



# ROBÓTICA

**CURSO 2010/2011**

Prácticas. Introducción a la utilización de RobotStudio ABB

## ■ Introducción a la utilización de RobotStudio ABB

1. Descripción del software.
2. Conceptos y nomenclatura. Interfaz de usuario.
3. Insertar un sistema robótico específico. El controlador virtual.
4. Cinemática directa e inversa. Movimiento lineal y articular.
5. Programación de un robot.
6. Panel de control y modos de funcionamiento. El *Virtual FlexPendant*.
7. Ejercicios de la práctica.



# Introducción a la utilización de RobotStudio ABB

## Objetivos

- Conocer las características y capacidades de RobotStudio ABB.
- Saber manejar y mover un robot dentro del entorno de simulación.
- Conocer la metodología para la creación y movimiento de un robot industrial dentro de RobotStudio.
- Tener la capacidad de crear trayectorias lineales y articulares mediante el entorno de simulación.



## 1. DESCRIPCIÓN DEL SOFTWARE

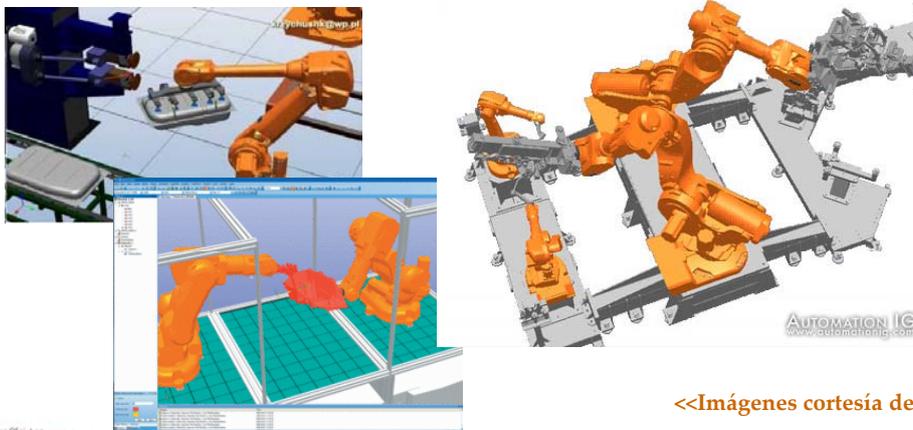
## III Descripción del software

- RobotStudio ABB (versión 5.13.02) permite crear, programar y simular células y estaciones de robots industriales ABB.
- Simulador comercial potente, con diversas características y capacidades:
  - Creación automática de cualquier tipo de estación.
  - Importación de geometrías y modelos 3D de cualquier formato (RobotStudio trabaja sobre CATIA).
  - Programación y simulación cinemática de las estaciones.
  - Facilidad de diseño y creación de células robóticas (robot y dispositivos).
  - Permite exportar los resultados obtenidos en simulación a la estación real.



## III Descripción del software

- Entorno virtual muy realista que permite simular de manera muy precisa tu aplicación/proceso real.



<<Imágenes cortesía de ABB>>



## ■ Descripción del software

- El simulador RobotStudio funciona sobre *RobotWare* (software que se instala antes) → conjunto de archivos necesarios para implementar todas las funciones (virtual-real), configuraciones, datos y programas necesarios para el control del sistema robot.



Por cada modelo físico de robot nuevo,  
se actualiza la versión de *RobotWare*

- Programación en RAPID →  
(lenguaje propio ABB)



```
PROC main()  
  
MoveL rob_reposo, v200, z5, pinza;  
WaitDI pieza_presente, 1;  
MoveL rob_cinta, v50, z5, pinza;  
if (pieza_correcta=TRUE) THEN  
  MoveL rob_correcta, v200, z5, pinza;  
ELSE  
  MoveL rob_correcta, v200, z5, pinza;  
ENDIF
```



## 2. SISTEMAS DE COORDENADAS. CONFIGURACIÓN DEL ROBOT

## ■ Sistemas de coordenadas

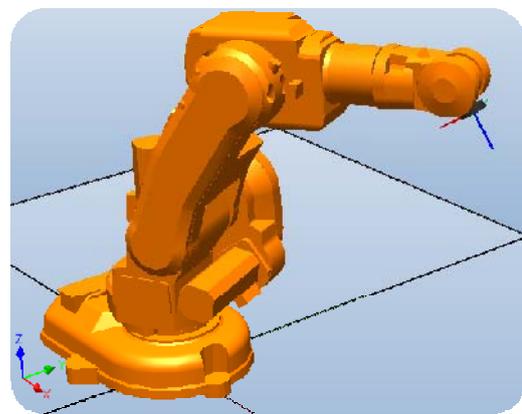
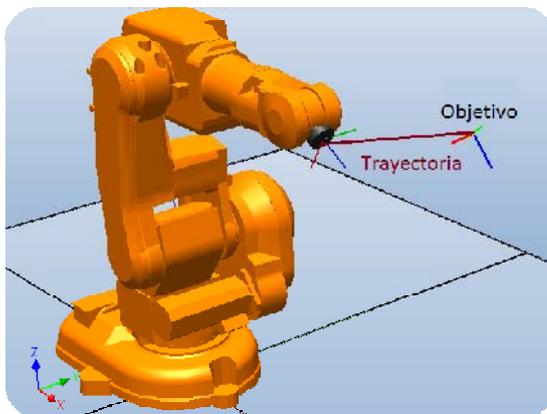
### ■ Objetivos y trayectorias

- Objetivo: posición o coordenada que debe ser alcanzada por el robot. Un objetivo posee una **posición** ( $x,y,z$  definidos en un sistema de coordenadas), una **orientación** ( $rx, ry, rz$ ), que cuando el robot la alcanza alinea el TCP (*Tool Central Point*) con dicho valor, y una **configuración**, que especifican la forma en el que el robot debe alcanzar el objetivo.
- Trayectoria: secuencia de instrucciones de movimiento hacia los objetivos. Una instrucción de movimiento se compone de:
  - Referencia a objetivo.
  - Datos del movimiento (tipo-articular, lineal, circular; velocidad, precisión, etc.)
  - Una referencia a *datos de herramienta* y a un *objeto de trabajo*.



