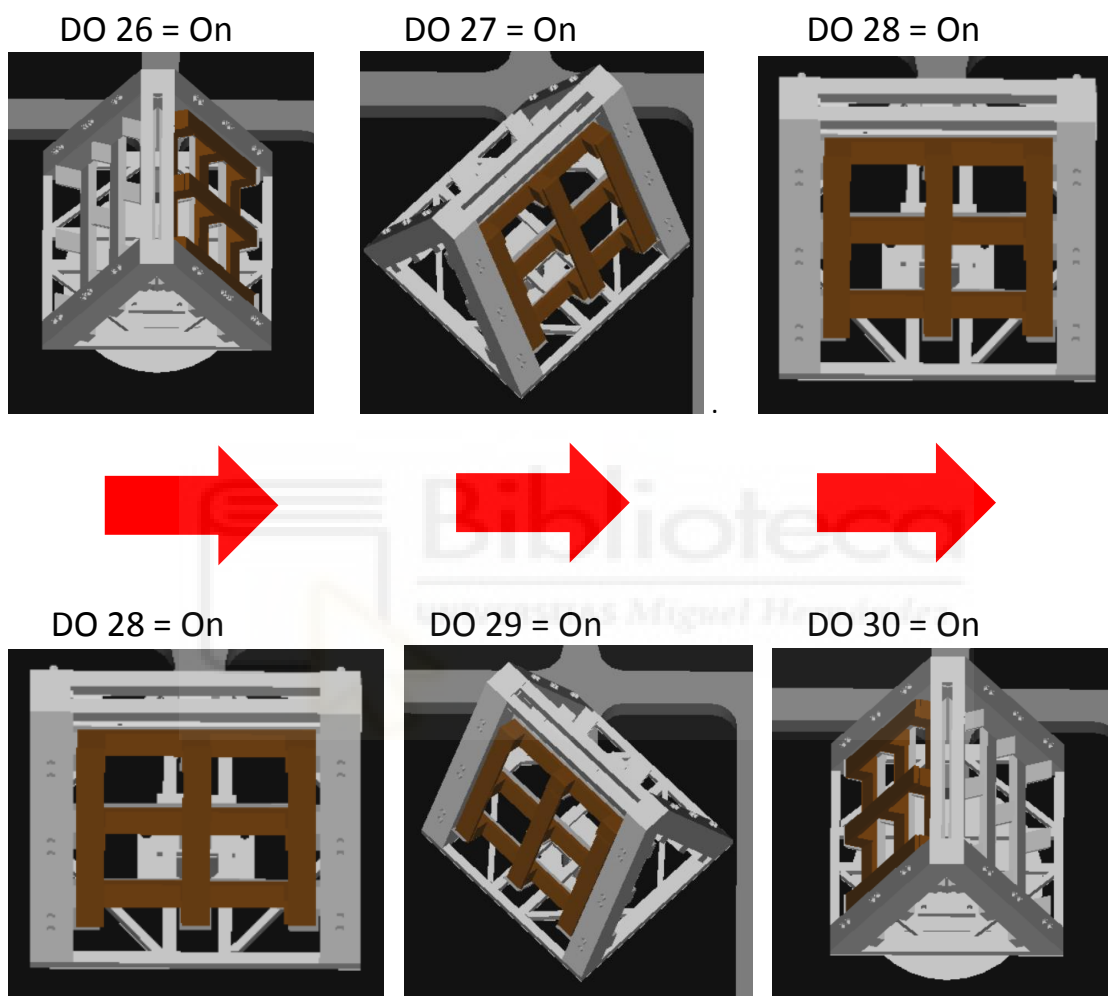
229: DO[1]=ON ; Activar entrada listón	256:J P[121] 100% CNT100 ;
230:J P[107] 100% CNT100 ;	257: DO[1]=ON ; Activar entrada listón
231:J P[108] 100% CNT100 ;	258:J P[122] 100% CNT100 ;
232:J P[109] 100% FINE ;	259:J P[123] 100% FINE ;
233: UTOOL_NUM=3 ;	260: UTOOL_NUM=3 ;
234: DO[1]=OFF ;	261: DO[1]=OFF ;
235:L P[110] 3000mm/sec CNT100 ;	262:L P[124] 3000mm/sec CNT100 ;
236:J P[111] 100% CNT100 ;	263:J P[125] 100% CNT100 ;
237:J P[112] 100% CNT100 ;	264:J P[126] 100% CNT100 ;
238:L P[113] 3000mm/sec FINE ;	265:L P[127] 3000mm/sec FINE ;
239: DO[31]=ON ; Drop listón	266: DO[33]=ON ; Drop listón
240: UFRAME_NUM=0 ;	267: UFRAME_NUM=0 ;
241: UTOOL_NUM=2 ;	268: UTOOL_NUM=2 ;
242:L P[114] 3000mm/sec CNT100 ;	269:J P[128] 100% CNT100 ;
244: DO[1]=ON ; Activar entrada listón	270:J P[129] 100% FINE ;
245:J P[116] 100% CNT100 ;	
246:J P[117] 100% FINE ;	
247: UTOOL_NUM=3 ;	
248: DO[1]=OFF ;	
249:L P[118] 3000mm/sec CNT100 ;	
250:J P[120] 100% CNT100 ;	
251:J P[119] 100% CNT100 ;	
252:L P[115] 3000mm/sec FINE ;	
253: DO[32]=ON ; Drop listón	
254: UFRAME_NUM=0 ;	
255: UTOOL_NUM=2 ;	



Figuras 86 y 87 Proceso final robot 01

La fase final del robot 01 se trata del control de la rotación del fixture central una vez ensamblados los componentes necesarios para continuar el proceso con el robo 02.

Como viene siendo habitual creamos la misma fixture en diferentes posiciones y el procedimiento será activar y desactivar señales digitales de tal forma que aparentemente la fixture central será capaz de girar 180 grados progresivamente.



Figuras 88 a 93 Volteo mesa ensamble

Una vez la fixture ha girado el proceso sigue con el otro robot que acabará de fabricar el pallet pero el robot 01 deberá empezar la fabricación del siguiente pallet así que insertamos un "jump label" hacia el inicio del código para volver a ejecutar cada una de las tareas.

Al hacer el salto al inicio del programa reestablecemos todas las señales digitales que vamos a utilizar cambiando su estado (sea cual sea) a Off y reiniciando todos los procesos del robot 01 (r1).

PROCESO r2 – 05 (PICK AND PLACE CAPA FINAL – FIXTURE)

Una vez concluida la primera parte de la simulación realizada por el robot 01 (r1) y después de la rotación del fixture de montaje central es momento ahora de terminar el ensamblaje del pallet gracias a la actuación del robot 02 (r2).

El primer proceso de éste robot será colocar los listones necesarios para terminar el ensamblaje (exactamente 7 listones) en la capa superior del pallet. Gracias a que ya tenemos creada la salida digital de entrada de material podemos utilizar ahora esta señal para simular el pick de entrada de listones.

Creamos un programa llamado simulation 01 y posteriormente añadimos unas líneas de código. Una vez hecho esto creamos un primer punto en joint así como la definición de tool y frame a utilizar exactamente igual que hemos hecho ya en el primer robot y al inicio de la simulación.

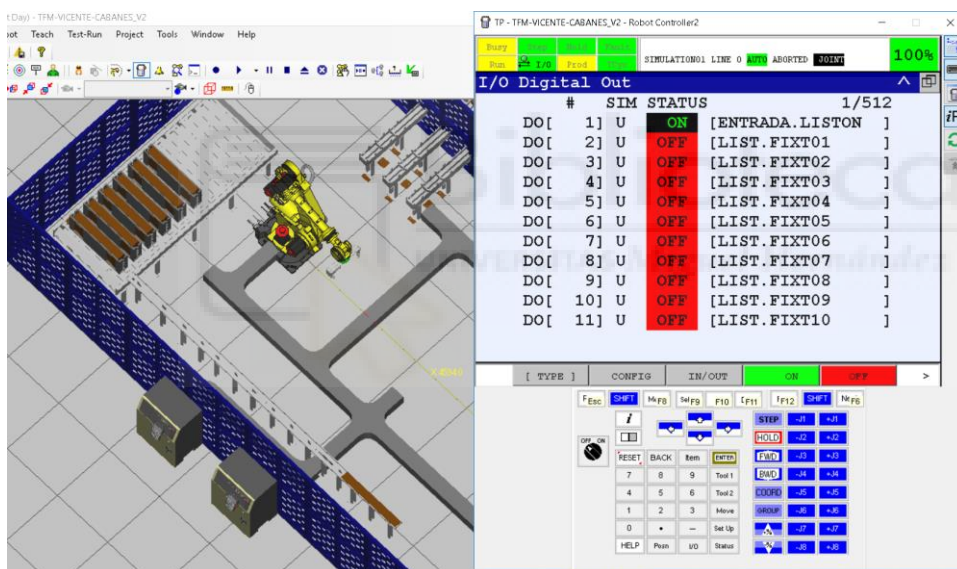


Figura 94 Entrada material robot 02

Una vez creado el primer punto activamos la señal digital correspondiente a la entrada de material (DO 1= On) y observamos como aparece el listón deseado en la posición donde vamos a efectuar la cogida del listón por parte de la herramienta del robot.

Creamos un punto de aproximación usando Joint continuous y posteriormente el punto final en Lineal y Fine donde deberá de coincidir la posición del listón creado en el tool 3 en apartados anteriores con la posición donde se encuentra actualmente sobre la cinta de transporte.

Una vez en este punto utilizamos el recuso que venimos utilizando para simular el proceso de cogida cambiando de herramienta seleccionada y al mismo tiempo desactivando la señal digital que controla la entrada del material (DO 1 = Off)

Ahora sí tenemos la pieza sobre la herramienta del robot así que creamos las trayectorias necesarias para realizar el proceso de dejada (drop) sobre el fixture central de cada uno de los listones necesarios para terminar la última capa del pallet.

Cada una de las piezas colocadas sobre la fixture debe tener la respectiva señal digital que active o desactive la visualización del componente montado sobre la fixture central. En este caso se utiliza desde la DO 5 hasta la DO 11 ambas incluidas de tal forma que en cada drop activaremos la señal correspondiente simultáneamente al cambio de herramienta seleccionado simulando así el proceso de dejar elementos.

Repetiremos el proceso de pick utilizando **exactamente los mismos puntos creados** y el mismo proceso (podemos copiar y pegar código)

```

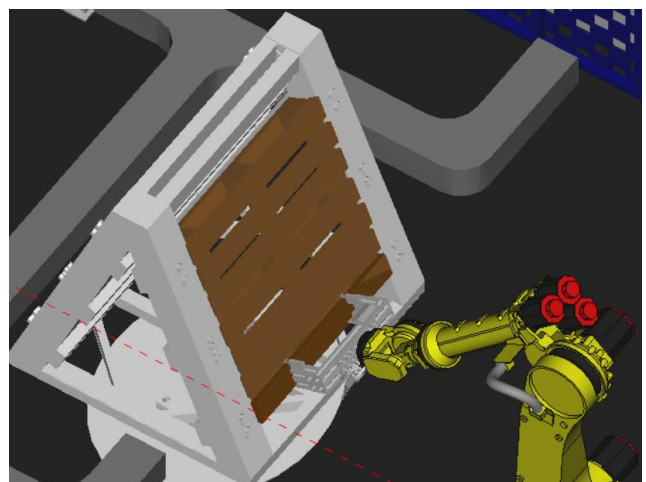
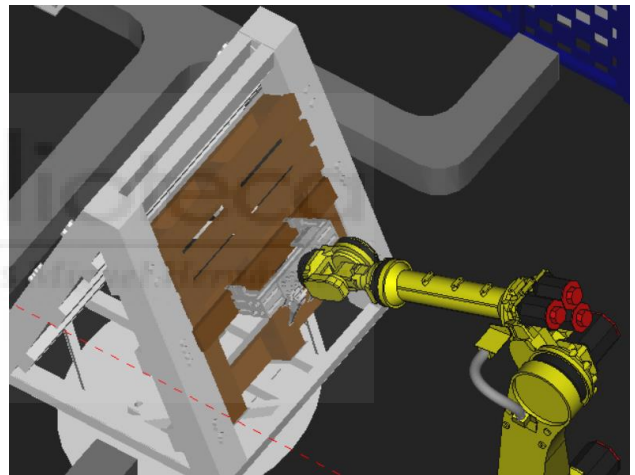
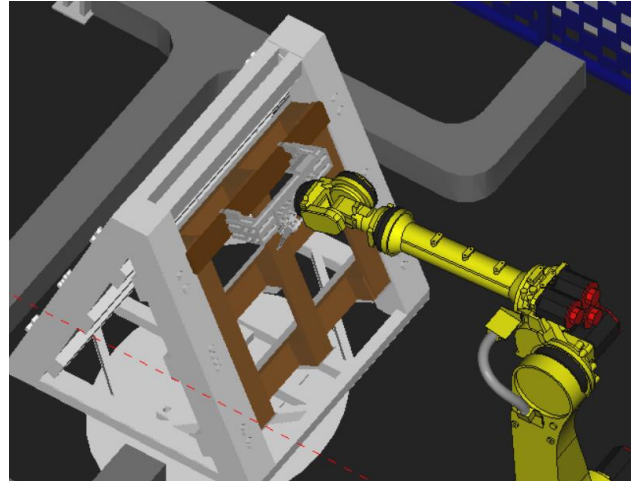
15: UFRAME_NUM=0 ;
16: UTOOL_NUM=2 ;
17: J P[147] 100% FINE ;
24: J P[85] 100% CNT100 ;
25: DO[1:ENTRADA.LISTON]=ON ;
26: J P[86] 100% CNT100 ;
27: L P[87] 3000mm/sec FINE ; Pick 01
28: UFRAME_NUM=0 ;
29: UTOOL_NUM=3 ;
30: DO[1:ENTRADA.LISTON]=OFF ;
31: L P[88] 3000mm/sec CNT100 ;
32: J P[89] 100% CNT100 ;
33: J P[90] 100% FINE
45: L P[91:L1] 3000mm/sec FINE ;
46: DO[5:LIST.FIXT04]=ON ; Drop 01
47: UFRAME_NUM=0 ;
48: UTOOL_NUM=2 ;
49: J P[92] 100% CNT100 ;
50: J P[93] 100% CNT100 ;
51: DO[1:ENTRADA.LISTON]=ON ;
52: J P[94] 100% CNT100 ;
53: L P[95] 3000mm/sec FINE; Pick 02
54: UFRAME_NUM=0 ;
55: UTOOL_NUM=3 ;
56: DO[1:ENTRADA.LISTON]=OFF ;
57: L P[96] 3000mm/sec CNT100 ;
58: J P[97] 100% CNT100 ;
59: J P[98] 100% CNT100 ;
60: L P[99:L2] 3000mm/sec FINE ;
61: DO[6:LIST.FIXT05]=ON ; Drop 02
62: UFRAME_NUM=0 ;
95: J P[117] 100% CNT100 ;

63: UTOOL_NUM=2 ;
64: J P[100] 100% CNT100 ;
65: J P[101] 100% CNT100 ;
66: DO[1:ENTRADA.LISTON]=ON ;
67: J P[102] 100% CNT100 ;
68: L P[103] 3000mm/sec FINE; Pick 03
69: UFRAME_NUM=0 ;
70: UTOOL_NUM=3 ;
71: DO[1:ENTRADA.LISTON]=OFF ;
72: L P[104] 3000mm/sec CNT100 ;
73: J P[105] 100% CNT100 ;
74: J P[106] 100% CNT100 ;
75: L P[107:L3] 3000mm/sec FINE ;
76: DO[7:LIST.FIXT06]=ON ; Drop 03
77: UFRAME_NUM=0 ;
78: UTOOL_NUM=2 ;
79: J P[108] 100% CNT100 ;
80: J P[109] 100% CNT100 ;
81: DO[1:ENTRADA.LISTON]=ON ;
82: J P[110] 100% CNT100 ;
83: L P[111] 3000mm/sec FINE; Pick 04
84: UFRAME_NUM=0 ;
85: UTOOL_NUM=3 ;
86: DO[1:ENTRADA.LISTON]=OFF ;
87: L P[112] 3000mm/sec CNT100 ;
88: J P[113] 100% CNT100 ;
89: J P[114] 100% CNT100 ;
90: J P[115] 100% FINE ;
91: DO[8:LIST.FIXT07]=ON ; Drop 04
92: UFRAME_NUM=0 ;
93: UTOOL_NUM=2 ;
94: J P[116] 100% CNT100 ;