## ■ Sistemas de coordenadas

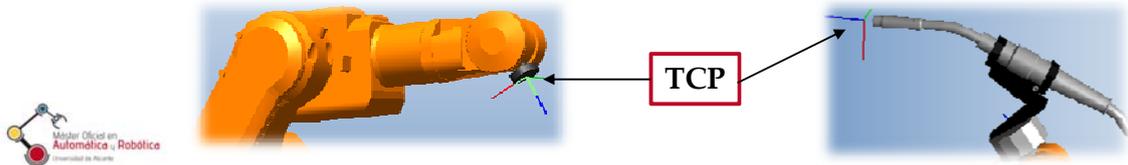
### ■ Conceptos y nomenclatura



## Configuraciones del robot

### Sistemas de coordenadas

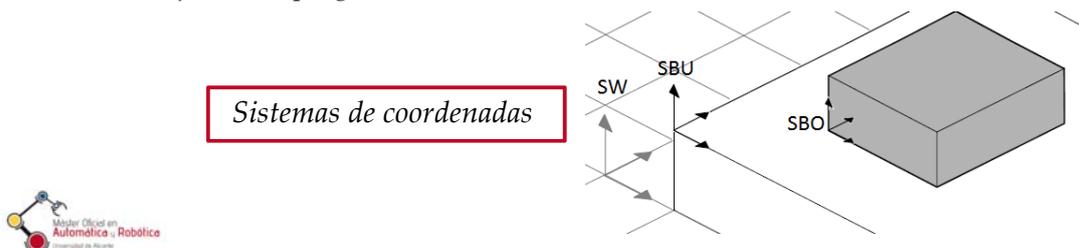
- Sistema de coordenadas del mundo: sistema de coordenadas que representa la totalidad de la estación o célula de robot. Se utiliza como referencia base del resto de sistemas de coordenadas (direcciones XYZ).
- Sistema de coordenadas de la base del robot: cada robot de la estación tiene un sistema de coordenadas situado en la base del robot, utilizado como referencia para sus objetivos y trayectorias.
- Sistema de coordenadas del punto central de la herramienta (TCP): el TCP es el punto situado en el centro de la herramienta. Todos los robots tienen un TCP predefinido en el punto de montaje de la herramienta del robot → *tool0*. Al ejecutar una trayectoria o programa, el robot mueve el TCP hasta dicha posición.



## Configuraciones del robot

### Sistemas de coordenadas

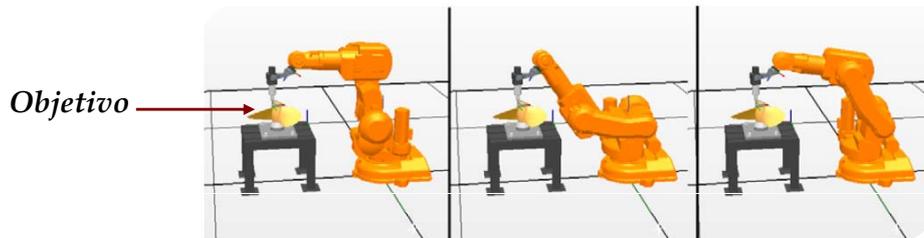
- Sistema de coordenadas del objeto de trabajo: el objeto de trabajo representa a la pieza de trabajo física y posee dos sistemas de coordenadas:
  - *Base de coordenadas del usuario*: definido como base para el objeto de trabajo.
  - *Base de coordenadas del objeto de trabajo*: si no se especifica ningún valor, los objetivos (posiciones) dependen del objeto de trabajo *Wobj0*, que coincide con el sistema de coordenadas de la base del robot.
- El uso de objetos de trabajo se utiliza cuando existen piezas de trabajo para ajustar los programas o movimiento del robot fácilmente.



## Configuraciones del robot

### Configuraciones de ejes

- Los objetivos se definen con respecto a *Wobj0*, sistema de coordenadas dextrógiro XYZ situado en la base del robot. El controlador puede dar diversas soluciones para alcanzar el mismo objetivo definido.



- Para distinguir entre las configuraciones, todos los objetivos tienen un valor de que especifica en qué cuadrante está cada eje.



## Configuraciones del robot

### Designación de la configuración

- Las configuraciones de los ejes del robot se designan con cuatro números enteros ( $n1, n2, n3, n4$ ) que especifican en qué cuadrante de una revolución completa de encuentran los ejes.

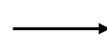
$(n1, n2, n3, n4)$



- 0,3: cuadrantes positivos (sentido contrario agujas del reloj)

- 1,-4: cuadrantes negativos (sentido agujas del reloj)

- $n1$  = posición del eje 1
- $n2$  = posición del eje 4
- $n3$  = posición del eje 6
- $n4$  = posición del eje  $\otimes$