```

```

96: DO[1:ENTRADA.LISTON]=ON ;
97: J P[118] 100% CNT100 ;
98: L P[119] 3000mm/sec FINE ; Pick 05
99: UFRAME_NUM=0 ;
100: UTOOL_NUM=3 ;
101: DO[1:ENTRADA.LISTON]=OFF ;
102: L P[120] 3000mm/sec CNT100 ;
103: J P[121] 100% CNT100 ;
104: J P[122] 100% CNT100 ;
105: J P[123] 100% FINE ;
106: DO[9:LIST.FIXT08]=ON ; Drop 05
107: UFRAME_NUM=0 ;
108: UTOOL_NUM=2 ;
109: J P[124] 100% CNT100 ;
110: J P[125] 100% CNT100 ;
111: DO[1:ENTRADA.LISTON]=ON ;
112: J P[126] 100% CNT100 ;
113: L P[127] 3000mm/sec FINE; Pick 06
114: UFRAME_NUM=0 ;
115: UTOOL_NUM=3 ;
116: DO[1:ENTRADA.LISTON]=OFF ;
117: L P[128] 3000mm/sec CNT100 ;
118: J P[129] 100% CNT100 ;
119: J P[130] 100% CNT100 ;
120: J P[131] 100% FINE ;
121: DO[10:LIST.FIXT09]=ON ; Drop 06
122: UFRAME_NUM=0 ;
123: UTOOL_NUM=2 ;
124: J P[132] 100% CNT100 ;
125: J P[133] 100% CNT100 ;
126: DO[1:ENTRADA.LISTON]=ON ;
127: J P[134] 100% CNT100 ;
128: L P[135] 3000mm/sec FINE; Pick 07
129: UFRAME_NUM=0 ;
130: UTOOL_NUM=3 ;
131: DO[1:ENTRADA.LISTON]=OFF ;
132: L P[136] 3000mm/sec CNT100 ;
133: J P[137] 100% CNT100 ;
134: J P[138] 100% CNT100 ;
135: L P[139] 3000mm/sec FINE ;
136: DO[11:LIST.FIXT10]=ON ; Drop 07
137: UFRAME_NUM=0 ;
138: UTOOL_NUM=2 ;
139: J P[140] 100% CNT100 ;
  
```



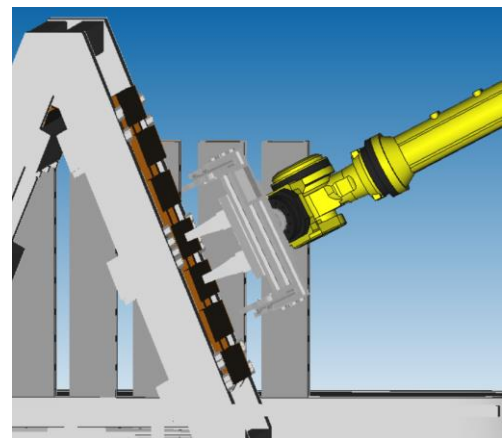
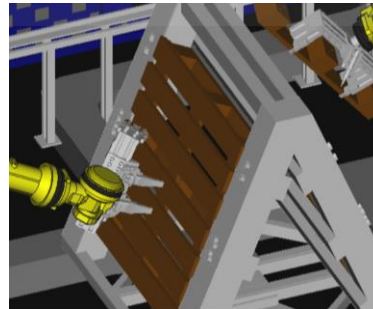
Figuras 95 a 97 Ensamblaje capa final pale

PROCESO r2 – 06 (UNIÓN CAPA INTERMEDIA Y FINAL GRAPAS)

Ésta última unión se realiza utilizando una herramienta equipada con dos pistolas de grapas que permiten reducir el tiempo de ciclo de éste proceso a la mitad que utilizando la pistola de clavos convencional.

Creamos todas las trayectorias en lineal ya que debemos mantener la orientación de la herramienta perpendicular a la superficie de aplicación (el pallet). Utilizamos lineales en continuo para trasladarnos de un punto a otro y lineales en fine para los puntos de trabajo donde la herramienta va a actuar y por lo tanto el robot deberá detenerse en dichos puntos.

140:J P[27] 100% CNT100 ;	166:L P[55:NAIL09] 3000mm/sec FINE ;
141:J P[28] 100% CNT100 ;	167:L P[56] 3000mm/sec CNT100 ;
142:L P[29:NAIL01] 3000mm/sec FINE ;	168:L P[57] 3000mm/sec CNT100 ;
143:L P[30] 3000mm/sec CNT100 ;	169:L P[58:NAIL10] 3000mm/sec FINE ;
144:L P[31] 3000mm/sec CNT100 ;	170:L P[59] 3000mm/sec CNT100 ;
145:L P[32:NAIL02] 3000mm/sec FINE ;	171:L P[60] 3000mm/sec CNT100 ;
146:L P[33] 3000mm/sec CNT100 ;	172:L P[61:NAIL11] 3000mm/sec FINE ;
147:L P[34] 3000mm/sec CNT100 ;	173:L P[62] 3000mm/sec CNT100 ;
148:L P[35:NAIL03] 3000mm/sec FINE ;	174:L P[63] 3000mm/sec CNT100 ;
149:L P[36] 3000mm/sec CNT100 ;	175:L P[64:NAIL12] 3000mm/sec FINE ;
150:L P[2] 3000mm/sec CNT100 ;	176:L P[65] 3000mm/sec CNT100 ;
151:L P[37:NAIL04] 3000mm/sec FINE ;	
152:L P[38] 3000mm/sec FINE ;	
153:L P[39] 3000mm/sec CNT100 ;	
154:L P[40:NAIL05] 3000mm/sec FINE ;	
155:L P[41] 3000mm/sec CNT100 ;	
156:L P[42] 3000mm/sec CNT100 ;	
157:L P[43:NAIL06] 3000mm/sec FINE ;	
158:L P[47] 3000mm/sec CNT100 ;	
159:L P[48] 3000mm/sec CNT100 ;	
160:L P[49:NAIL07] 3000mm/sec FINE ;	
161:L P[50] 3000mm/sec CNT100 ;	
162:L P[51] 3000mm/sec CNT100 ;	
163:L P[52:NAIL08] 3000mm/sec FINE ;	
164:L P[53] 3000mm/sec CNT100 ;	
165:J P[54] 100% CNT100 ;	



Figuras 98 y 99 Proceso fijación grapas

PROCESO r2 – 07 (PICK AND PLACE PRODUCTO FINAL)

El proceso final de la simulación consiste en sacar el pallet terminado del fixture y depositarlo sobre la cinta de transporte de salida de la celda de producción.

Para este proceso vamos a utilizar la última herramienta creada con anterioridad para el robot 02 (tool 04), llevando el robot a la posición de pick y creando un punto en Fine donde el robot cambiará su herramienta y a la vez desactivamos las señales que representan todos los elementos montados sobre la fixture central (DO 2 a DO 12).

A continuación se crea la trayectoria desde la fixture hasta la cinta de salida utilizando puntos en Joint y continuos para por último detener el robot en un punto Fine donde simulará el proceso de dejada (drop) sobre la cinta final.

Como siempre necesitamos los elementos que simulan la dejada en forma de señales digitales para poder activarlas en el justo momento cuando el robot cambie a la herramienta de trabajo habitual y simule la dejada del pallet sobre la cinta. Creamos para este último proceso la señal digital DO 13 que cambiaremos de estado a On en el último momento simulando la dejada del pallet.

```

178:J P[84] 100% CNT100 ;
179:L P[44] 3000mm/sec FINE ;
180: UFRAME_NUM=0 ;
181: UTOOL_NUM=4 ;
182: DO[2:LIST.FIXT01]=OFF ;
183: DO[3:LIST.FIXT02]=OFF ;
184: DO[4:LIST.FIXT03]=OFF ;
185: DO[5:LIST.FIXT04]=OFF ;
186: DO[6:LIST.FIXT05]=OFF ;
187: DO[7:LIST.FIXT06]=OFF ;
188: DO[8:LIST.FIXT07]=OFF ;
189: DO[9:LIST.FIXT08]=OFF ;
190: DO[10:LIST.FIXT09]=OFF ;
191: DO[11:LIST.FIXT10]=OFF ;
192: DO[12:CLEAR]=OFF ;
193:J P[141] 100% CNT100 ;
194:J P[142] 100% CNT100 ;
195:J P[143] 100% CNT100 ;
196:L P[144:DROP] 3000mm/sec FINE ;
197: DO[13:FINAL]=ON ;
198: UFRAME_NUM=0 ;
199: UTOOL_NUM=2 ;
200:L P[145] 3000mm/sec CNT100 ;
201:J P[146] 100% FINE ;

```

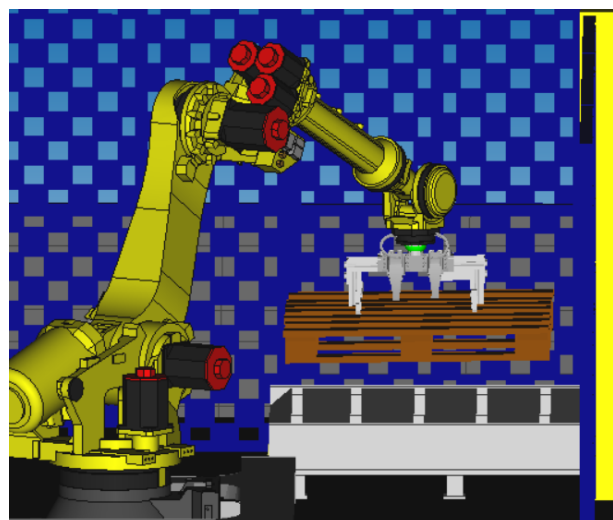


Figura 100 y 101 Movimientos salida producto final

AJUSTAR CICLO

Una vez realizada la simulación es momento de ajustar el ciclo con los parámetros necesarios para poder realizar un ciclo continuo de producción lo más cercano a la realidad del proceso.

El primer paso para conseguir un ciclo continuo es colocar la instrucción de jump label al final de cada código de producción y su respectivo label al principio del código de tal forma que al terminar el proceso vuelva a empezar cada vez.

Después de esto uno de los aspectos más importantes a tener en cuenta se trata del ciclo de llenado. Éste ciclo se produce en todas las celdas de producción donde los procesos productivos de los robots no son independientes, es decir un robot espera la finalización del proceso del anterior para poder ejercer su labor.

En este caso el robot 02 debe esperar a la finalización del proceso del robot 01 para poder arrancar su proceso para ello vamos a colocar un wait que sólo debe ejecutarse en el primer ciclo es decir al arrancar la producción.

Independientemente del ciclo de llenado el robot 01 es más lento en la producción en continuo por lo que debemos crear la manera de actuar de "PLC" de control de ciclo estableciendo un tiempo de espera del robot 02 al robot 01 para poder arrancar su producción. Para ello utilizamos los siguientes recursos:

<pre>13: R[1]=0 19: WAIT 50(sec) ; 21: LBL[200] ;</pre>	}	<p>INICIO DE CICLO Establecer valor registro 1 a cero Esperar 50 segundos (sólo el primer ciclo) El segundo ciclo salto al label 200</p>
<pre>35: IF R[1]=0,JMP LBL[300] ; 36: WAIT 5.10(sec) ; 37: LBL[300] ;</pre>	}	<p>INICIO CICLO Esperar 5.10 segundos si el registro 1 es diferente de cero Sólo espera si ya he pasado por el final del ciclo</p>
<pre>204: R[1]=R[1]+1 ; 205: JMP LBL[200] ;</pre>	}	<p>FINAL CICLO Sumar 1 al registro numero 1</p>

Por otra parte el robot 01 nunca se detiene, es decir puede estar en producción sin esperas y es éste robot el que marcará el tiempo de ciclo obtenido en la celda de producción ya que es el robot más lento el que

PROTOTIPADO DE LA ESTACIÓN

Un prototipo es un modelo (representación, demostración o simulación) fácilmente ampliable y modificable de un sistema planificado, probablemente incluyendo su interfaz y su funcionalidad de entradas y salidas.

El prototipado modela el producto final y permite efectuar un test sobre determinados atributos del mismo sin necesidad de que está disponible. Se trata, simplemente, de testar haciendo uso del modelo.

Es conocido en el mundo del prototipado rápido el caso de un fabricante de ascensores que, tras un proceso de implementación de la tecnología de impresión 3D, ha conseguido un ahorro en costes de producción del 98% tan sólo en su fase de prototipado. El proceso es, sin duda, un ejemplo práctico de cómo una pequeña inversión para incorporar la impresión 3D en cualquier actividad empresarial puede ayudar al desarrollo y mejora de la empresa, y hacer de ella un negocio rentable.

La tecnología de Modelado de fusión por deposición (FDM) es considerada a menudo el método existente más sencillo. La tecnología de modelado por deposición fundida o FDM se basa en 3 elementos principales: una placa/cama de impresión en la que se imprime la pieza, una bobina de filamento que sirve como material de impresión y una cabeza de extrusión también llamada extrusor. En resumen, el filamento es succionado y fundido por el extrusor de la impresora 3D, que deposita el material de forma precisa capa por capa sobre la cama de impresión.

La construcción de una maqueta utilizando esta técnica permite sin duda visualizar el resultado final de la simulación en cuanto a la disposición de los elementos y distribución de espacios se refiere, además ser un elemento que transmite mucha información visual de forma rápida permitiendo conocer y entender el funcionamiento de la celda robotizada más rápida y eficientemente.

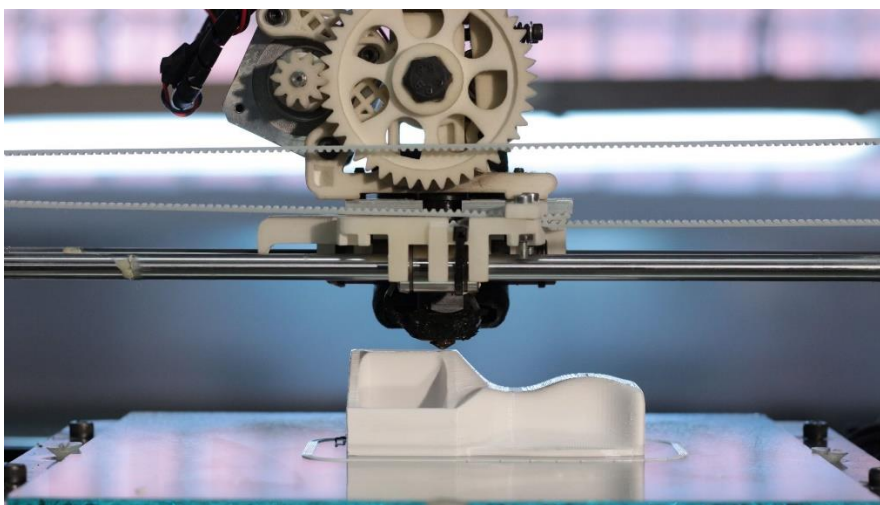


Figura 102 Impresión FDM

BLOQUE 03 BIBLIOGRAFIA Y ANEXOS

- 1 Bibliografía
- 2 Programa offline Matlab
- 3 Programa offline robot 01 Roboguide
- 4 Programa offline robot 02 Roboguide
- 5 Hoja características técnicas robots
- 6 Planos elementos



BIBLIOGRAFIA

WEBSITES:

<https://www.fanuc.eu/es/es/robots/accesorios/roboguide>
<https://es.scribd.com/doc/207315378/Tutorial-ROBOGUIDE-Fanuc>
<https://repositori.udl.cat/bitstream/handle/10459.1/45763/Anerillas.pdf?sequence=1>
<https://www.fanuc.eu/es/es/robots/p%C3%A1gina-filtro-robots/serie-r-2000/r-2000ic-165f>
<https://es.scribd.com/doc/135863075/Roboguide-Training-Manual-FRDE-Z-KAE-TRN-Roboguide-1-01-en>
<https://www.maderea.es/curiosidades-sobre-los-palets-de-madera/>
<https://impresiontresde.com/prototipado-rapido-impresion-3d-ventajas/>
<https://www.3dnatives.com/es/modelado-por-deposicion-fundida29072015/>

MANUALES:

Practical session 5: Rapid programming. ARTE.
Arturo Gil Aparicio

Tema 8: Introducción a Rapid. Control de robots y sistemas sensoriales. Universidad
Arturo Gil Aparicio

Seat Robótica Programación Fanuc
Autor: FANUC Robotics Ibérica S.L.
Actualización y revisión: Salvador García Chinchilla

Fanuc Robot R-2000iB Mechanical Unit
Operator's Manual

Memorias del XVII Congreso Mexicano de Robótica 2015
Instituto Tecnológico de la Paz

Simuladores en Robótica
Trabajo realizado para la asignatura de Introducción al Diseño de Micro robots Móviles
de la Universidad de Alcalá