Eje virtual que especifica el centro de la muñeca respecto a los demás ejes



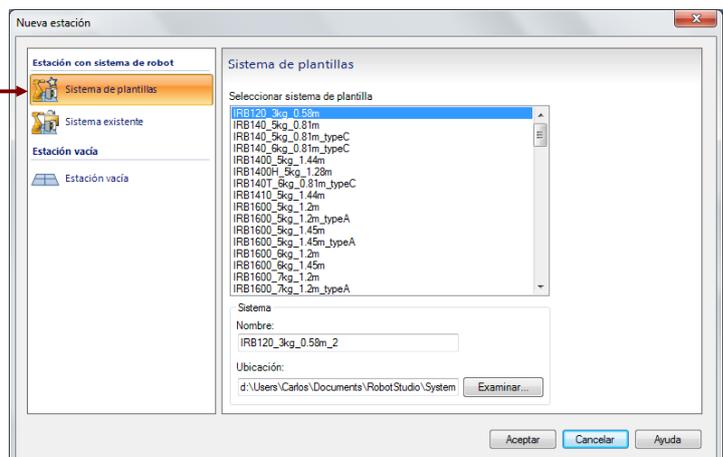
Punto muñeca

# 3. INSERCIÓN DE UN SISTEMA ROBÓTICO. EL CONTROLADOR VIRTUAL

## III Implementación de un sistema robótico

- Nueva estación
  - Sistema de plantillas
  - Sistema existente

↓  
Sistemas robóticos guardados  
previamente por el usuario

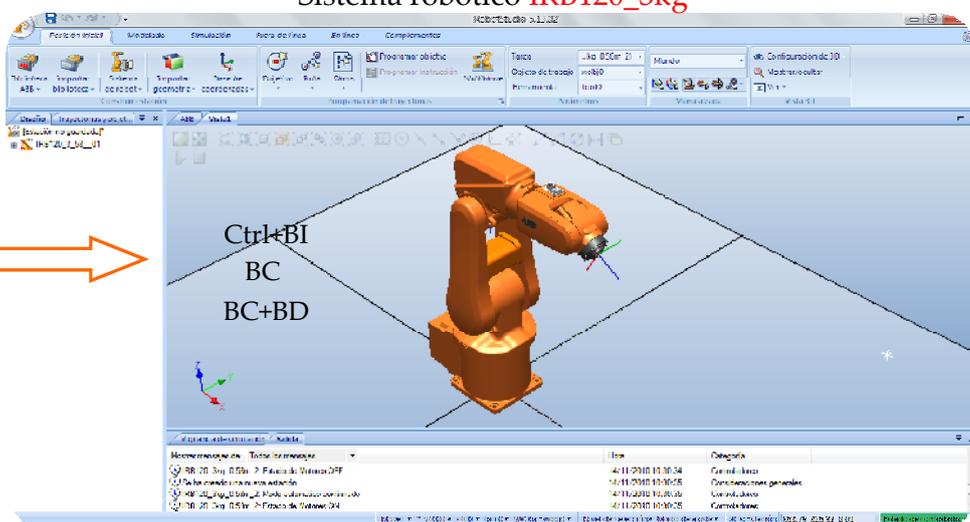


# Implementación de un sistema robótico

Sistema robótico IRB120\_3kg

Nueva estación

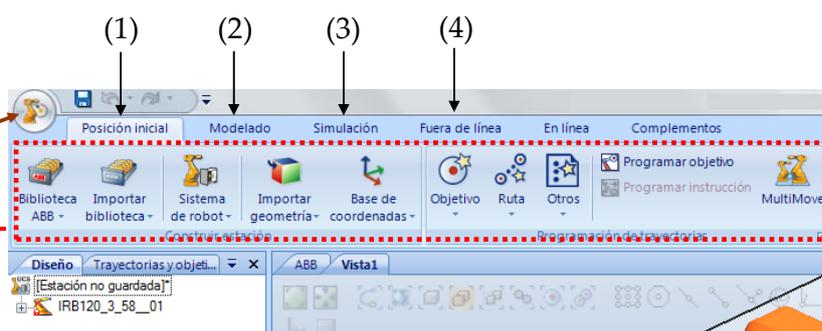
- Vista 3D
- Traslado
- Zoom
- Giro



# Implementación de un sistema robótico

- Interfaz gráfica

Menú aplicación



(1) Inicio: controles para construir estaciones, crear sistemas y programar trayectorias

(2) Modelado: controles para crear y agrupar componentes, cuerpos y diseño CAD

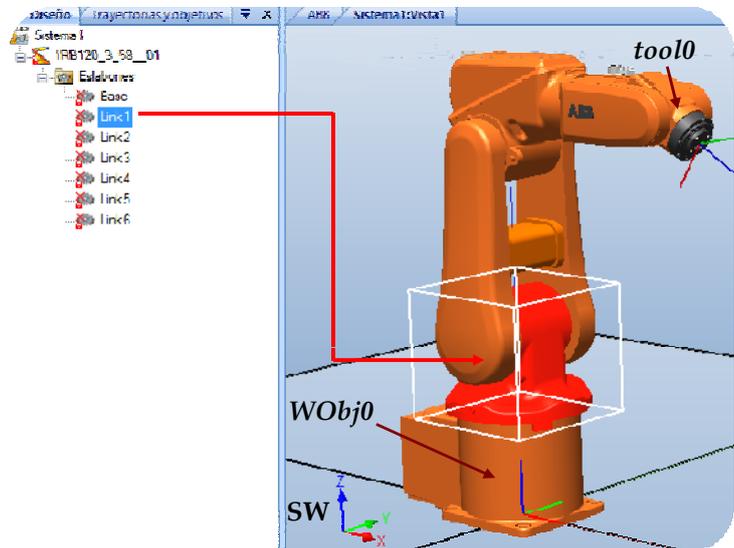
(3) Simulación: controles para crear, configurar, monitorizar y grabar simulaciones

(4) Fuera de línea: controles para la sincronización, configuración y tareas del controlador virtual



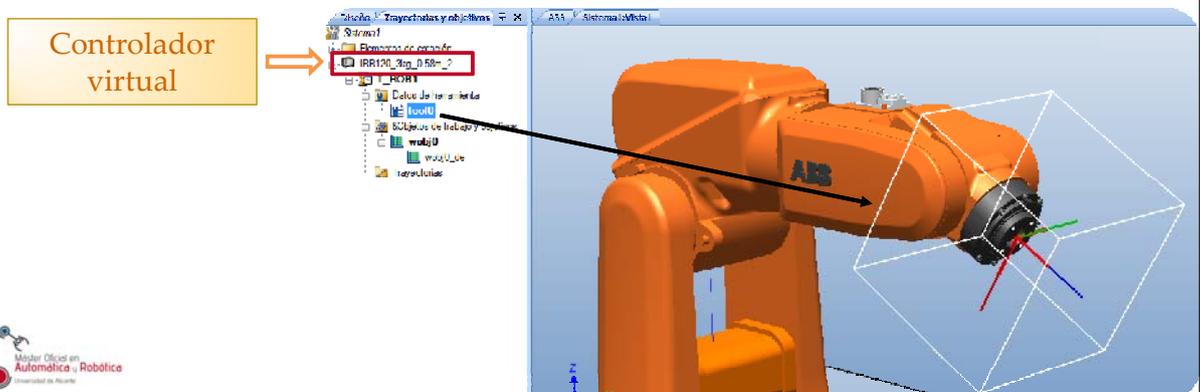
## Implementación de un sistema robótico

- Submenú *Diseño*
  - Dentro de este menú se encontrarán todos los mecanismos incluidos en el proyecto (robots, herramientas terminales, *tracks*, piezas 3D, etc.).
  - Información sobre el robot y su **pose inicial**: *Modificar mecanismo*.



## Implementación de un sistema robótico

- Submenú *Trayectorias y Objetivos*
  - Dentro de este menú se visualizarán los objetivos y trayectorias programados del sistema robótico. También se encuentran datos relativos a *tool0* y *Wobj0*.



## III Implementación de un sistema robótico

- El controlador virtual (VC)
  - El controlador virtual de un sistema robótico simulado utiliza el mismo software implementado por el controlador real para ejecutar código RAPID, para calcular los movimientos del robot y para el manejo de las señales E/S.
  - Inicialización VC



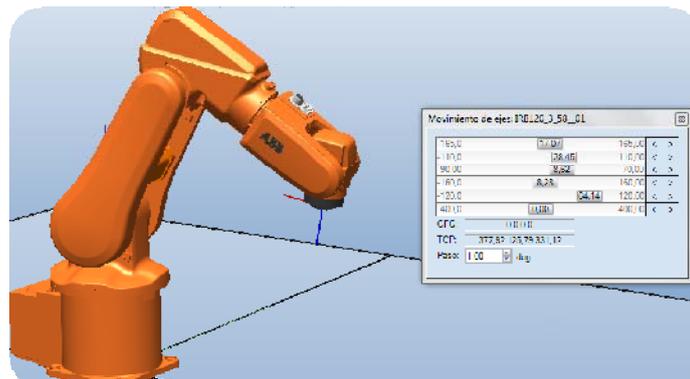
Puesta en marcha	Descripción
Automáticamente, al crear una estación	En la mayoría de los casos, el controlador virtual se inicia al crear una nueva estación. A continuación, los archivos de biblioteca de los robots utilizados por el sistema se importan a la estación
Manualmente, al conectarse a una biblioteca importada	Si ha importado manualmente una biblioteca de robot que desea utilizar con un sistema, en lugar de importar una nueva biblioteca al iniciar el programa, puede conectar esta biblioteca a un controlador.



## 4. CINEMÁTICA DIRECTA E INVERSA

## III Movimiento articular

- Cinemática directa/Movimiento articular
  - *Movimiento de ejes de mecanismo*
    - BD sobre el mecanismo robot en submenú Diseño
    - Menú contextual al pulsar mecanismo robot en submenú Diseño



Valores articulares  
( $q_1 \dots q_6$ )

Ejes robot

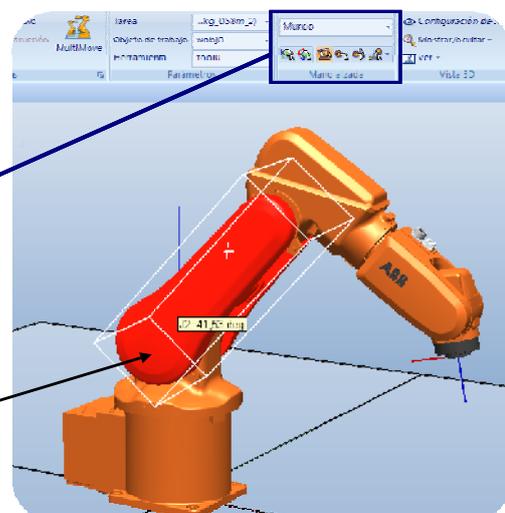


## III Movimiento articular

- Cinemática directa/Movimiento articular
  - *Movimiento de eje (mano alzada)*

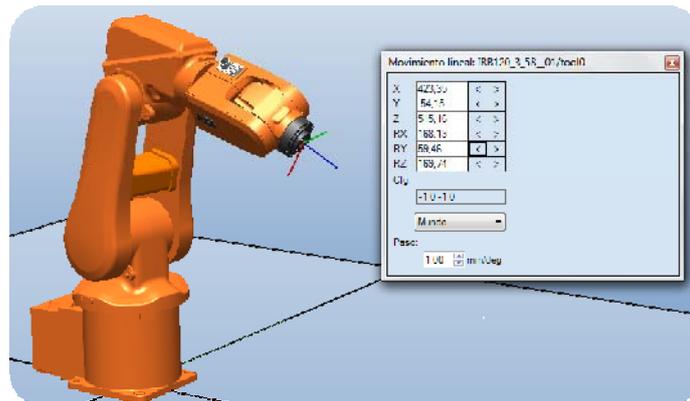
Movimiento interactivo:  
seleccionas el eslabón y se gira  
con el movimiento del ratón

$q_1 \dots q_6$  (Coord. Articulares)



## ▮▮▮ Movimiento lineal

- Cinemática inversa/Movimiento lineal
  - *Movimiento lineal del mecanismo*
    - BD sobre el mecanismo robot en submenú Diseño
    - Menú contextual al pulsar mecanismo robot en submenú Diseño



Valores cartesianos  
 Pos. → (X, Y, Z)  
 Ori. → (RX, RY, RZ)