Manual de RoboWorks
Carlos Rodríguez y Ana María Franco

Manual de Técnicas para el Diseño Participativo de Interfaces de Usuario de Sistemas basados
en Software y Hardware Universidad Zaragoza

PROGRAMA OFFLINE MATLAB

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Author: Vicente Cabanes Ribelles
% email: robot.cabanes@gmail.com
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

% CARGAR ESCENARIO

```

function kuka_pallet_simulation
global robot1 robot2

```

```

% Para que la ejecucion no sea tan lenta
configuration.delta_time=0.04;

```

%Cargar robots

```

robot1=load_robot('KUKA','KR90_R2700_pro');
robot2=load_robot('KUKA','KR90_R2700_pro');

```

%Para dibujar los ejes del robot

```

%Change 0 to 1 to see the axis on the robot
robot1.graphical.draw_axes = 0;
robot2.graphical.draw_axes = 0;

```

%Cargar entorno

```

%(Debe empezar en {1} y no saltarse ningun
numero)

```

%BASE ENTORNO

```

robot1.equipment{1}=load_robot('TFM', 'BASE');

```

%FENSE

```

robot1.equipment{2}=load_robot('TFM', 'PANEL1');
robot1.equipment{3}=load_robot('TFM', 'PANEL2');
robot1.equipment{4}=load_robot('TFM', 'PANEL');
robot1.equipment{5}=load_robot('TFM', 'PANEL6');
robot1.equipment{6}=load_robot('TFM', 'PANEL3');
robot1.equipment{7}=load_robot('TFM', 'PANEL4');
robot1.equipment{8}=load_robot('TFM', 'PANEL5');
robot1.equipment{9}=load_robot('TFM', 'PANEL7');
robot1.equipment{10}=load_robot('TFM', 'PANEL8');
robot1.equipment{11}=load_robot('TFM', 'PANEL9');
robot1.equipment{12}=load_robot('TFM', 'P_PUERTA');
robot1.equipment{13}=load_robot('TFM', 'PUERTA');
robot1.equipment{14}=load_robot('TFM', 'PANEL16');
robot1.equipment{15}=load_robot('TFM', 'PANEL10');
robot1.equipment{16}=load_robot('TFM', 'PANEL11');
robot1.equipment{17}=load_robot('TFM', 'PANEL12');
robot1.equipment{18}=load_robot('TFM', 'PANEL13');
robot1.equipment{19}=load_robot('TFM', 'PANEL14');
robot1.equipment{20}=load_robot('TFM', 'PANEL15');

```

%FIXTURE

```

robot1.equipment{21}=load_robot('TFM', 'FIXTURE');

```

%ENTRADA TACOS

```

robot1.equipment{22}=load_robot('TFM', 'CINTA');
robot1.equipment{23}=load_robot('TFM', 'MESA.TACO');
robot1.equipment{24}=load_robot('TFM', 'TOWER.TACO');
robot1.equipment{25}=load_robot('TFM', 'TOWER.TACO_02');
robot1.equipment{26}=load_robot('TFM', 'TOWER.TACO_03');
robot1.equipment{27}=load_robot('TFM', 'TOWER.TACO_04');
robot1.equipment{28}=load_robot('TFM', 'TOWER.TACO_05');
robot1.equipment{29}=load_robot('TFM', 'TOWER.TACO_06');
robot1.equipment{30}=load_robot('TFM', 'TOWER.TACO_07');
robot1.equipment{31}=load_robot('TFM', 'TOWER.TACO_08');
robot1.equipment{32}=load_robot('TFM', 'TOWER.TACO_09');

```

%ENTRADA LISTONES

```

robot1.equipment{33}=load_robot('TFM', 'CINTA.LARGA');
robot1.equipment{34}=load_robot('TFM', 'CONVEYOR-02');
robot1.equipment{35}=load_robot('TFM', 'TOWER.LISTON');
robot1.equipment{36}=load_robot('TFM', 'TOWER.LISTON.2');
robot1.equipment{37}=load_robot('TFM', 'TOWER.LISTON.3');
robot1.equipment{38}=load_robot('TFM', 'TOWER.LISTON.4');

```

%SALIDA PALLETS

```

robot1.equipment{39}=load_robot('TFM', 'CONVEYOR-01');
robot1.equipment{40}=load_robot('TFM', 'PALLET');

```

%PUERTAS CORREDERAS

```

robot1.equipment{41}=load_robot('TFM', 'CORREDERA-A');
robot1.equipment{42}=load_robot('TFM', 'CRISTAL-A');
robot1.equipment{43}=load_robot('TFM', 'CORREDERA-B');
robot1.equipment{44}=load_robot('TFM', 'CRISTAL-B');

```

%ELEMENTOS EXTRA

```

robot1.equipment{45}=load_robot('TFM', 'KABINET-01');
robot1.equipment{46}=load_robot('TFM', 'KABINET-02');
robot1.equipment{47}=load_robot('TFM', 'GUIA_CABLES');

```

%PIEZAS TRABAJO

```

%robot1.piece=load_robot('TFM','TACO');
robot1.equipment{48}=load_robot('TFM', 'TACO');
robot1.equipment{49}=load_robot('TFM', 'TACO02');
robot1.equipment{50}=load_robot('TFM', 'TACO03');
robot1.equipment{51}=load_robot('TFM', 'TACO04');
robot1.equipment{52}=load_robot('TFM', 'TACO05');
robot1.equipment{53}=load_robot('TFM', 'TACO06');
robot1.equipment{54}=load_robot('TFM', 'TACO07');
robot1.equipment{55}=load_robot('TFM', 'TACO08');
robot1.equipment{56}=load_robot('TFM', 'TACO09');
robot1.equipment{57}=load_robot('TFM', 'LISTON');

```

```
%Cargar tool del robot
robot1.tool=
load_robot('TFM','gripper_0000');
```

```
%Para dibujar los ejes de la tool
robot1.tool.graphical.draw_axes = 0;
```

```
%Place robots at a different base position
%Move along X and Y directions
robot1.T0= [0 1 0 -0.5; %Ultima columna
traslación eje x origen coordenadas
-1 0 0 0; % traslación eje y
0 0 1 0; % traslación eje z
0 0 0 0.9]; % escala del mdelo
```

```
robot2.T0= [0 -1 0 -0.5; %Ultima
columna traslación eje x origen
coordenadas
1 0 0 -3.5; % traslación eje y
0 0 1 0; % traslación eje z
0 0 0 0.9]; % escala del modelo
```

```
%Proporcionar valores articulares
q1= [0 0 0 0 -30 80];
q2= [0 0 0 0 0 0];
```

```
%Dibujar en un entorno 3d
drawrobot3d(robot1, q1)
drawrobot3d(robot2, q2, 1)
```

```
main() %Llamada programa principal
end8
```

```
%% SIMULACIÓN
function main
global TD_pallets robot1 robot2
```

```
%Ajustar vista deseada del escenario
%adjust_view(robot2);
disp('Orientar y pulsar una tecla para
empezar la simulación')
pause
```

```
movimientos_taco_01
movimientos_taco_02
movimientos_liston_01
end
```

```
function movimientos_taco_01
global TD_pallets robot1 robot2
```

```
%Proporcionar valores articulares
q1= [0 0 0 0 -30 80];
q2= [0 0 0 0 0 0];
```

%MOVIMIENTO TACO 01

```
for i=1:12,
%just update Y position in the T0 matrix corresponding to the piece
%robot1.piece.T0(2,4)=robot1.piece.T0(2,4)+i*0.0095;
```

```
robot1.equipment{48}.T0(2,4)=robot1.equipment{48}.T0(2,4)+i*0.0095;
```

```
%drawrobot3d(robot1, robot1.q, 1);
drawrobot3d(robot1, q1)
drawrobot3d(robot2, q2, 1)
%pause(0.002);
```

```
end
```

```
for i=1:17,
%just update X position in the T0 matrix corresponding to the piece
%robot1.piece.T0(1,4)=robot1.piece.T0(1,4)-i*0.0099;
robot1.equipment{48}.T0(1,4)=robot1.equipment{48}.T0(1,4)-
i*0.0099;
```

```
%drawrobot3d(robot1, robot1.q, 1)
drawrobot3d(robot1, q1)
drawrobot3d(robot2, q2, 1)
%pause(0.002);
```

```
end
```

```
end
```

```
function movimientos_taco_02
```

```
global TD_pallets robot1 robot2
%Proporcionar valores articulares
q1= [0 0 0 0 -30 80];
q2= [0 0 0 0 0 0];
```

%MOVIMIENTO TACO 02

```
for i=1:4,
%just update the Z position in the T0 matrix corresponding to the
piece
%robot1.piece.T0(3,4)=robot1.piece.T0(3,4)+i*0.0095;
robot1.equipment{49}.T0(3,4)=robot1.equipment{49}.T0(3,4)-
i*0.0094;
```

```
%drawrobot3d(robot1, robot1.q, 1);
drawrobot3d(robot1, q1)
drawrobot3d(robot2, q2, 1)
%pause(0.002);
```

```
End
```

```
for i=1:15,
%just update Y position in the T0 matrix corresponding to the piece
%robot1.piece.T0(2,4)=robot1.piece.T0(2,4)+i*0.0095;
```

```
robot1.equipment{49}.T0(2,4)=robot1.equipment{49}.T0(2,4)+i*0.0095;
```

```
%drawrobot3d(robot1, robot1.q, 1);
drawrobot3d(robot1, q1)
drawrobot3d(robot2, q2, 1)
%pause(0.002);
```

```
end
```

```

for i=1:17,
    %just update the X position in the T0 matrix
    corresponding to the piece
    %robot1.piece.T0(1,4)=robot1.piece.T0(1,4)-i*0.0099;

robot1.equipment{49}.T0(1,4)=robot1.equipment{49}.T0(1,
4)-i*0.0099;

    %drawrobot3d(robot1, robot1.q, 1)
    drawrobot3d(robot1, q1)
    drawrobot3d(robot2, q2, 1)

    %pause(0.002);
end
end

function movimientos_liston_01
global TD_pallets robot1 robot2

%Proporcionar valores articulares
q1= [0 0 0 0 -30 80];
q2= [0 0 0 0 0 0];

function pick01()
global RT_tp1 RT_tp2 RT_tp3 RT_tp4 RT_tp5 TD_tool1

TD_tool1=[1, [[0,0,0], [1,0,0,0]], [0.001, [0,0,0.001], [1,0,0,0], 0,0,0]];

RT_tp1=[[1.0706,0.0000,1.1301], [0.0056,0.7063,0.7079,-
0.0056], [0,0,0,0], [9E9,9E9,9E9,9E9,9E9,9E9]];
RT_tp2=[[0.0044,-1.0706,1.1301], [0.0080,1.0000,0.0032,-0.0000], [-
1,0,0,0], [9E9,9E9,9E9,9E9,9E9,9E9]];
RT_tp3=[[-0.0275,-1.4245,0.8377], [0.0125,0.9998,-0.0145,0.0001], [-
2,0,1,0], [9E9,9E9,9E9,9E9,9E9,9E9]];
RT_tp4=[[0.0056,-1.3786,0.7548], [0.0090,1.0000,0.0032,-0.0000], [-
1,0,0,0], [9E9,9E9,9E9,9E9,9E9,9E9]];
RT_tp5=[[0.0063,1.3786,0.7548], [0.0000,0.0012,1.0000,-
0.0090], [0,0,0,0], [9E9,9E9,9E9,9E9,9E9,9E9]];

MoveJ(RT_tp1, 'vmax' , 'fine' , TD_tool1, 'wobj0');
MoveJ(RT_tp2, 'vmax' , 'fine' , TD_tool1, 'wobj0');
MoveJ(RT_tp3, 'vmax' , 'fine' , TD_tool1, 'wobj0');

simulation_grip_piece;

MoveJ(RT_tp4, 'vmax' , 'fine' , TD_tool1, 'wobj0');
MoveJ(RT_tp5, 'vmax' , 'fine' , TD_tool1, 'wobj0');

end

```

%MOVIMIENTO LISTON 01

```

for i=1:12,
    %just update the Y position in the T0 matrix to the piece
    %robot1.piece.T0(2,4)=robot1.piece.T0(2,4)+i*0.0095;
    robot1.equipment{57}.T0(2,4)=robot1.equipment{57}.T0(2,4)-
i*0.021;

```

```

    %drawrobot3d(robot1, robot1.q, 1);
    drawrobot3d(robot1, q1)
    drawrobot3d(robot2, q2, 1)
    %pause(0.002);

```

end

```

for i=1:6,

```

```

    %just update X position in the T0 matrix to the piece
    %robot1.piece.T0(1,4)=robot1.piece.T0(1,4)-i*0.0099;

```

```

robot1.equipment{57}.T0(1,4)=robot1.equipment{57}.T0(1,4)+i*0.0
07;

```

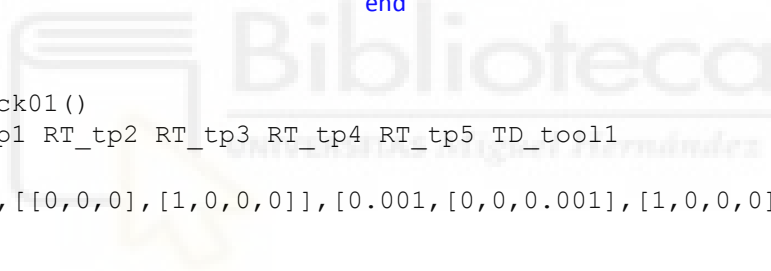
```

    %drawrobot3d(robot1, robot1.q, 1);
    drawrobot3d(robot1, q1)
    drawrobot3d(robot2, q2, 1)
    %pause(0.002);

```

end

end



PROG SIMULATION01 ROBOT RB01

```
1: LBL[200] ;
3: UFRAME_NUM=0 ;
4: UTOOL_NUM=2 ;
6: DO[1]=OFF ;
7: DO[2]=OFF ;
8: DO[3]=OFF ;
9: DO[4]=OFF ;
10: DO[5]=OFF ;
11: DO[6]=OFF ;
12: DO[7]=OFF ;
13: DO[11]=OFF ;
14: DO[12]=OFF ;
15: DO[13]=OFF ;
16: DO[14]=OFF ;
17: DO[15]=OFF ;
18: DO[16]=OFF ;
19: DO[17]=OFF ;
20: DO[18]=OFF ;
21: DO[19]=OFF ;
22: DO[20]=OFF ;
23: DO[21]=OFF ;
24: DO[22]=OFF ;
25: DO[23]=OFF ;
26: DO[24]=OFF ;
27: DO[25]=OFF ;
28: DO[31]=OFF ;
29: DO[32]=OFF ;
30: DO[33]=OFF ;
31: DO[34]=OFF ;
33: DO[8]=ON ;
34: DO[9]=ON ;
35: DO[10]=ON ;
36: DO[26]=ON ;
37:J P[5] 100% CNT100 ;
38: DO[1]=ON ;
39:J P[3] 100% CNT100 ;
40:J P[6] 100% CNT100 ;
41:J P[1] 100% FINE ;
42: UTOOL_NUM=3 ;
43: DO[1]=OFF ;
44:L P[2] 3000mm/sec CNT100 ;
45:J P[4] 100% CNT100 ;
46:J P[7] 100% CNT100 ;
47: DO[2]=ON ;
48:J P[8] 100% CNT100 ;
49:L P[9] 3000mm/sec FINE
50: DO[3]=ON ;
51: UFRAME_NUM=0 ;
52: UTOOL_NUM=2 ;
53:L P[10] 3000mm/sec CNT100 ;
54:J P[11] 100% CNT100 ;
55:J P[12] 100% CNT100 ;
56: DO[1]=ON ;
57:J P[14] 100% CNT100 ;
58:J P[15] 100% FINE ;
59: UTOOL_NUM=3 ;
60: DO[1]=OFF ;
61:L P[16] 3000mm/sec CNT100 ;
62:J P[17] 100% CNT100 ;
63: DO[4]=ON ;
64:J P[18] 100% CNT100 ;
65:L P[19] 3000mm/sec FINE
66: DO[5]=ON ;
67: UFRAME_NUM=0 ;
68: UTOOL_NUM=2 ;
69:L P[13] 3000mm/sec CNT100 ;
70:J P[20] 100% CNT100 ;
71:J P[21] 100% CNT100 ;
72: DO[1]=ON ;
73:J P[22] 100% CNT100 ;
74:J P[23] 100% FINE ;
75: UTOOL_NUM=3 ;
76: DO[1]=OFF ;
77:L P[24] 3000mm/sec CNT100 ;
78:J P[25] 100% CNT100 ;
79: DO[6]=ON ;
80:J P[26] 100% CNT100 ;
81:J P[27] 100% CNT100 ;
82:L P[28] 3000mm/sec FINE
83: DO[7]=ON ;
84: UFRAME_NUM=0 ;
85: UTOOL_NUM=2 ;
86:J P[29] 100% CNT100
```