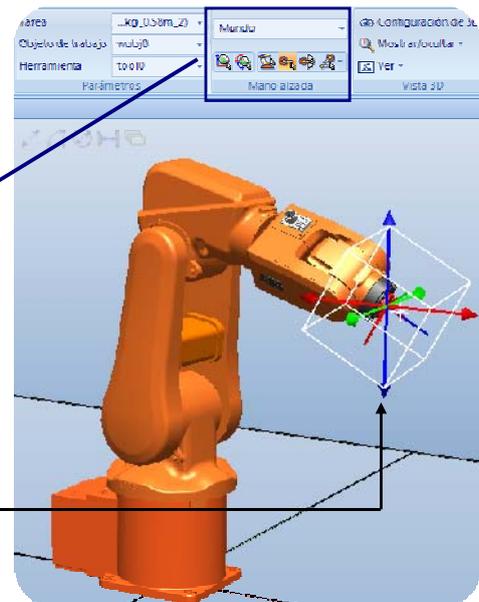


## ▮▮▮ Movimiento lineal

- Cinemática inversa/Movimiento lineal
  - *Movimiento lineal (mano alzada)*

Movimiento interactivo:  
 seleccionas el extremo y se activan tres ejes XYZ para moverlo de forma lineal

X, Y, Z (Coord. Cartesianas)

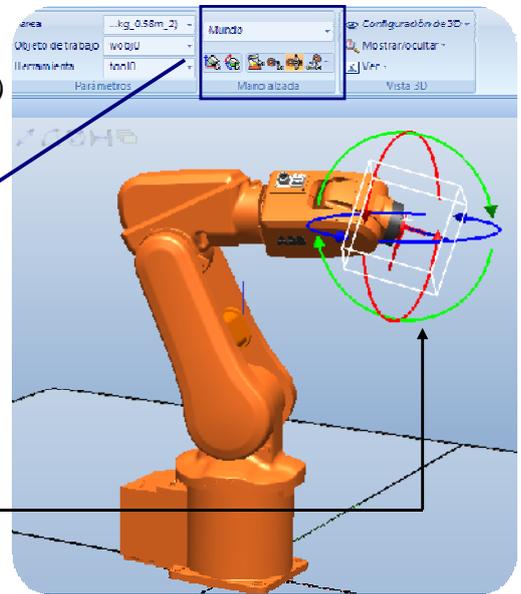


## Movimiento lineal

- Cinemática inversa/Orientación
  - *Movimiento de reorientación (mano alzada)*

Movimiento interactivo:  
seleccionas el extremo o la base y se  
activan tres ejes para rotación XYZ  
para orientar al sistema robótico

Roll (RZ), Pitch (RY), Yaw (RZ)

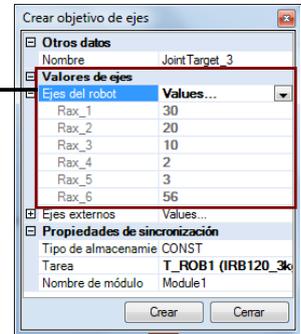
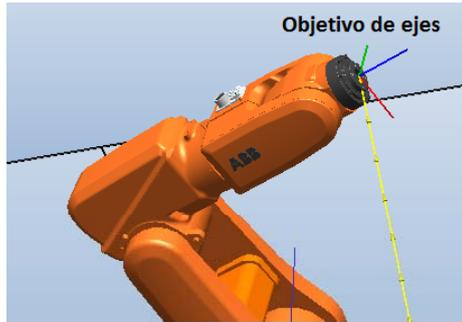


## 5. PROGRAMACIÓN DE UN ROBOT

## Programación de un robot

- Creación de objetivos (ejes)
  - Posición inicial → Objetivo → Crear objetivo de ejes.
    - Valores articulares →  $q_1 \dots q_6$  (°).

Se almacena en objetivos de ejes (submenú trayectorias y objetivos)



Patrones

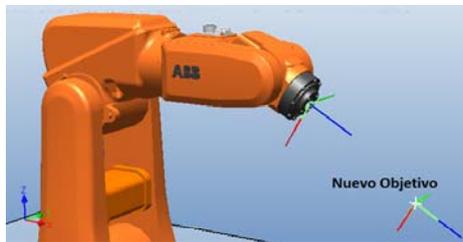
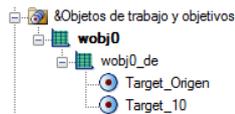
Ejes robot → `-MoveAbsJ`  
 Ejes externos → `-MoveExtJ`



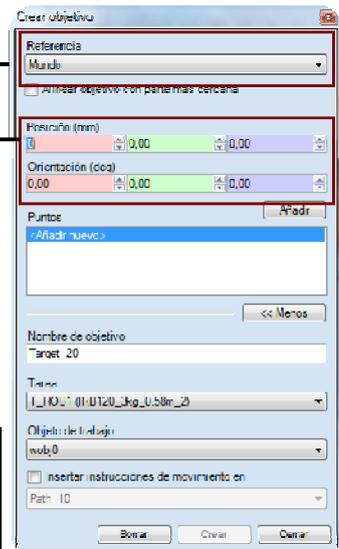
## Programación de un robot

- Creación de objetivos (puntos)
  - Posición inicial → Objetivo → Crear objetivo.
    - Posición X, Y, Z y orientación RX, RY, RZ.

Se almacena en objetivos (submenú trayectorias y objetivos)

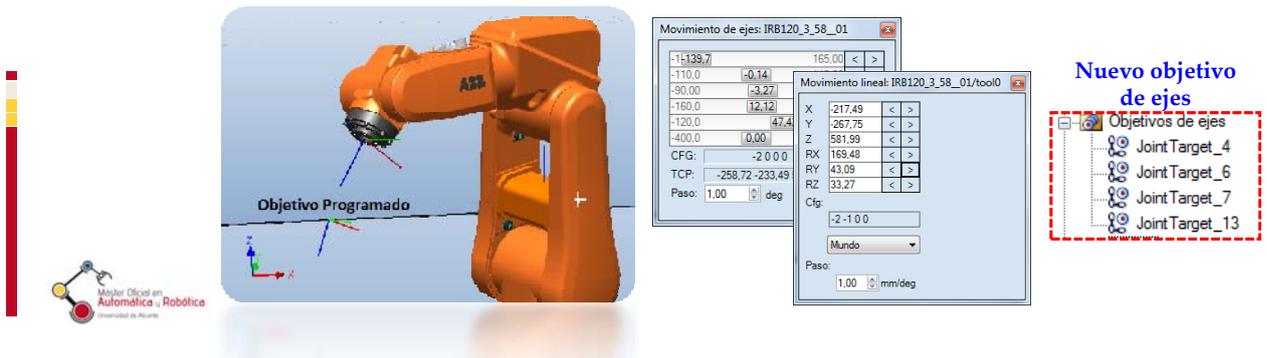


De forma absoluta en el sistema de coordenadas mundo de la estación	Mundo
De forma relativa con respecto a la posición del objeto de trabajo activo	Objeto de trabajo
En un sistema de coordenadas definido por el usuario	UCS



## Programación de un robot

- Programación de objetivos
  - Posición inicial → Programar objetivo
    - Para programar un objetivo, se mueve el robot a una posición determinada con alguna de las herramientas para la cinemática comentadas y se pulsa el control *Programar objetivo*.
    - Se crea de forma automática un nuevo **objetivo de ejes** al pulsar el control.



## Programación de un robot

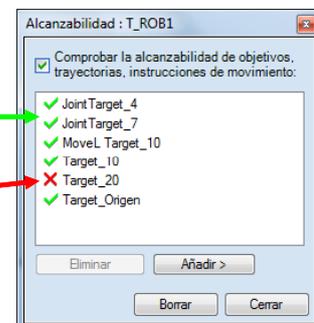
- Comprobar el alcance de los objetivos y trayectorias
  - BD sobre objetivo/trayectoria → Comprobar alcanzabilidad
    - Comprueba si el robot es capaz de alcanzar dicho objetivo. Esto no significa que el robot pueda realizar la trayectoria generada hasta dicho objetivo.

Posibles valores de alcanzabilidad según el color

El objeto puede ser alcanzado.

El objeto puede ser alcanzado en su posición actual, pero no con su orientación actual.

El objeto no puede ser alcanzado en su posición actual.



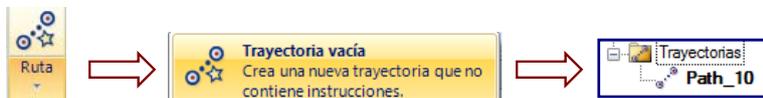
## Programación de un robot

- Comprobar el alcance de los objetivos y trayectorias
  - BD sobre objetivo/trayectoria → Ver robot en objetivo



## Programación de un robot

- Creación de trayectorias (I)
  - Las trayectorias se componen de varios órdenes de movimiento para el robot: *MoveJ* (movimiento articular) y *MoveL* (movimiento lineal).
  - Para crear una trayectoria: **Posición inicial** → **Ruta** → **Trayectoria vacía**.

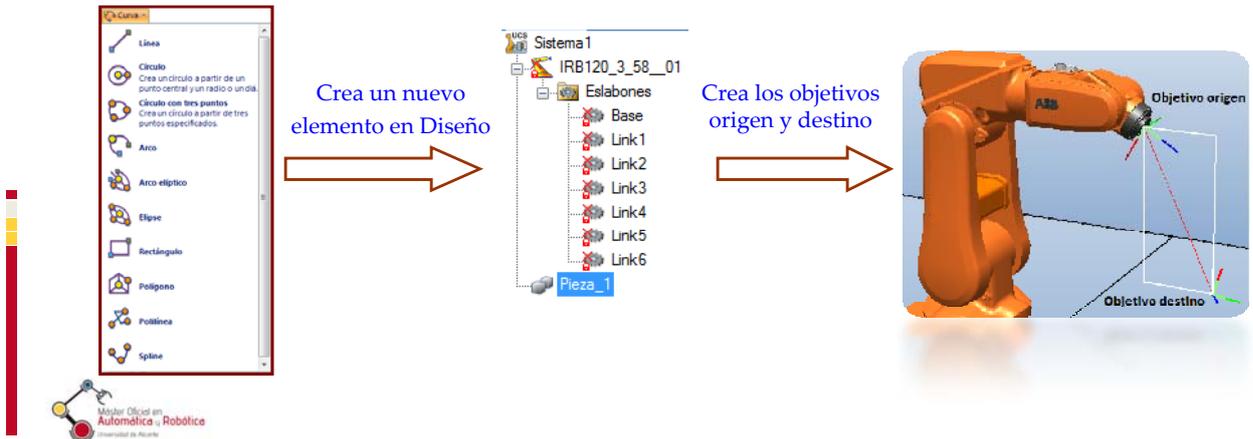


- A partir de los objetivos creados, se crean los órdenes de movimiento.
- Para crear un orden: **BD sobre objetivo** → **Añadir a trayectoria Path10** → **Añadir a nueva trayectoria**



## Programación de un robot

- Creación de trayectorias (II)
  - Las trayectorias también pueden ser creadas a partir de curvas.
  - En primer lugar es necesario crear una curva: **Modelado** → **Curva**



## Programación de un robot

- Creación de trayectorias (II)
  - Después de crear la curva: **Posición inicial** → **Trayectoria partir de curva**