87:J P[30] 100% CNT100 ;
 88:L P[31:NAIL01] 3000mm/sec FINE
 89:L P[32] 3000mm/sec CNT100 ;
 90:L P[33] 3000mm/sec CNT100
 91:L P[34:NAIL02] 100mm/sec FINE
 92:J P[35] 100% CNT100 ;
 93:J P[37] 100% CNT100 ;
 94:L P[38:NAIL03] 3000mm/sec FINE
 95:L P[39] 3000mm/sec CNT100 ;
 96:L P[40] 3000mm/sec CNT100 ;
 97:L P[41:NAIL04] 3000mm/sec FINE
 98:L P[42] 3000mm/sec CNT100 ;
 99:L P[36] 3000mm/sec CNT100 ;
 100:L P[43:NAIL05] 3000mm/sec FINE
 101:L P[44] 3000mm/sec CNT100 ;
 102:L P[45] 3000mm/sec CNT100 ;
 103:L P[46:NAIL06] 3000mm/sec FINE
 104:L P[47] 3000mm/sec CNT100 ;
 105:L P[48] 3000mm/sec CNT100 ;
 106:L P[49:NAIL07] 3000mm/sec FINE
 107:L P[50] 3000mm/sec CNT100 ;
 108:L P[51] 3000mm/sec CNT100 ;
 109:L P[52:NAIL08] 3000mm/sec FINE
 110:L P[53] 3000mm/sec CNT100 ;
 111:L P[54] 3000mm/sec CNT100 ;
 112:L P[55:NAIL09] 3000mm/sec FINE
 113:L P[56] 3000mm/sec CNT100 ;
 114:L P[57] 3000mm/sec CNT100 ;
 115:L P[58:NAIL10] 3000mm/sec FINE
 116:L P[59] 3000mm/sec CNT100 ;
 117:L P[60] 3000mm/sec CNT100 ;
 118:L P[61:NAIL11] 3000mm/sec FINE
 119:L P[62] 3000mm/sec CNT100 ;
 120:L P[63] 3000mm/sec CNT100 ;
 121:L P[64:NAIL12] 3000mm/sec FINE
 122:L P[65] 3000mm/sec CNT100 ;
 123:L P[66] 3000mm/sec CNT100 ;
 124:L P[67:NAIL13] 3000mm/sec FINE
 125:L P[68] 3000mm/sec CNT100 ;
 126:L P[69] 3000mm/sec CNT100 ;
 127:L P[70:NAIL14] 3000mm/sec FINE
 128:L P[71] 3000mm/sec CNT100
 129:L P[72] 3000mm/sec CNT100
 130:L P[73:NAIL15] 3000mm/sec FINE
 131:L P[74] 3000mm/sec CNT100 ;
 132:L P[75] 3000mm/sec CNT100 ;
 133:L P[76:NAIL16] 3000mm/sec FINE
 134:L P[77] 3000mm/sec CNT100 ;
 135:L P[78] 3000mm/sec CNT100 ;
 136:L P[79:NAIL17] 3000mm/sec FINE
 137:L P[80] 3000mm/sec CNT100 ;
 138:L P[81] 3000mm/sec CNT100 ;
 139:L P[82:NAIL18] 3000mm/sec FINE
 140:L P[83] 100mm/sec CNT100 ;
 141:J P[85:WAIT.POS] 100% FINE ;
 142: ;
 143: DO[6]=OFF ;
 144: DO[7]=OFF ;
 145: DO[8]=OFF ;
 146: DO[11]=ON ;
 147: DO[20]=ON ;
 148: WAIT .25(sec) ;
 149: DO[4]=OFF ;
 150: DO[5]=OFF ;
 151: DO[9]=OFF ;
 152: DO[12]=ON ;
 153: DO[21]=ON ;
 154: WAIT .25(sec) ;
 155: DO[2]=OFF ;
 156: DO[3]=OFF ;
 157: DO[10]=OFF ;
 158: DO[13]=ON ;
 159: DO[22]=ON ;
 160: WAIT .25(sec) ;
 161: ;
 162: ;
 163: DO[20]=OFF ;
 164: DO[14]=ON ;
 165: WAIT .25(sec) ;
 166: DO[21]=OFF ;

167: DO[15]=ON ;
168: WAIT .25(sec) ;
169: DO[22]=OFF ;
170: DO[16]=ON ;
171: WAIT .25(sec) ;
172: ;
173: DO[11]=OFF ;
174: DO[14]=OFF ;
175: DO[8]=ON ;
176: DO[17]=ON ;
177: WAIT .25(sec) ;
178: DO[12]=OFF ;
179: DO[15]=OFF ;
180: DO[9]=ON ;
181: DO[18]=ON ;
182: WAIT .25(sec) ;
183: DO[13]=OFF ;
184: DO[16]=OFF ;
185: DO[10]=ON ;
186: DO[19]=ON ;
187: WAIT .25(sec) ;
188: ;
189: J P[84] 100% CNT100 ;
190: L P[86] 3000mm/sec FINE ;
191: UFRAME_NUM=0 ;
192: UTOOL_NUM=4 ;
193: DO[19]=OFF ;
194: J P[87] 100% CNT100 ;
195: J P[88] 100% CNT100 ;
196: J P[89] 100% CNT100 ;
197: L P[90] 3000mm/sec FINE ;
198: DO[23]=ON ;
199: UFRAME_NUM=0 ;
200: UTOOL_NUM=2 ;
201: L P[91] 3000mm/sec CNT100 ;
202: J P[92] 100% CNT100 ;
203: L P[93] 3000mm/sec FINE ;
204: UFRAME_NUM=0 ;
205: UTOOL_NUM=4 ;
206: DO[18]=OFF ;
207: L P[94] 3000mm/sec CNT100 ;
208: J P[95] 100% CNT100 ;
209: J P[97] 100% CNT100 ;
210: J P[96] 100% FINE ;
211: DO[24]=ON ;
212: UFRAME_NUM=0 ;
213: UTOOL_NUM=2 ;
214: J P[98] 100% CNT100 ;
215: J P[100] 100% CNT100 ;
216: L P[99] 3000mm/sec FINE ;
217: UFRAME_NUM=0 ;
218: UTOOL_NUM=4 ;
219: DO[17]=OFF ;
220: J P[101] 100% CNT100 ;
221: J P[102] 100% CNT100 ;
222: J P[103] 100% CNT100 ;
223: L P[104] 3000mm/sec FINE ;
224: DO[25]=ON ;
225: UFRAME_NUM=0 ;
226: UTOOL_NUM=2 ;
227: J P[105] 100% CNT100 ;
228: ;
229: DO[1]=ON ;
230: J P[107] 100% CNT100 ;
231: J P[108] 100% CNT100 ;
232: J P[109] 100% FINE ;
233: UTOOL_NUM=3 ;
234: DO[1]=OFF ;
235: L P[110] 3000mm/sec CNT100 ;
236: J P[111] 100% CNT100 ;
237: J P[112] 100% CNT100 ;
238: L P[113] 3000mm/sec FINE ;
239: DO[31]=ON ;
240: UFRAME_NUM=0 ;
241: UTOOL_NUM=2 ;
242: L P[114] 3000mm/sec CNT100 ;
243: ;
244: DO[1]=ON ;
245: J P[116] 100% CNT100 ;
246: J P[117] 100% FINE ;
247: UTOOL_NUM=3 ;
248: DO[1]=OFF ;
249: L P[118] 3000mm/sec CNT100 ;
250: J P[120] 100% CNT100 ;

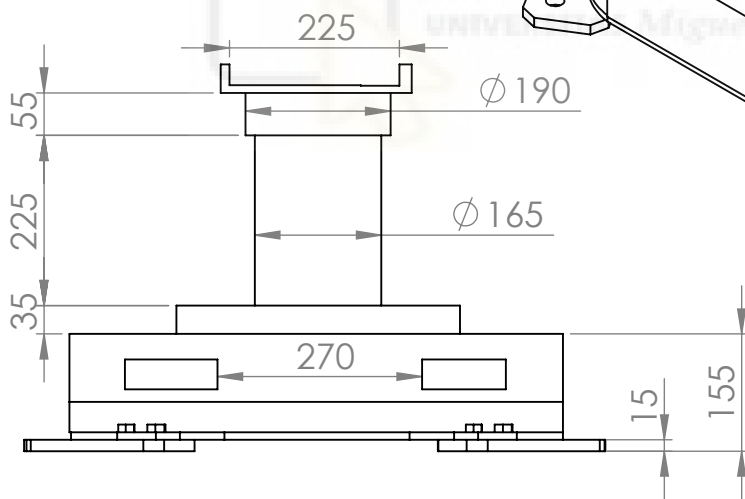
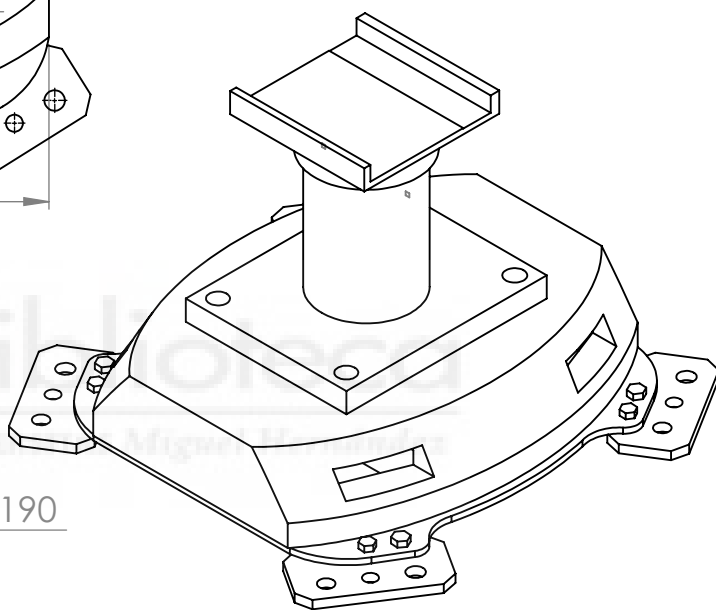
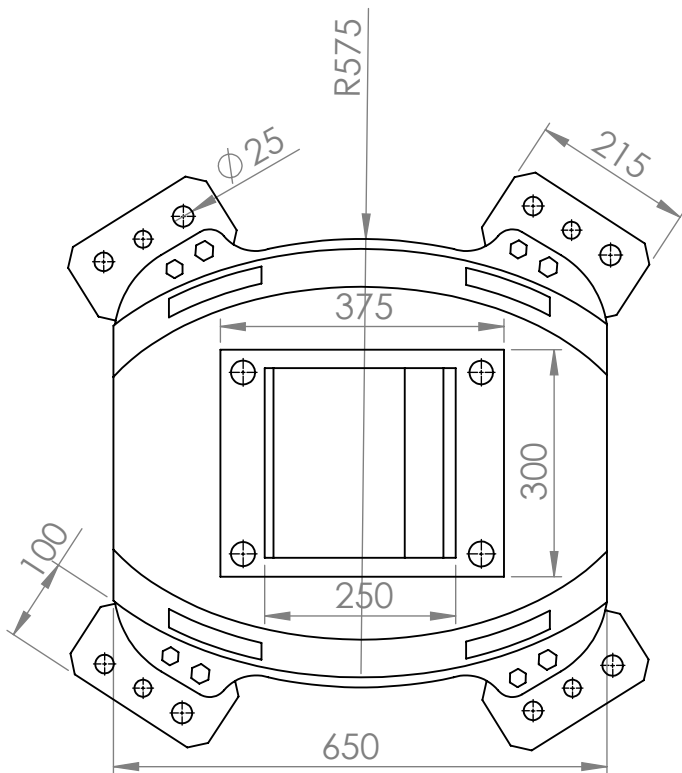
251:J P[119] 100% CNT100 ;
252:L P[115] 3000mm/sec FINE ;
253: DO[32]=ON ;
254: UFRAME_NUM=0 ;
255: UTOOL_NUM=2 ;
256:J P[121] 100% CNT100 ;
257: DO[1]=ON ;
258:J P[122] 100% CNT100 ;
259:J P[123] 100% FINE ;
260: UTOOL_NUM=3 ;
261: DO[1]=OFF ;
262:L P[124] 3000mm/sec CNT100 ;
263:J P[125] 100% CNT100 ;
264:J P[126] 100% CNT100 ;
265:L P[127] 3000mm/sec FINE ;
266: DO[33]=ON ;
267: UFRAME_NUM=0 ;
268: UTOOL_NUM=2 ;
269:J P[128] 100% CNT100 ;
270:J P[129] 100% FINE ;
271: ;
272: DO[26]=OFF ;
273: DO[30]=OFF ;
274: DO[31]=OFF ;
275: DO[32]=OFF ;
276: DO[33]=OFF ;
277: DO[27]=ON ;
278: WAIT .25(sec) ;
279: DO[27]=OFF ;
280: DO[28]=ON ;
281: WAIT .25(sec) ;
282: DO[28]=OFF ;
283: DO[29]=ON ;
284: WAIT .25(sec) ;
285: DO[29]=OFF ;
286: DO[30]=ON ;
287: DO[34]=ON ;
288: WAIT .25(sec) ;
289: DO[34]=OFF ;
290: ;
291: JMP LBL[200] ;