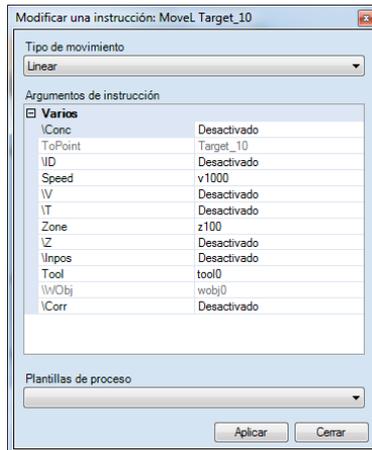
Se selecciona la curva creada

**Parámetros de ajuste de objetivos**

Especificar la rotación alrededor del eje X.	<b>Aproximación</b>
Especificar la rotación alrededor del eje Y.	<b>Recorrido</b>
Especificar la rotación alrededor del eje Z.	<b>Giro</b>
Crear un objetivo de inicio en la distancia especificada en el primer objetivo de la curva, en la dirección de aproximación.	<b>Aproximación</b>
Crear un objetivo de partida en la distancia especificada en el primer objetivo de la curva, en la dirección de aproximación	<b>Partida</b>
Iniciar la trayectoria en una ubicación de la curva distinta del punto de inicio.	<b>Inicio de offset</b>
Finalizar la trayectoria en una ubicación de la curva distinta del punto final.	<b>Fin de offset</b>
Para crear un offset que será igual en todos los objetivos de la trayectoria.	<b>Offset de objetivo local</b>

## Programación de un robot

- Modificación de parámetros de una instrucción de movimiento
  - Para modificar los parámetros: **BD sobre movimiento MoveL/MoveJ** → **Modificar una instrucción.**



### Parámetros instrucción

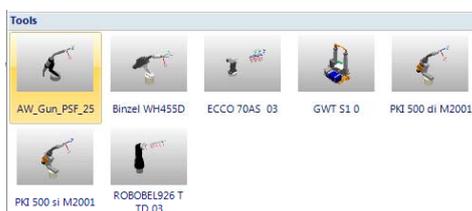
Que las instrucciones posteriores se ejecuten inmediatamente.	\Conc
El objetivo de destino para el objetivo de instrucción.	ToPoint
La velocidad para el punto central de la herramienta, la reorientación de la herramienta y el eje externo.	Speed
La velocidad del TCP en mm/s directamente en la instrucción (será reemplazada por los datos de velocidad correspondientes).	V
El tiempo total en segundos durante el cual se mueve el robot (será sustituido por los datos de velocidad correspondientes).	VT
El tamaño de la trayectoria de esquina generada.	Zone
Especifica con qué precisión se debe alcanzar el TCP la posición solicitada antes de permitir que el robot se mueva hasta el punto siguiente. El valor de z es en mm.	VZ
La herramienta utilizada para el movimiento (el TCP de esta herramienta se situará en el objetivo de destino).	\Tool
El objeto de trabajo al que pertenece la posición de robot de la instrucción.	\Wobj



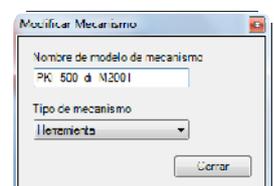
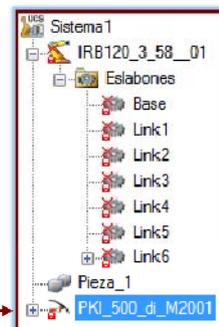
## Programación de un robot

- Adición de una herramienta al sistema robótico
  - Importar una herramienta de una librería como elemento externo.
  - Importar biblioteca** → **Equipamiento** → **Tools**

### Herramientas ABB



Creas un nuevo elemento en Diseño

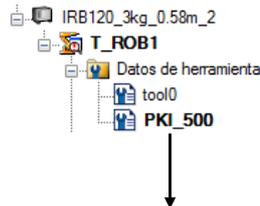


Modificar mecanismo

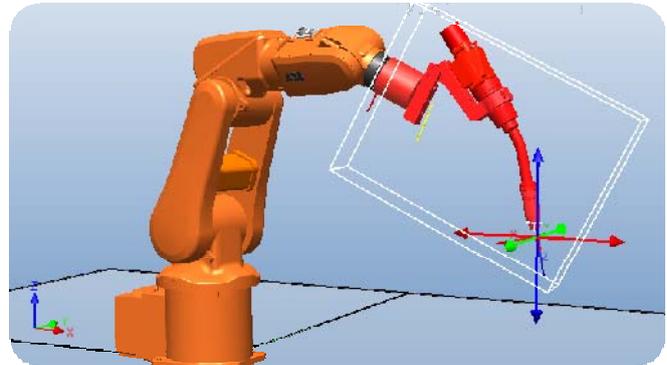


## Programación de un robot

- Adición de una herramienta al sistema robótico
  - Para conectar la herramienta al robot: **BD sobre herramienta → Conectar a**



- Se crea un nuevo sistema de referencia para la herramienta
- Es necesario volver a sincronizar el VC

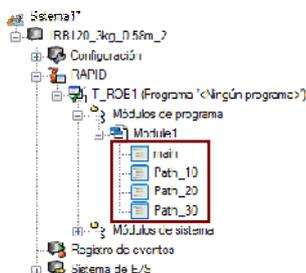


Los movimientos del robot ya se realizan con la herramienta añadida



## Programación de un robot

- Edición del código RAPID
  - El VC genera todo el código RAPID de nuestra simulación. Dicho código es exportable en ejecución en la estación real.
    - Para ver el código generado de la simulación hay que ir a **Fuera de línea**



```

1 MODULE Module1
2   CONST robtarget Target_10:=[[524.401927555647,-2.08378519
3   CONST jointtarget JointTarget_1:=[[30,40,40,0,0,0],[9E9,9
4   CONST jointtarget JointTarget_7:=[[ -12.2837639580932,34.7
5   CONST jointtarget JointTarget_13:=[[ -139.764799058712,14.
6   CONST jointtarget JointTarget_4:=[[60,-40,0,0,0,0],[9E9,9
7   PROC Path_10 ()
8     MoveL Target_10,v50,z100,tool10\Wobj:=wobj0;
9     MoveAbsJ JointTarget_4,v100,z100,tool10\Wobj:=wobj0;
10  ENDPROC
11 PROC main ()
12   Path_10;
13 ENDPROC
14 PROC Path_20 ()
15   MoveAbsJ JointTarget_7,v1000,z100,tool10\Wobj:=wobj0;
16 ENDPROC
17 PROC Path_30 ()
18   MoveAbsJ JointTarget_13,v1000,z100,tool10\Wobj:=wobj0;
19 ENDPROC
20
    
```