PROG SIMULATION01ROBOT RB02

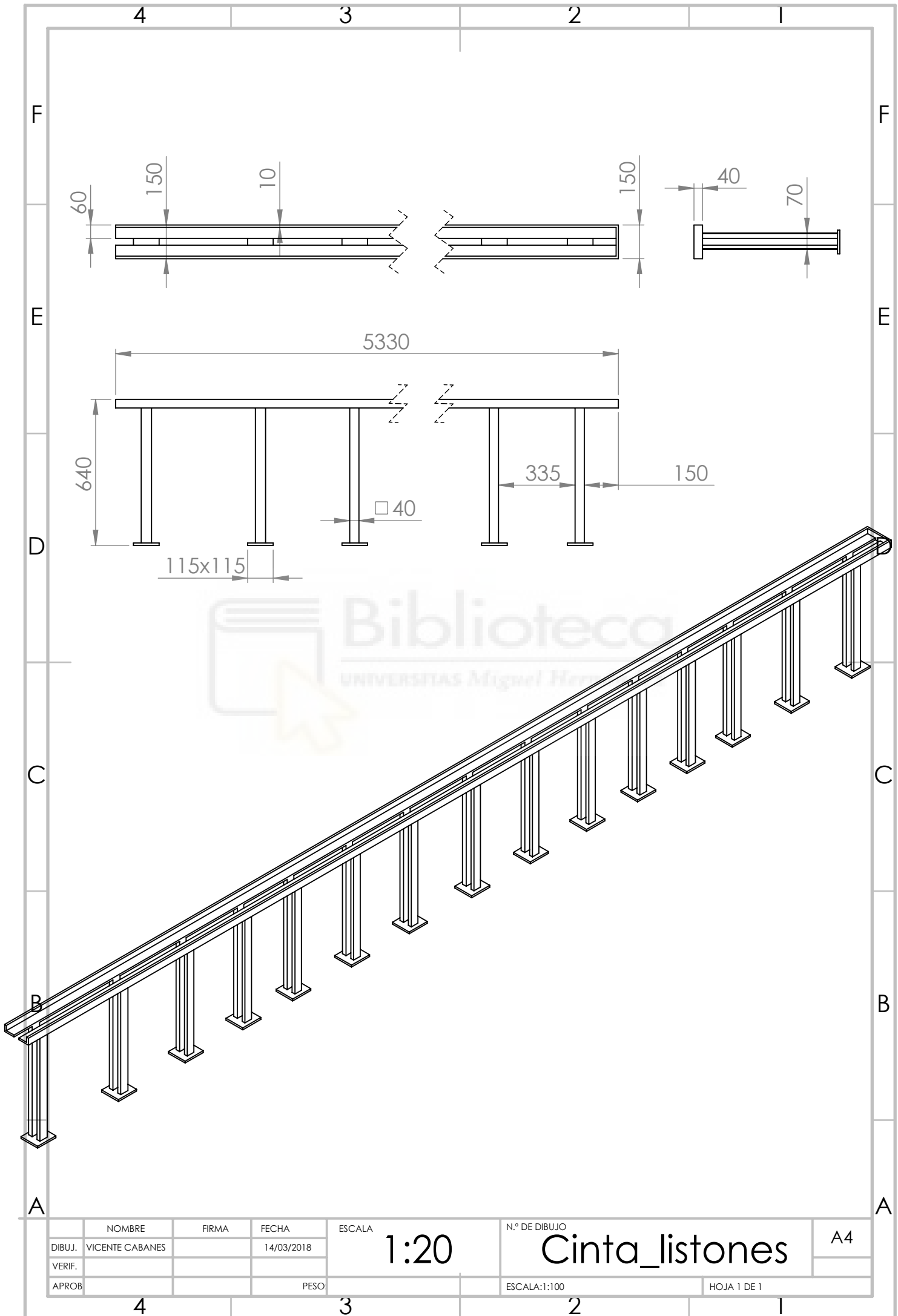
```
1: DO[1:ENTRADA.LISTON]=OFF ;
2: DO[2:LIST.FIXT01]=OFF ;
3: DO[3:LIST.FIXT02]=OFF ;
4: DO[4:LIST.FIXT03]=OFF ;
5: DO[5:LIST.FIXT04]=OFF ;
6: DO[6:LIST.FIXT05]=OFF ;
7: DO[7:LIST.FIXT06]=OFF ;
8: DO[8:LIST.FIXT07]=OFF ;
9: DO[9:LIST.FIXT08]=OFF ;
10: DO[10:LIST.FIXT09]=OFF ;
11: DO[11:LIST.FIXT10]=OFF ;
12: DO[12:CLEAR]=OFF ;
13: R[1]=0 ;
15: UFRAME_NUM=0 ;
16: UTOOL_NUM=2 ;
17: J P[147] 100% FINE ;
18: ;
19: WAIT 2.25(sec) ;
21: LBL[200] ;
22: ;
23: J P[1] 100% FINE ;
24: J P[85] 100% CNT100 ;
25: DO[1:ENTRADA.LISTON]=ON ;
26: J P[86] 100% CNT100 ;
27: L P[87] 3000mm/sec FINE ;
28: UFRAME_NUM=0 ;
29: UTOOL_NUM=3 ;
30: DO[1:ENTRADA.LISTON]=OFF ;
31: L P[88] 3000mm/sec CNT100 ;
32: J P[89] 100% CNT100 ;
33: J P[90] 100% FINE ;
34: ;
35: IF R[1]=0,JMP LBL[300] ;
36: WAIT 5.10(sec) ;
37: LBL[300] ;
39: ;
40: IF R[1]>0,JMP LBL[100] ;
41: WAIT 50.00(sec) ;
42: LBL[100] ;
44: DO[12:CLEAR]=ON ;
45: L P[91:L1] 3000mm/sec FINE ;
46: DO[5:LIST.FIXT04]=ON ;
47: UFRAME_NUM=0 ;
48: UTOOL_NUM=2 ;
49: J P[92] 100% CNT100 ;
50: J P[93] 100% CNT100 ;
51: DO[1:ENTRADA.LISTON]=ON ;
52: J P[94] 100% CNT100 ;
53: L P[95] 3000mm/sec FINE ;
54: UFRAME_NUM=0 ;
55: UTOOL_NUM=3 ;
56: DO[1:ENTRADA.LISTON]=OFF ;
57: L P[96] 3000mm/sec CNT100 ;
58: J P[97] 100% CNT100 ;
59: J P[98] 100% CNT100 ;
60: L P[99:L2] 3000mm/sec FINE ;
61: DO[6:LIST.FIXT05]=ON ;
62: UFRAME_NUM=0 ;
63: UTOOL_NUM=2 ;
64: J P[100] 100% CNT100 ;
65: J P[101] 100% CNT100 ;
66: DO[1:ENTRADA.LISTON]=ON ;
67: J P[102] 100% CNT100 ;
68: L P[103] 3000mm/sec FINE ;
69: UFRAME_NUM=0 ;
70: UTOOL_NUM=3 ;
71: DO[1:ENTRADA.LISTON]=OFF ;
72: L P[104] 3000mm/sec CNT100 ;
73: J P[105] 100% CNT100 ;
74: J P[106] 100% CNT100 ;
75: L P[107:L3] 3000mm/sec FINE ;
76: DO[7:LIST.FIXT06]=ON ;
77: UFRAME_NUM=0 ;
78: UTOOL_NUM=2 ;
79: J P[108] 100% CNT100 ;
80: J P[109] 100% CNT100 ;
81: DO[1:ENTRADA.LISTON]=ON ;
82: J P[110] 100% CNT100 ;
83: L P[111] 3000mm/sec FINE ;
84: UFRAME_NUM=0 ;
85: UTOOL_NUM=3 ;
86: DO[1:ENTRADA.LISTON]=OFF ;
87: L P[112] 3000mm/sec CNT100 ;
88: J P[113] 100% CNT100 ;
```

89:J P[114] 100% CNT100 ;
 90:J P[115] 100% FINE ;
 91: DO[8:LIST.FIXT07]=ON ;
 92: UFRAME_NUM=0 ;
 93: UTOOL_NUM=2 ;
 94:J P[116] 100% CNT100 ;
 95:J P[117] 100% CNT100 ;
 96: DO[1:ENTRADA.LISTON]=ON ;
 97:J P[118] 100% CNT100 ;
 98:L P[119] 3000mm/sec FINE ;
 99: UFRAME_NUM=0 ;
 100: UTOOL_NUM=3 ;
 101: DO[1:ENTRADA.LISTON]=OFF ;
 102:L P[120] 3000mm/sec CNT100 ;
 103:J P[121] 100% CNT100 ;
 104:J P[122] 100% CNT100 ;
 105:J P[123] 100% FINE ;
 106: DO[9:LIST.FIXT08]=ON ;
 107: UFRAME_NUM=0 ;
 108: UTOOL_NUM=2 ;
 109:J P[124] 100% CNT100 ;
 110:J P[125] 100% CNT100 ;
 111: DO[1:ENTRADA.LISTON]=ON ;
 112:J P[126] 100% CNT100 ;
 113:L P[127] 3000mm/sec FINE ;
 114: UFRAME_NUM=0 ;
 115: UTOOL_NUM=3 ;
 116: DO[1:ENTRADA.LISTON]=OFF ;
 117:L P[128] 3000mm/sec CNT100 ;
 118:J P[129] 100% CNT100 ;
 119:J P[130] 100% CNT100 ;
 120:J P[131] 100% FINE ;
 121: DO[10:LIST.FIXT09]=ON ;
 122: UFRAME_NUM=0 ;
 123: UTOOL_NUM=2 ;
 124:J P[132] 100% CNT100 ;
 125:J P[133] 100% CNT100 ;
 126: DO[1:ENTRADA.LISTON]=ON ;
 127:J P[134] 100% CNT100 ;
 128:L P[135] 3000mm/sec FINE ;
 129: UFRAME_NUM=0 ;
 130: UTOOL_NUM=3 ;
 131: DO[1:ENTRADA.LISTON]=OFF ;
 132:L P[136] 3000mm/sec CNT100 ;
 133:J P[137] 100% CNT100 ;
 134:J P[138] 100% CNT100 ;
 135:L P[139] 3000mm/sec FINE ;
 136: DO[11:LIST.FIXT10]=ON ;
 137: UFRAME_NUM=0 ;
 138: UTOOL_NUM=2 ;
 139:J P[140] 100% CNT100 ;
 140:J P[27] 100% CNT100 ;
 141:J P[28] 100% CNT100 ;
 142:L P[29:NAIL01] 3000mm/sec FINE
 143:L P[30] 3000mm/sec CNT100 ;
 144:L P[31] 3000mm/sec CNT100 ;
 145:L P[32:NAIL02] 3000mm/sec FINE
 146:L P[33] 3000mm/sec CNT100 ;
 147:L P[34] 3000mm/sec CNT100 ;
 148:L P[35:NAIL03] 3000mm/sec FINE
 149:L P[36] 3000mm/sec CNT100 ;
 150:L P[2] 3000mm/sec CNT100 ;
 151:L P[37:NAIL04] 3000mm/sec FINE
 152:L P[38] 3000mm/sec FINE ;
 153:L P[39] 3000mm/sec CNT100 ;
 154:L P[40:NAIL05] 3000mm/sec FINE
 155:L P[41] 3000mm/sec CNT100 ;
 156:L P[42] 3000mm/sec CNT100 ;
 157:L P[43:NAIL06] 3000mm/sec FINE
 158:L P[47] 3000mm/sec CNT100 ;
 159:L P[48] 3000mm/sec CNT100 ;
 160:L P[49:NAIL07] 3000mm/sec FINE
 161:L P[50] 3000mm/sec CNT100 ;
 162:L P[51] 3000mm/sec CNT100 ;
 163:L P[52:NAIL08] 3000mm/sec FINE
 164:L P[53] 3000mm/sec CNT100 ;
 165:J P[54] 100% CNT100 ;
 166:L P[55:NAIL09] 3000mm/sec FINE
 167:L P[56] 3000mm/sec CNT100 ;
 168:L P[57] 3000mm/sec CNT100 ;
 169:L P[58:NAIL10] 3000mm/sec FINE
 170:L P[59] 3000mm/sec CNT100 ;
 171:L P[60] 3000mm/sec CNT100 ;
 172:L P[61:NAIL11] 3000mm/sec FINE
 173:L P[62] 3000mm/sec CNT100 ;
 174:L P[63] 3000mm/sec CNT100 ;

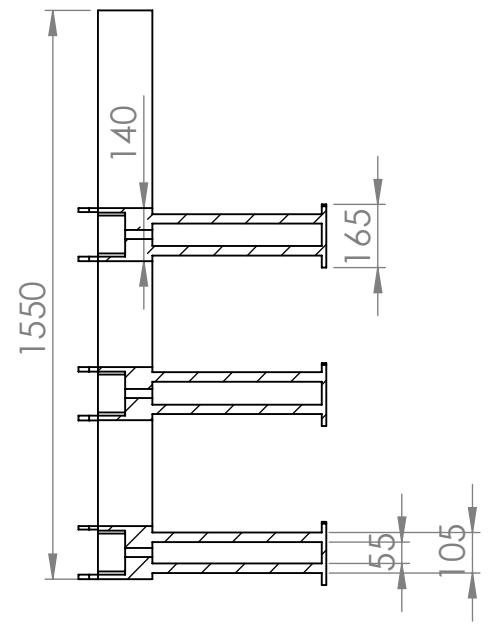
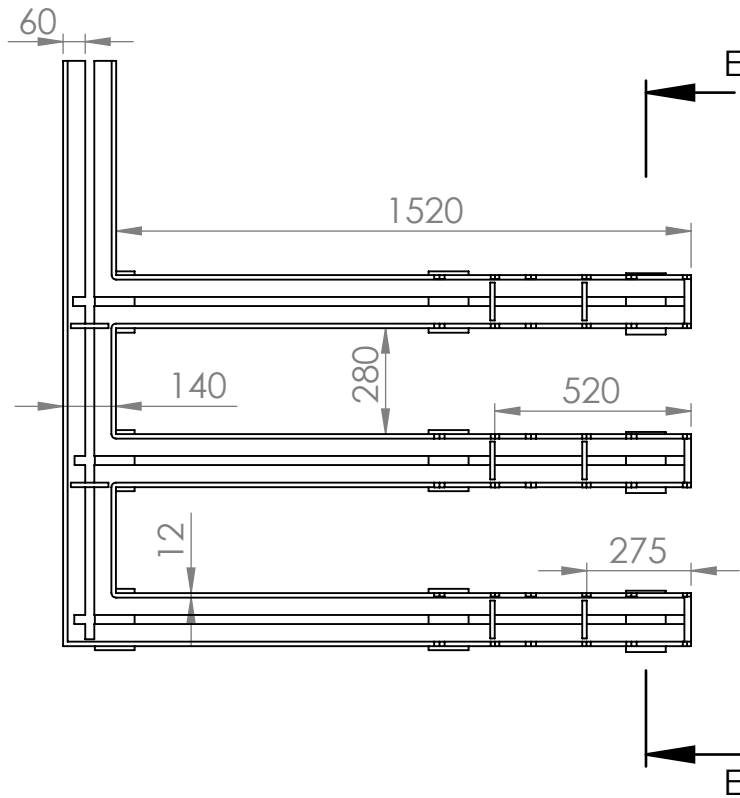
```
175:L P[64:NAIL12] 3000mm/sec FINE
;
176:L P[65] 3000mm/sec CNT100 ;
177: ;
178:J P[84] 100% CNT100 ;
179:L P[44] 3000mm/sec FINE ;
180: UFRAME_NUM=0 ;
181: UTOOL_NUM=4 ;
182: DO[2:LIST.FIXT01]=OFF ;
183: DO[3:LIST.FIXT02]=OFF ;
184: DO[4:LIST.FIXT03]=OFF ;
185: DO[5:LIST.FIXT04]=OFF ;
186: DO[6:LIST.FIXT05]=OFF ;
187: DO[7:LIST.FIXT06]=OFF ;
188: DO[8:LIST.FIXT07]=OFF ;
189: DO[9:LIST.FIXT08]=OFF ;
190: DO[10:LIST.FIXT09]=OFF ;
191: DO[11:LIST.FIXT10]=OFF ;
192: DO[12:CLEAR]=OFF ;
193:J P[141] 100% CNT100 ;
194:J P[142] 100% CNT100 ;
195:J P[143] 100% CNT100 ;
196:L P[144:DROP] 3000mm/sec FINE
;
197: DO[13:FINAL]=ON ;
198: UFRAME_NUM=0 ;
199: UTOOL_NUM=2 ;
200:L P[145] 3000mm/sec CNT100 ;
201:J P[146] 100% FINE ;
202: ;
203: ;
204: R[1]=R[1]+1 ;
205: JMP LBL[200] ;
```

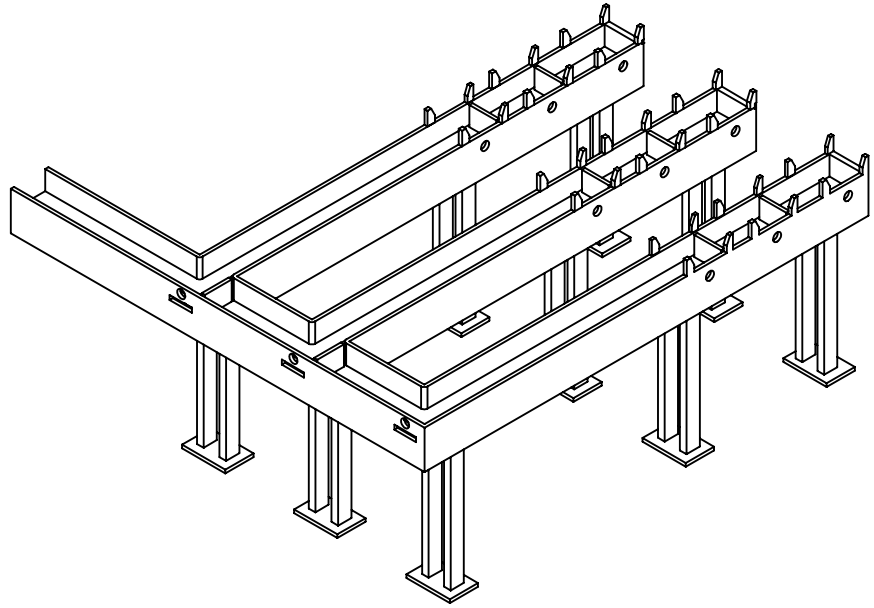
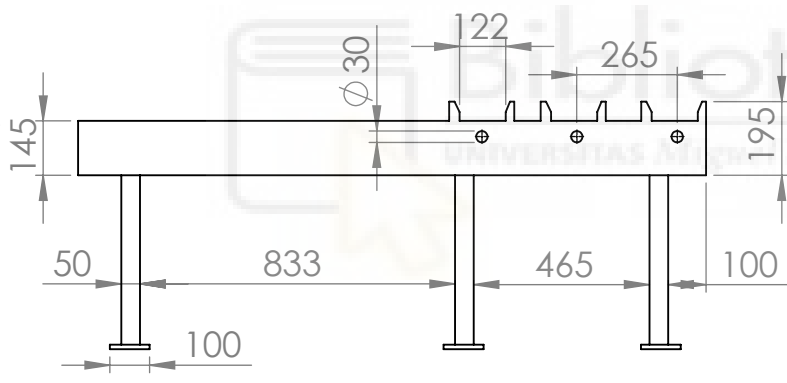
NOMBRE		FIRMA		FECHA		ESCALA		N.º DE DIBUJO	
DIBUJ. VICENTE CABANES				14/03/2018		1:10		Base_Fixture	
VERIF.								A4	
APROB.				PESO		ESCALA:1:20		HOJA 1 DE 1	