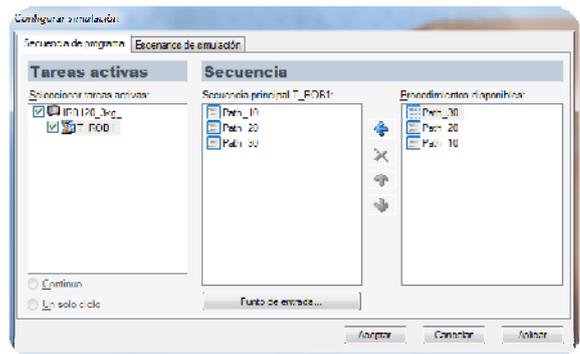
## Programación de un robot

- Simulación de programas
  - Esta opción simula el código RAPID generado por el VC.
  - Permite detectar colisiones, simulación E/S y gestión de eventos.
  - Permite grabar la simulación y guardarlo en formato vídeo.



Para la simulación de las trayectorias generadas tan sólo es necesario seleccionarlas

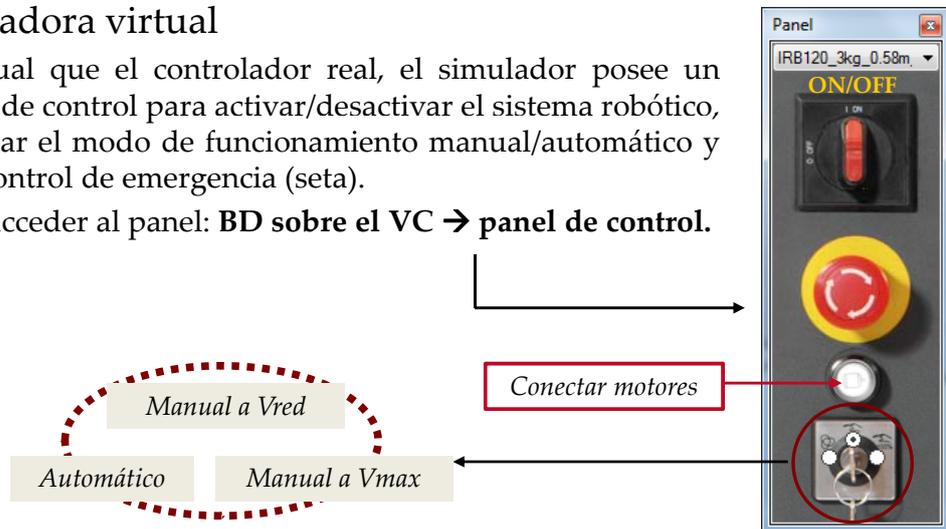
**Simulación → Configuración de simulación**  
y pulsar a **Reproducir**



## 6. PANEL DE CONTROL Y MODOS DE FUNCIONAMIENTO

## Panel de control

- Controladora virtual
  - Al igual que el controlador real, el simulador posee un panel de control para activar/desactivar el sistema robótico, cambiar el modo de funcionamiento manual/automático y una control de emergencia (seta).
  - Para acceder al panel: **BD sobre el VC → panel de control.**



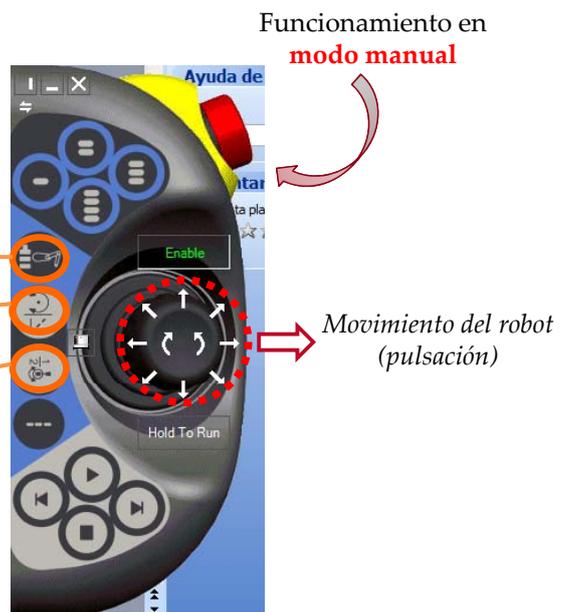
## Manejador manual FlexPendant

- Similar al "Teach Pendant"
  - Simula al controlador manual real.
  - Permite editar programas, configurar el robot, ejecutar instrucciones de movimiento, etc.

**Unidad de movimiento**  
Para mover el robot u otros mecanismos

**Tipo de movimiento**  
Movimiento reorientación o lineal

**Tipo de movimiento**  
Movimiento eje a eje





## 7. EJERCICIOS DE LA PRÁCTICA



### ■ Ejercicios a realizar

- Creación y simulación de un sistema robótico.
  - Cada alumno deberá escoger un sistema robótico determinado.
  - El proyecto RobotStudio deberá contener diversos objetivos (5) y la simulación de trayectorias tanto lineales como articulares.
  - Se deberá crear un par de trayectorias a partir de curvas (una lineal y otra circular).
  - Inclusión de un herramienta existente en las librerías de ABB y se tendrá que conectar con el sistema robótico.
  - El archivo del proyecto deberá ser entregado al profesor de prácticas al final de la sesión o durante la siguiente semana.



# ROBÓTICA

**CURSO 2010/2011**

Prácticas. Introducción a la utilización de RobotStudio ABB