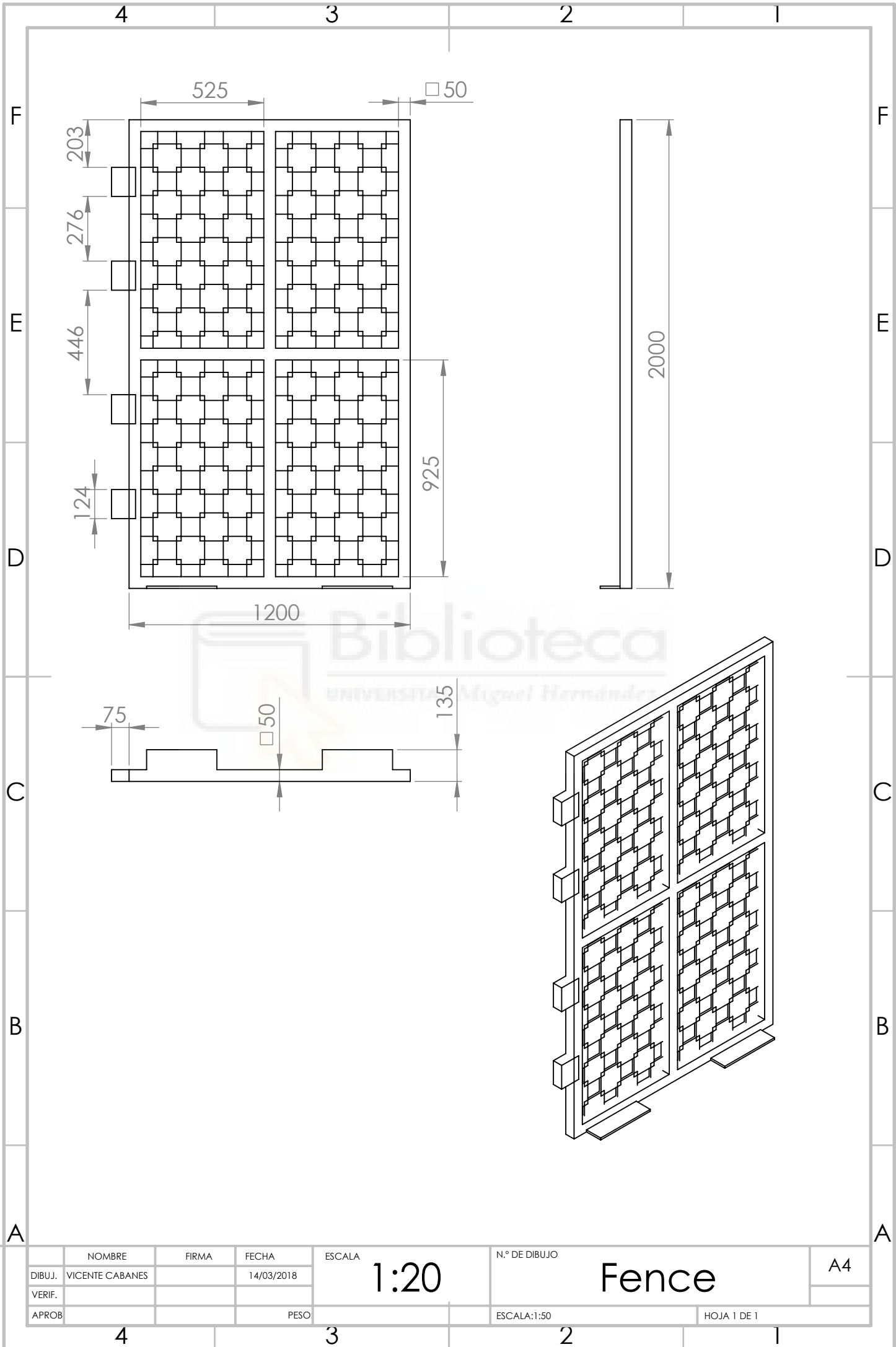
	NOMBRE	FIRMA	FECHA	ESCALA	N.º DE DIBUJO	
DIBUJ.	VICENTE CABANES		14/03/2018	1:20	Cinta_listones	A4
VERIF.						
APROB			PESO	ESCALA:1:100	HOJA 1 DE 1	



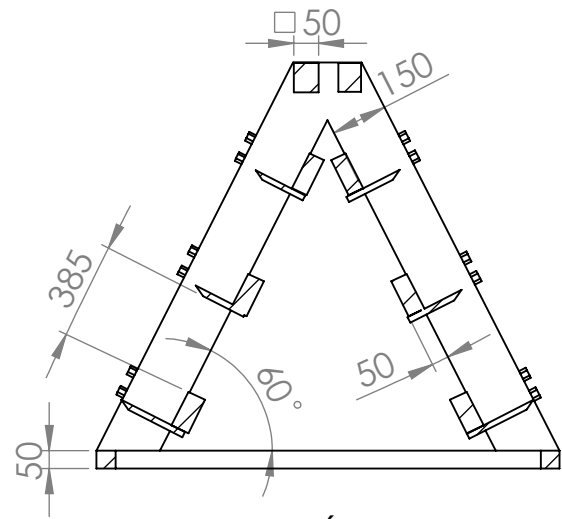
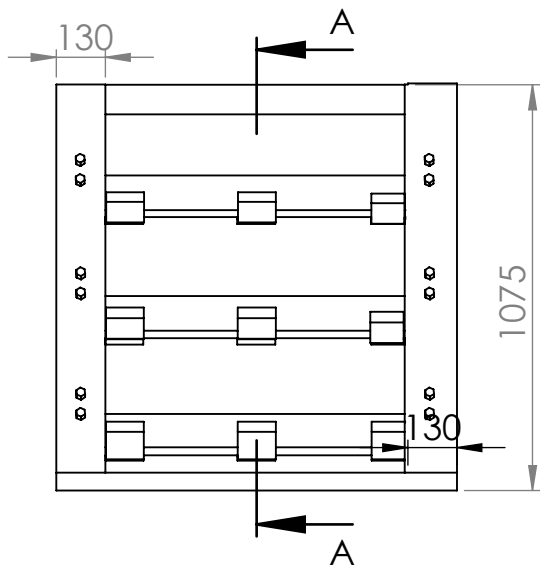
SECCIÓN E-E
ESCALA 1 : 20



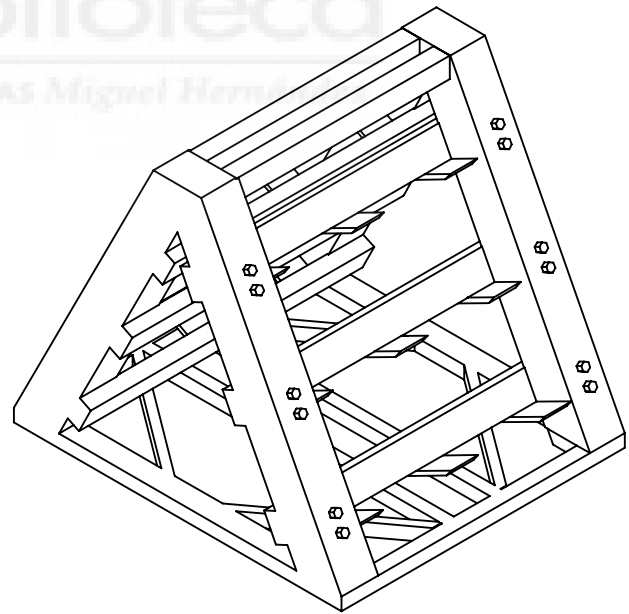
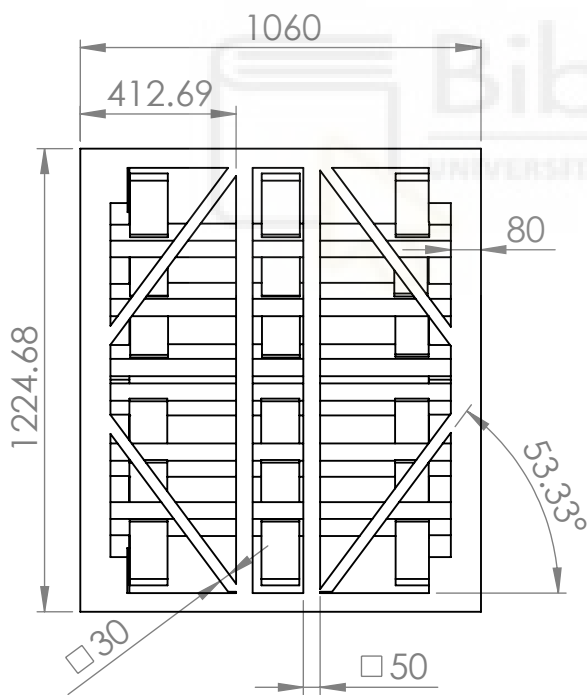
	NOMBRE	FIRMA	FECHA	ESCALA	N.º DE DIBUJO	
DIBUJ.	VICENTE CABANES		14/03/2018	1:20	Cinta_tacos	A4
VERIF.						
APROB			PESO	ESCALA:1:50	HOJA 1 DE 1	



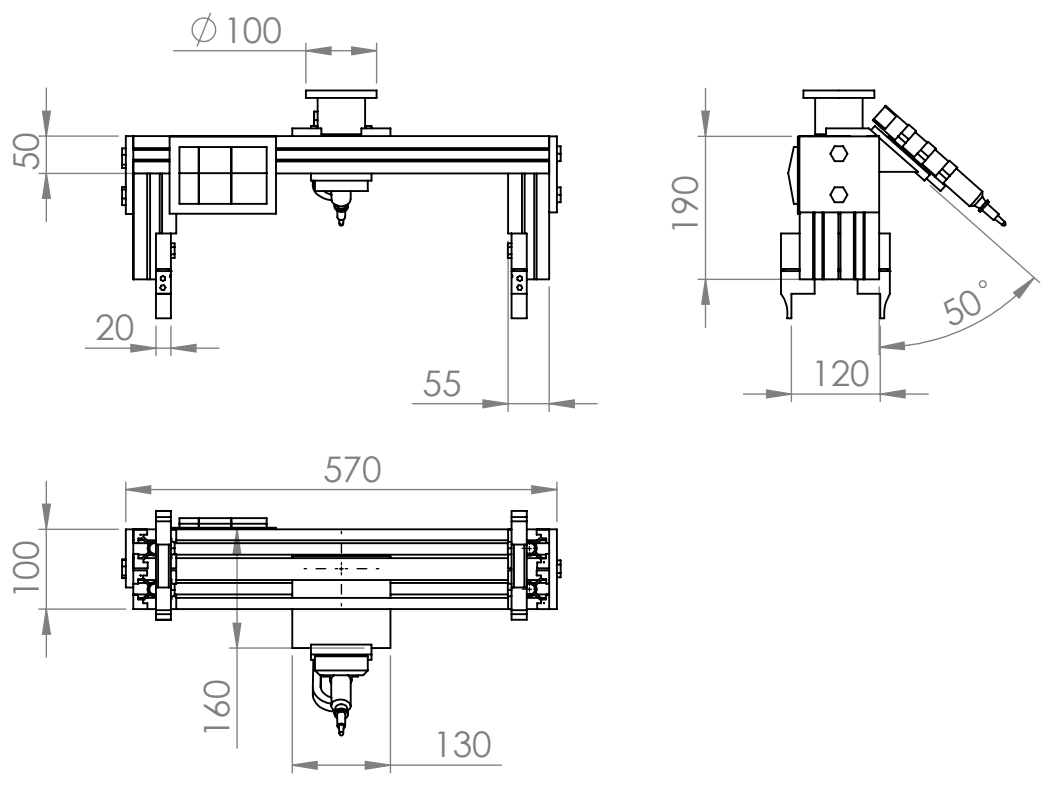
	NOMBRE	FIRMA	FECHA	ESCALA	N.º DE DIBUJO	
DIBUJ.	VICENTE CABANES		14/03/2018	1:20	Fence	A4
VERIF.						
APROB.			PESO	ESCALA:1:50	HOJA 1 DE 1	



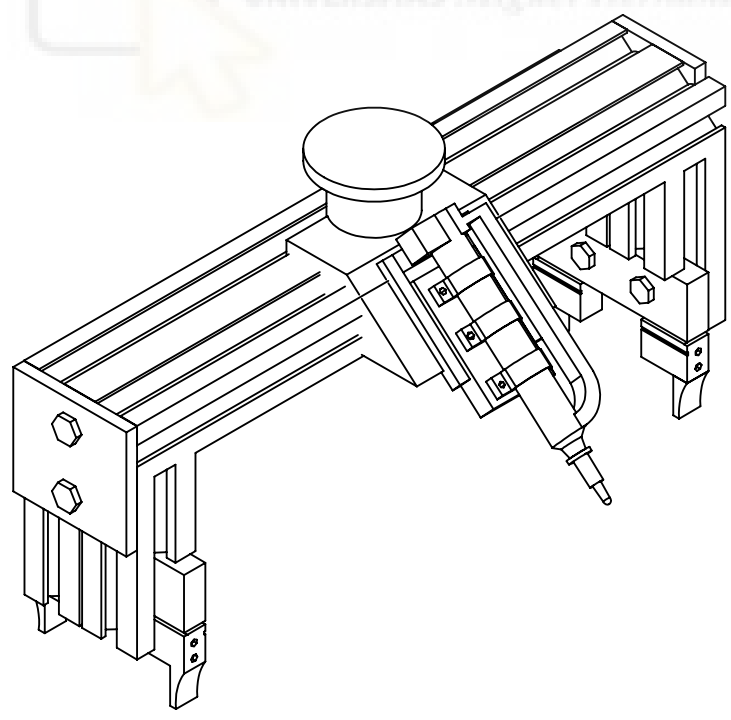
SECCIÓN A-A



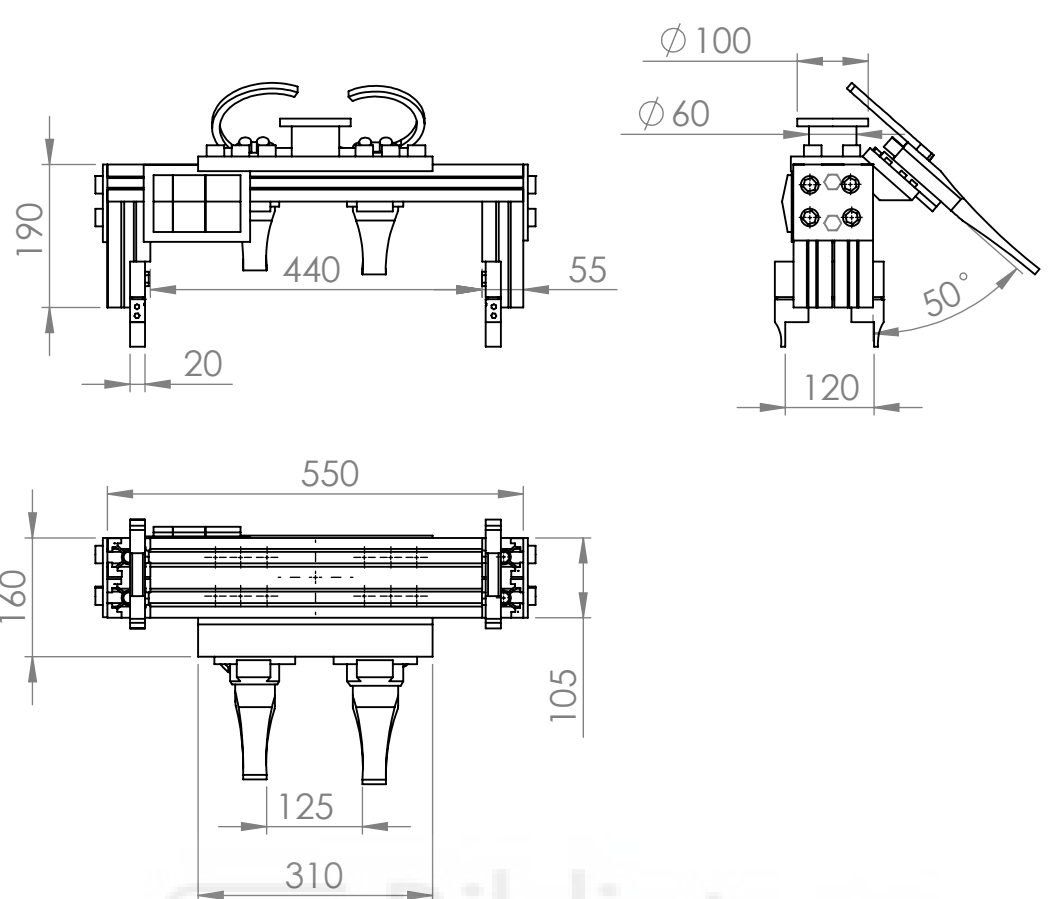
NOMBRE		FIRMA	FECHA	ESCALA	N.º DE DIBUJO	A4	
DIBUJ.	VICENTE CABANES		14/03/2018	1:20	Fixture		
VERIF.							
APROB.			PESO	ESCALA:1:20	HOJA 1 DE 1		



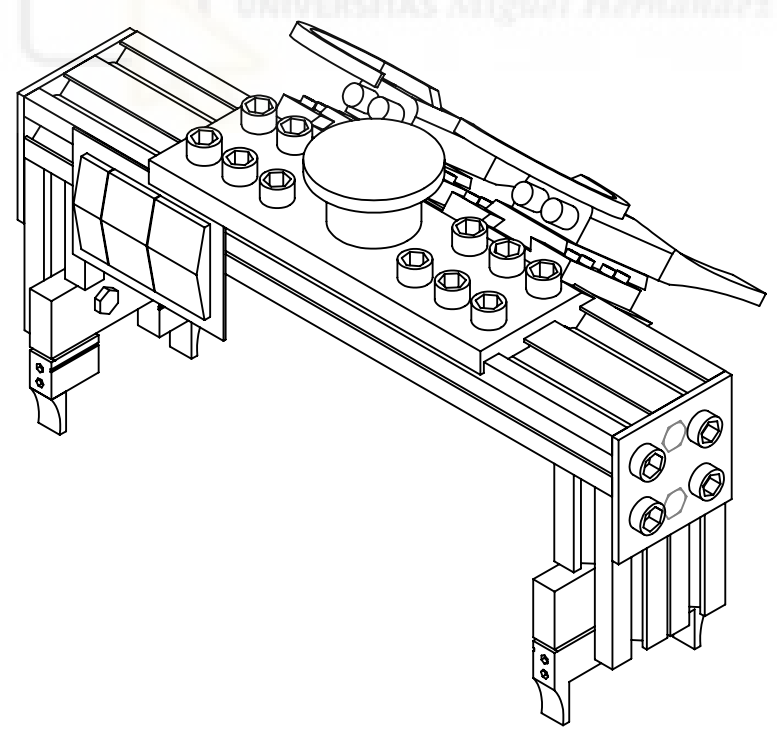
Biblioteca
UNIVERSITAS Miguel Hernández



	NOMBRE	FIRMA	FECHA	ESCALA	N.º DE DIBUJO	
DIBUJ.	VICENTE CABANES		14/03/2018	1:10	Herramienta01	A4
VERIF.						
APROB.			PESO	ESCALA:1:10	HOJA 1 DE 1	



Biblioteca
UNIVERSITAT Miguel Hernández



	NOMBRE	FIRMA	FECHA	ESCALA	N.º DE DIBUJO	
DIBUJ.	VICENTE CABANES		14/03/2018	1:10	Herramienta02	A4
VERIF.						
APROB			PESO	ESCALA:1:10	HOJA 1 DE 1	

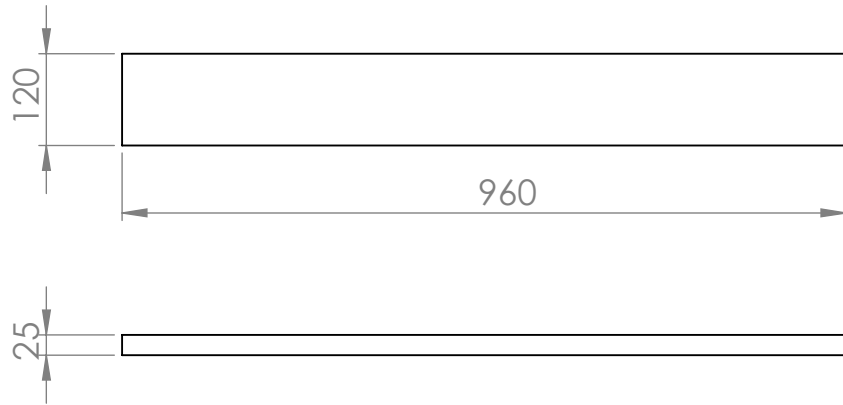
4 3 2 1

F

F

E

E



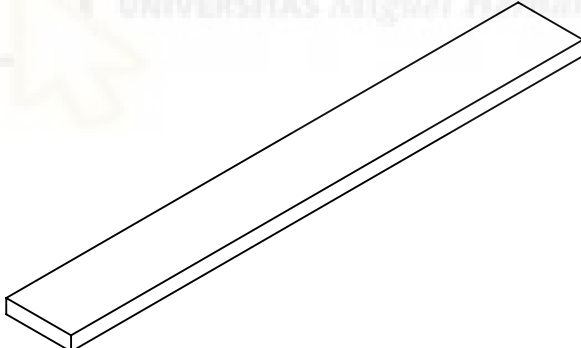
D

D



C

C



B

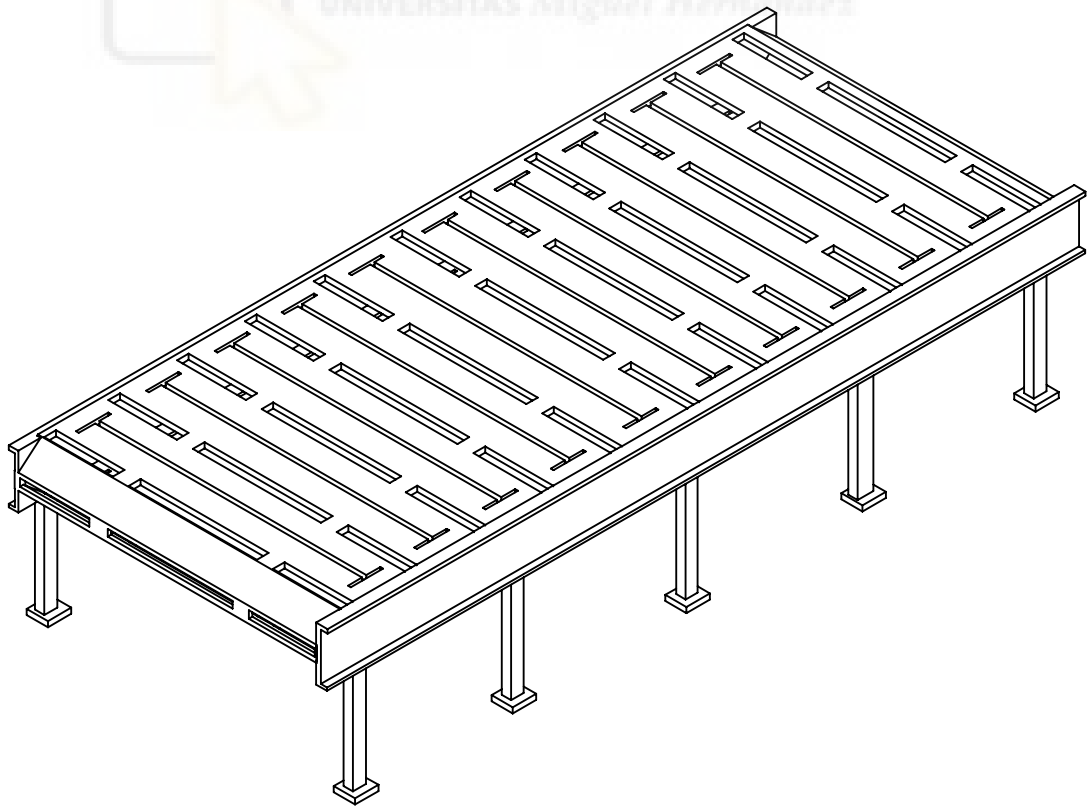
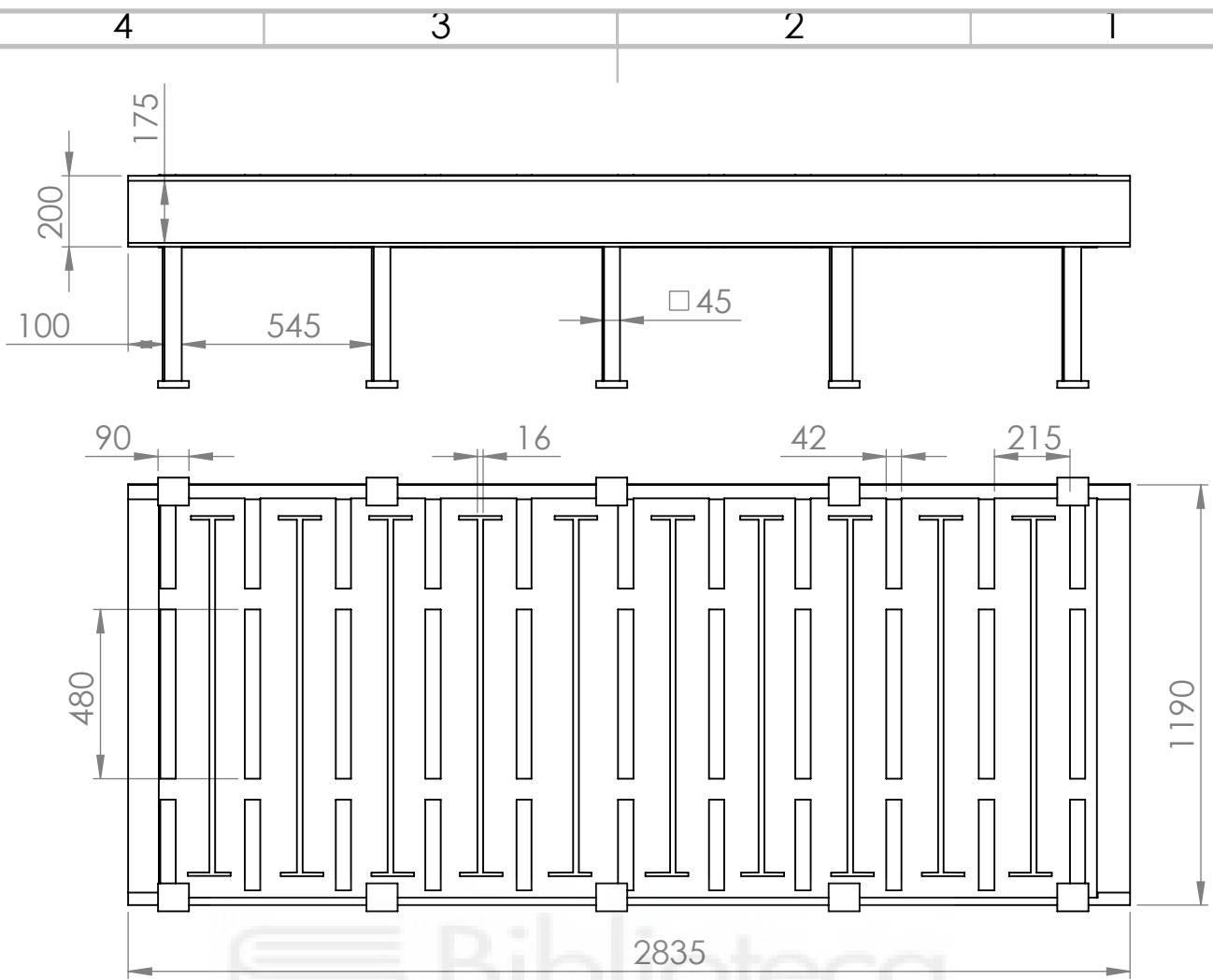
B

A

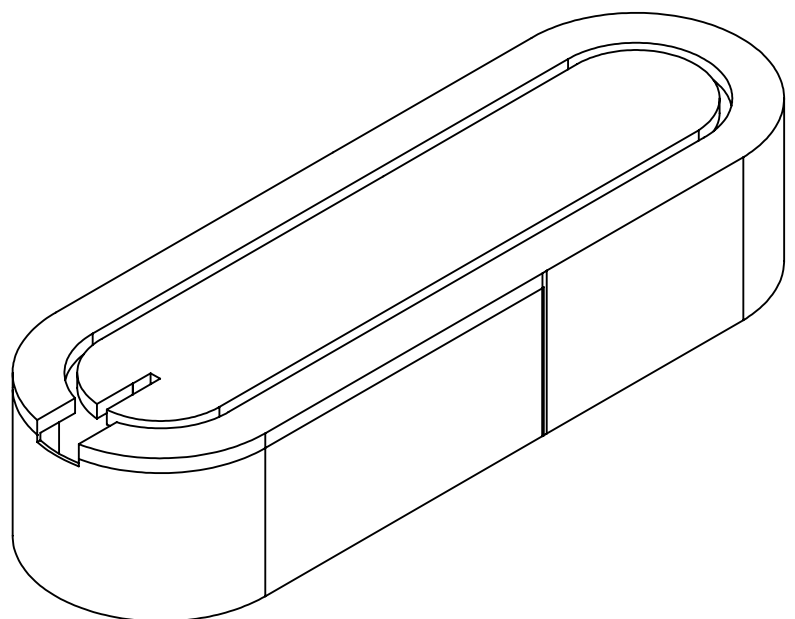
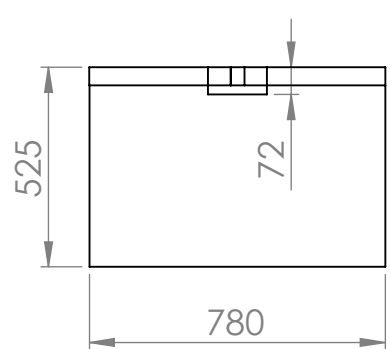
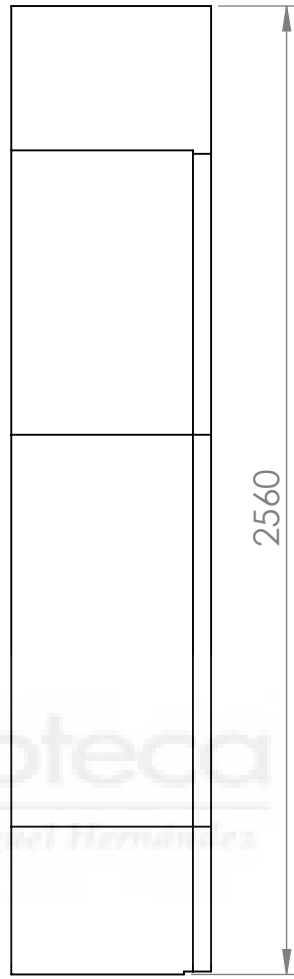
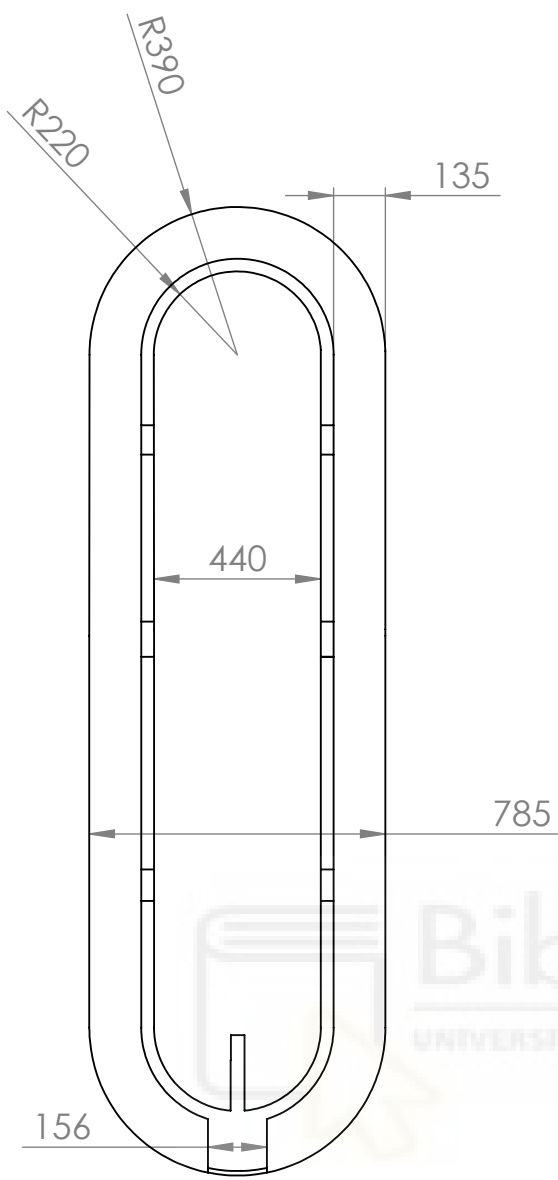
A

	NOMBRE	FIRMA	FECHA	ESCALA	N.º DE DIBUJO	
DIBUJ.	VICENTE CABANES		14/03/2018	1:10	Listón	A4
VERIF.						
APROB.			PESO	ESCALA:1:10	HOJA 1 DE 1	

4 3 2 1

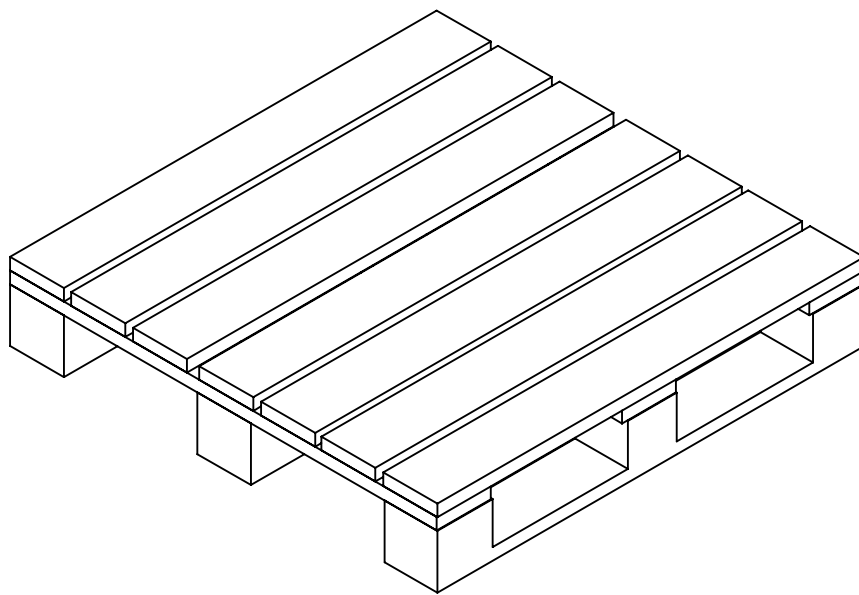
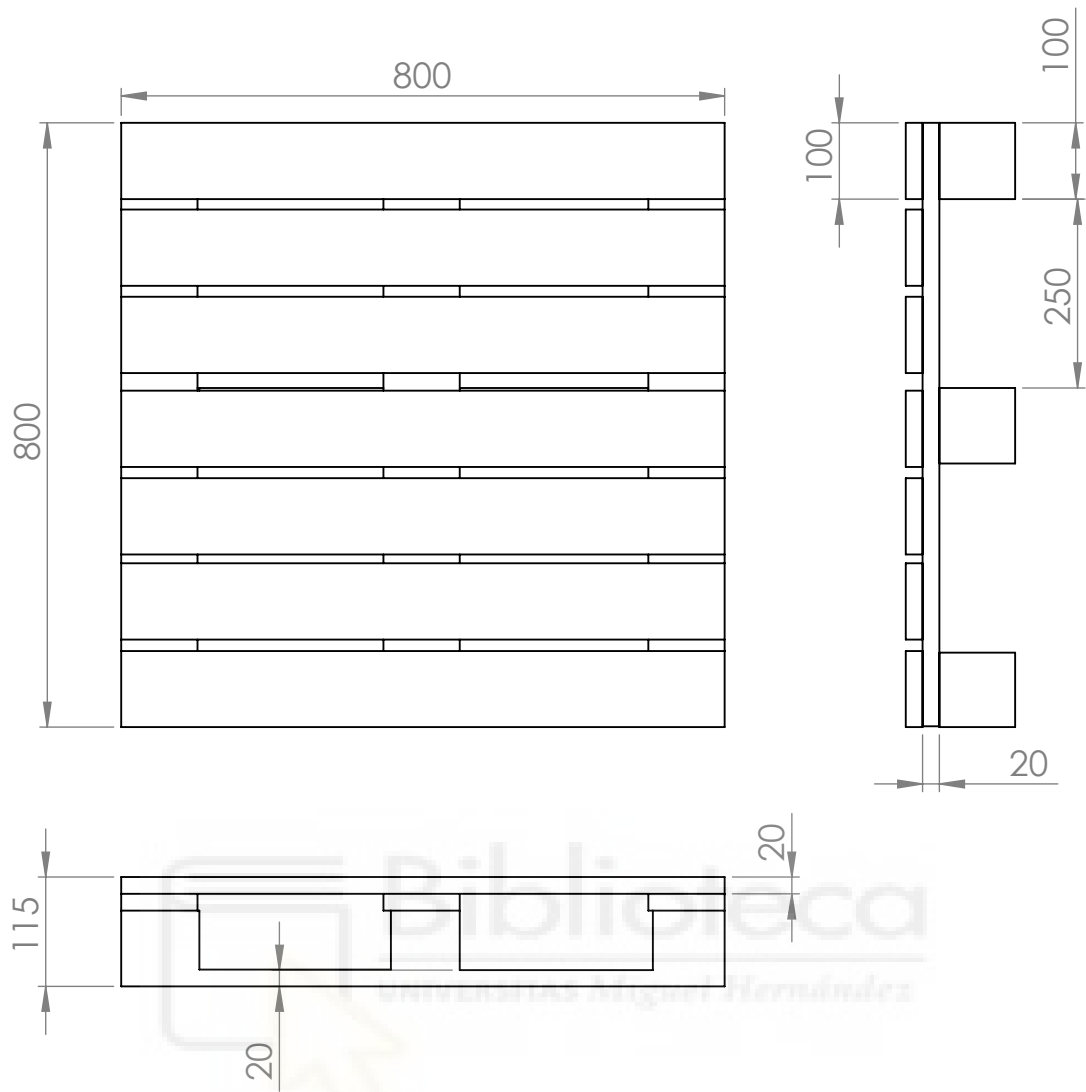


	NOMBRE	FIRMA	FECHA	ESCALA	N.º DE DIBUJO	A4
DIBUJ.	VICENTE CABANES		14/03/2018	1:20	Mesa_listones	
VERIF.						
APROB.			PESO	ESCALA:1:50	HOJA 1 DE 1	



Biblioteca
UNIVERSITAS MIGUEL LEON DE CERDAS

	NOMBRE	FIRMA	FECHA	ESCALA	N.º DE DIBUJO	
DIBUJ.	VICENTE CABANES		14/03/2018	1:20	Mesa_tacos	A4
VERIF.						
APROB.			PESO	ESCALA:1:20	HOJA 1 DE 1	



	NOMBRE	FIRMA	FECHA	ESCALA	N.º DE DIBUJO	
DIBUJ.	VICENTE CABANES		14/03/2018	1:10	Palet	A4
VERIF.						
APROB.			PESO	ESCALA:1:10	HOJA 1 DE 1	

4 3 2 1

F

F

115

96

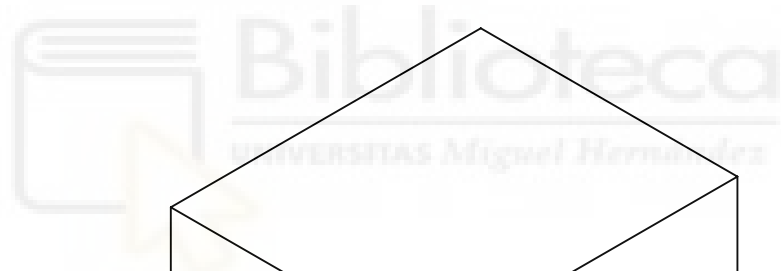
E

E

115

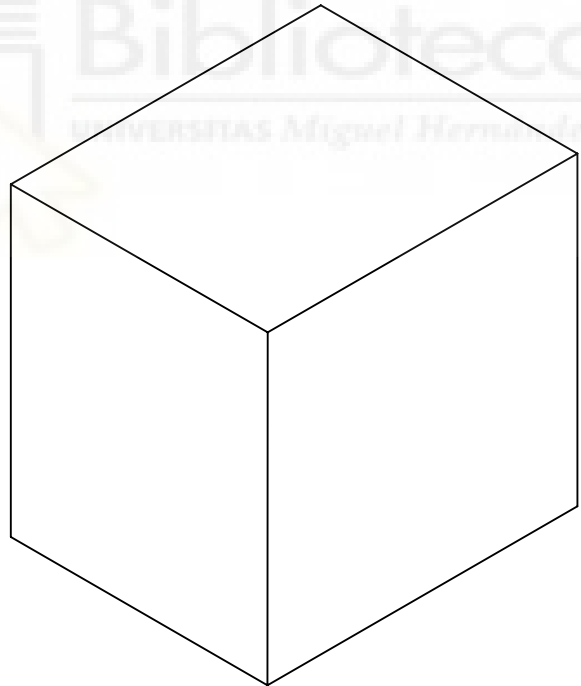
D

D



C

C



B

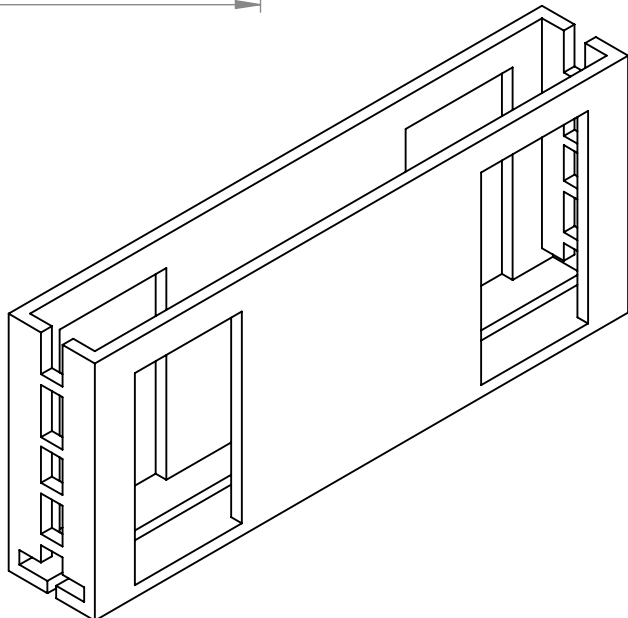
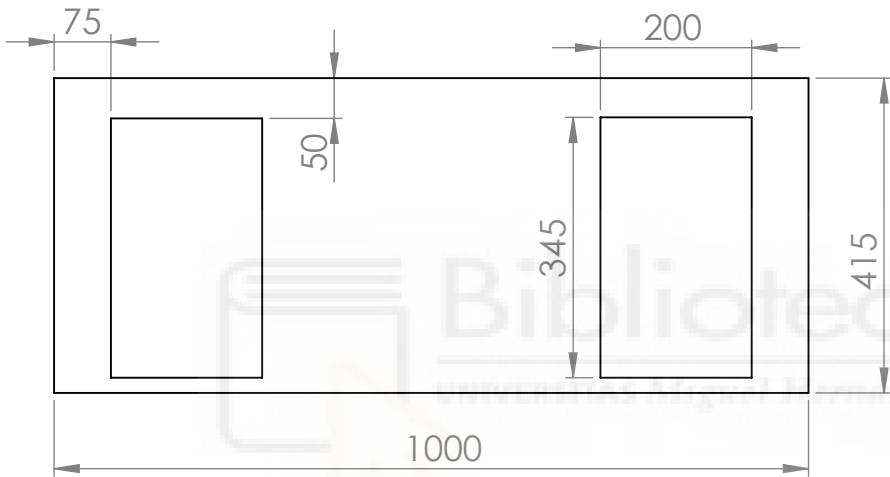
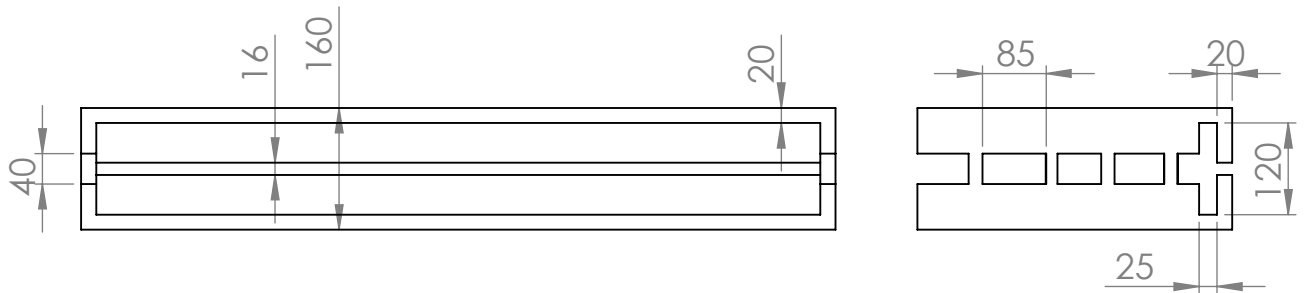
B

A

A

	NOMBRE	FIRMA	FECHA	ESCALA	N.º DE DIBUJO	
DIBUJ.	VICENTE CABANES		14/03/2018	1:2	Taco	A4
VERIF.						
APROB.			PESO	ESCALA:1:2	HOJA 1 DE 1	

4 3 2 1

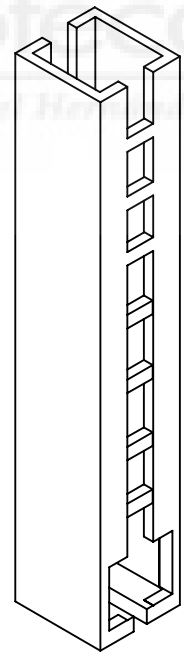
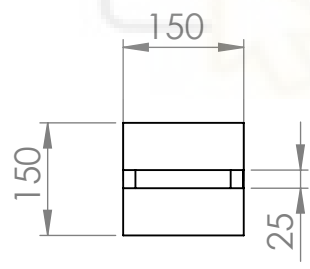
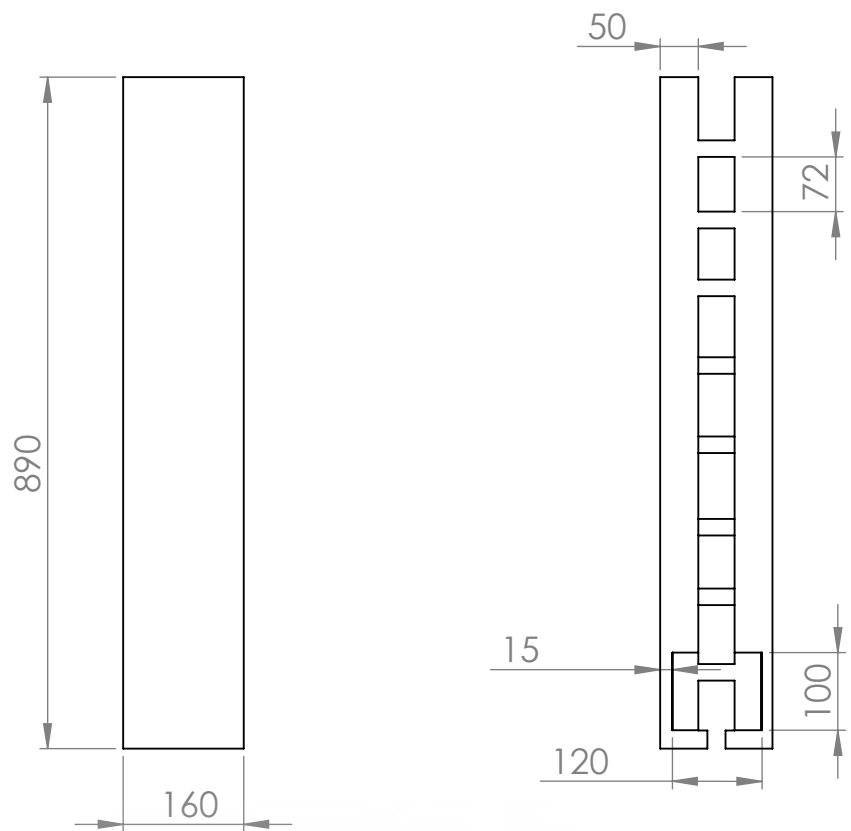


	NOMBRE	FIRMA	FECHA	ESCALA	N.º DE DIBUJO	
DIBUJ.	VICENTE CABANES		14/03/2018	1:10	Torre_listones	A4
VERIF.						
APROB.			PESO	ESCALA:1:20	HOJA 1 DE 1	

4 3 2 1

F
E
D
C
B
A

F
E
D
C
B
A



Biblioteca
UNIVERSITAS Miguel Hernández

NOMBRE		FIRMA		FECHA		ESCALA		N.º DE DIBUJO		A4	
DIBUJ. VICENTE CABANES				14/03/2018		1:10		Torre_tacos			
VERIF.								ESCALA:1:10		HOJA 1 DE 1	
APROB.				PESO							

4 3 2 1